Final Project Submission

Please fill out:

- Student name: Sue Lim
- Student pace: self paced
- Scheduled project review date/time: Mon, Feb 27, 2023, 11:30 AM - 12:15 PM
- Instructor name: Mark Barbour
- Blog post URL: https://medium.com/@limsue9123/which-movie-to-create-for-your-new-business-7bb1440cd7e6 (https://medium.com/@limsue9123/which-movie-to-create-for-your-new-business-7bb1440cd7e6)

Which Movie to Create?

I. Introduction

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. I am charged with exploring what types of films are currently doing the best at the box office. I 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.

In [50]:
```
# Import relevant libraries and set options

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

pd.set_option('float_format', '{:,.0f}'.format)
```

In [51]:
```
# Import raw data files

budget_gross = pd.read_csv("Data/tn.movie_budgets.csv")
genre = pd.read_csv("Data/imdb.title.basics.csv")
gross_studio = pd.read_csv("Data/bom.movie_gross.csv")
people = pd.read_csv("Data/imdb.title.principals.csv")
people_names = pd.read_csv("Data/imdb.name.basics.csv")
```

II. Analysis Overview

To determine which type of movies to create for the new business, I explore which genres were linked to the greatest gross revenue among the movies released between 2010-2020. I use gross revenue because it is an objective indicator of popularity, and it is crucial for the new business to create high-impact popular movies that would boost its reputation, enabling its subsequent movies to receive attention. For this reason, I prioritize gross revenue over profit.

Once I determine which genre of movies to produce, I investigate which actors/actresses or directors are associated with the greatest average gross revenue. This is to offer preliminary guidance on whom to hire for the new movies.

Finally, I analyze the relationship between the runtime and gross revenue, and identify the optimal range of runtime for a movie.

III. Data

Box Office Mojo (https://www.boxofficemojo.com/) provides gross revenue information for movies.

In [52]:
```
# Gross revenue data 1

gross_studio
```

Out[52]:

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415,000,000	652000000	2010
1	Alice in Wonderland (2010)	BV	334,200,000	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296,000,000	664300000	2010
3	Inception	WB	292,600,000	535700000	2010
4	Shrek Forever After	P/DW	238,700,000	513900000	2010
...	...	...	...	...	...
3382	The Quake	Magn.	6,200	NaN	2018
3383	Edward II (2018 re-release)	FM	4,800	NaN	2018
3384	El Pacto	Sony	2,500	NaN	2018
3385	The Swan	Synergetic	2,400	NaN	2018
3386	An Actor Prepares	Grav.	1,700	NaN	2018

3387 rows × 5 columns

However, the list is not comprehensive, and there is another data source that [The Numbers (https://www.the-numbers.com/)](https://www.the-numbers.com/) provides.

In [53]:
```
# Gross revenue data 2

budget_gross
```

Out[53]:

	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
...	...	...	...	...	...	...
5777	78	Dec 31, 2018	Red 11	$7,000	$0	$0
5778	79	Apr 2, 1999	Following	$6,000	$48,482	$240,495
5779	80	Jul 13, 2005	Return to the Land of Wonders	$5,000	$1,338	$1,338
5780	81	Sep 29, 2015	A Plague So Pleasant	$1,400	$0	$0
5781	82	Aug 5, 2005	My Date With Drew	$1,100	$181,041	$181,041

5782 rows × 6 columns

[IMDB (https://www.imdb.com/)](https://www.imdb.com/) provides genre and runtime information on movies.

In [54]:
```
# Dataset containing genre information

genre
```

Out[54]:

	tconst	primary_title	original_title	start_year	runtime_minutes	genres
0	tt0063540	Sunghursh	Sunghursh	2013	175	Action,Crime,Drama
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114	Biography,Drama
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122	Drama
3	tt0069204	Sabse Bada Sukh	Sabse Bada Sukh	2018	nan	Comedy,Drama
4	tt0100275	The Wandering Soap Opera	La Telenovela Errante	2017	80	Comedy,Drama,Fantasy
...	...	...	...	...	...	...
146139	tt9916538	Kuambil Lagi Hatiku	Kuambil Lagi Hatiku	2019	123	Drama
146140	tt9916622	Rodolpho Teóphilo - O Legado de um Pioneiro	Rodolpho Teóphilo - O Legado de um Pioneiro	2015	nan	Documentary
146141	tt9916706	Dankyavar Danka	Dankyavar Danka	2013	nan	Comedy
146142	tt9916730	6 Gunn	6 Gunn	2017	116	NaN
146143	tt9916754	Chico Albuquerque - Revelações	Chico Albuquerque - Revelações	2013	nan	Documentary

146144 rows × 6 columns

IMDB also offers data on people associated with each movie.

In [55]:
```
# Dataset on people associated with each movie

people
```

Out[55]:

	tconst	ordering	nconst	category	job	characters
0	tt0111414	1	nm0246005	actor	NaN	["The Man"]
1	tt0111414	2	nm0398271	director	NaN	NaN
2	tt0111414	3	nm3739909	producer	producer	NaN
3	tt0323808	10	nm0059247	editor	NaN	NaN
4	tt0323808	1	nm3579312	actress	NaN	["Beth Boothby"]
...	...	...	...	...	...	...
1028181	tt9692684	1	nm0186469	actor	NaN	["Ebenezer Scrooge"]
1028182	tt9692684	2	nm4929530	self	NaN	["Herself","Regan"]
1028183	tt9692684	3	nm10441594	director	NaN	NaN
1028184	tt9692684	4	nm6009913	writer	writer	NaN
1028185	tt9692684	5	nm10441595	producer	producer	NaN

1028186 rows × 6 columns

However, the list above does not have the persons' actual names, so I use the dataset below from IMDB to identify the names.

In [56]:
```
# Mapping between "nconst" and actual names

people_names
```

Out[56]:

	nconst	primary_name	birth_year	death_year	primary_profession	known_for_titles
0	nm0061671	Mary Ellen Bauder	nan	nan	miscellaneous,production_manager,producer	tt0837562,tt2398241,tt0844471,tt0118553
1	nm0061865	Joseph Bauer	nan	nan	composer,music_department,sound_department	tt0896534,tt6791238,tt0287072,tt1682940
2	nm0062070	Bruce Baum	nan	nan	miscellaneous,actor,writer	tt1470654,tt0363631,tt0104030,tt0102898
3	nm0062195	Axel Baumann	nan	nan	camera_department,cinematographer,art_department	tt0114371,tt2004304,tt1618448,tt1224387
4	nm0062798	Pete Baxter	nan	nan	production_designer,art_department,set_decorator	tt0452644,tt0452692,tt3458030,tt2178256
...	...	...	...	...	...	...
606643	nm9990381	Susan Grobes	nan	nan	actress	NaN
606644	nm9990690	Joo Yeon So	nan	nan	actress	tt9090932,tt8737130
606645	nm9991320	Madeline Smith	nan	nan	actress	tt8734436,tt9615610
606646	nm9991786	Michelle Modigliani	nan	nan	producer	NaN
606647	nm9993380	Pegasus Envoyé	nan	nan	director,actor,writer	tt8743182

606648 rows × 6 columns

IV. Process Steps

I combine the datasets introduced above to use it for different analyses. As the first step, I stack the two gross revenue datasets and calculate the total gross for a movie. Then, I merge it in the genre dataset. This will be the first final dataset.

In [86]:
```
# Stack two datasets containing gross revenue figures for movies,
# after 1) calculating the total gross revenue; 2) deriving release years; and 3) cleaning up the title names

gross_studio["total_gross"] = gross_studio["domestic_gross"].fillna(0) + pd.to_numeric(gross_studio["foreign_gross"].s
gross_studio.groupby("studio")["total_gross"].mean().sort_values(ascending = False)

budget_gross[["worldwide_gross", "domestic_gross", "production_budget"]] = budget_gross[["worldwide_gross", "domestic_
budget_gross["total_gross"] = budget_gross["worldwide_gross"]
budget_gross["year"] = pd.to_numeric(budget_gross["release_date"].apply(lambda x: x[-4:]))

gross = pd.concat([gross_studio, budget_gross.rename(columns = {"movie": "title"})])
gross["title"] = gross["title"].str.replace("&", "and")
gross["title"] = gross["title"].str.replace(" \(.*\)|\W", "")
gross["title"] = gross["title"].str.replace(" Ep ", " Episode ")
gross["title"] = gross["title"].str.replace(" I$", " 1")
gross["title"] = gross["title"].str.replace(" II$", " 2")

gross = gross.sort_values(by = "total_gross")
gross = gross.drop_duplicates(subset = "title", keep = "first")
gross = gross[gross["year"] >= 2010]
gross
```

Out[86]:

	title	studio	domestic_gross	foreign_gross	year	total_gross	id	release_date	production_budget	worldwide_gross
1476	Storage24	Magn.	100	NaN	2013	100	nan	NaN	nan	nan
2757	Satanic	Magn.	300	NaN	2016	300	nan	NaN	nan	nan
2756	NewsFromPlanetMars	KL	300	NaN	2016	300	nan	NaN	nan	nan
2321	TheChambermaid	FM	300	NaN	2015	300	nan	NaN	nan	nan
3078	222	Magn.	400	NaN	2017	400	nan	NaN	nan	nan
...	...	...	...	...	...	...	...	...	...	...
5761	StoriesofOurLives	NaN	nan	NaN	2014	nan	62	Dec 31, 2014	15,000	nan
5771	FamilyMotocross	NaN	nan	NaN	2015	nan	72	May 19, 2015	10,000	nan
5772	Newlyweds	NaN	nan	NaN	2012	nan	73	Jan 13, 2012	9,000	nan
5777	Red11	NaN	nan	NaN	2018	nan	78	Dec 31, 2018	7,000	nan
5780	APlagueSoPleasant	NaN	nan	NaN	2015	nan	81	Sep 29, 2015	1,400	nan

4164 rows × 10 columns

In [88]:
```
# Merge the dataset containing genres for movies and the stacked gross revenue dataset

genre["primary_title"] = genre["primary_title"].str.replace("&", "and")
genre["primary_title"] = genre["primary_title"].str.replace(" \(.*\)|\W", "")
genre["primary_title"] = genre["primary_title"].str.replace(" Ep ", " Episode ")

merged = genre.merge(gross, left_on = ["primary_title", "start_year"], right_on = ["title", "year"], how = "inner")
merged
```

Out[88]:

	tconst	primary_title	original_title	start_year	runtime_minutes	genres	title	studio	domestic_g
0	tt0249516	Foodfight	Foodfight!	2012	91	Action,Animation,Comedy	Foodfight	NaN	
1	tt0315642	Wazir	Wazir	2016	103	Action,Crime,Drama	Wazir	Relbig.	1,100
2	tt0337692	OntheRoad	On the Road	2012	124	Adventure,Drama,Romance	OntheRoad	IFC	744
3	tt0359950	TheSecretLifeofWalterMitty	The Secret Life of Walter Mitty	2013	114	Adventure,Comedy,Drama	TheSecretLifeofWalterMitty	Fox	58,200
4	tt0365907	AWalkAmongtheTombstones	A Walk Among the Tombstones	2014	114	Action,Crime,Drama	AWalkAmongtheTombstones	Uni.	26,300
...	...	...	...	...	...	...	...	...	
2480	tt8852552	Icarus	Icarus	2010	78	Thriller	Icarus	NaN	
2481	tt9024106	Unplanned	Unplanned	2019	106	Biography,Drama	Unplanned	NaN	
2482	tt9078374	LastLetter	Ni hao, Zhihua	2018	114	Drama,Romance	LastLetter	CL	181
2483	tt9151704	BurntheStageTheMovie	Burn the Stage: The Movie	2018	84	Documentary,Music	BurntheStageTheMovie	Trafalgar	4,200
2484	tt9225192	Unstoppable	Seongnan hwangso	2018	116	Action,Crime	Unstoppable	WGUSA	101

2485 rows × 16 columns

The second final dataset will be the first final dataset with the genre variable split into multiple variables. Often, movies are associated with multiple genres and I split them to be able to analyze the movies by unique genre.

In [62]:
```
# As there can be more than one genre for a movie, split the genre field into multiple variables,
# and make the merged dataset wide to long

merged[["genre_1", "genre_2", "genre_3"]] = merged["genres"].str.split(",", expand=True)
merged0 = pd.melt(merged, id_vars = list(merged.columns[:-3]), value_vars = ["genre_1", "genre_2", "genre_3"])
merged1 = merged0.dropna(subset = ["value"])
merged1
```

Out[62]:

	tconst	primary_title	original_title	start_year	runtime_minutes	genres	title	studio	domestic_grc
0	tt0249516	Foodfight	Foodfight!	2012	91	Action,Animation,Comedy	Foodfight	NaN	
1	tt0315642	Wazir	Wazir	2016	103	Action,Crime,Drama	Wazir	Relbig.	1,100,
2	tt0337692	OntheRoad	On the Road	2012	124	Adventure,Drama,Romance	OntheRoad	IFC	744,
3	tt0359950	TheSecretLifeofWalterMitty	The Secret Life of Walter Mitty	2013	114	Adventure,Comedy,Drama	TheSecretLifeofWalterMitty	NaN	58,236,
4	tt0365907	AWalkAmongtheTombstones	A Walk Among the Tombstones	2014	114	Action,Crime,Drama	AWalkAmongtheTombstones	Uni.	26,300,
...	...	...	...	...	...	...	...	...	
7344	tt8043306	TeefainTrouble	Teefa in Trouble	2018	155	Action,Comedy,Crime	TeefainTrouble	NaN	
7346	tt8097306	NobodysFool	Nobody's Fool	2018	110	Comedy,Drama,Romance	NobodysFool	Par.	31,700,
7349	tt8155288	HappyDeathDay2U	Happy Death Day 2U	2019	100	Drama,Horror,Mystery	HappyDeathDay2U	NaN	28,051,
7351	tt8266310	BlindedbytheLight	Blinded by the Light	2019	117	Biography,Comedy,Drama	BlindedbytheLight	NaN	
7355	tt8364368	Crawl	Crawl	2019	nan	Action,Horror,Thriller	Crawl	NaN	

5919 rows × 18 columns

The third final dataset is the dataset that maps between people associated with a movie to the movie's genre, gross revenue, etc.

In [63]:
```
# Next, to explore directors and actors/actresses,
# merge the main dataset with a dataset mapping between movies and people associated with them

people_names0 = people.merge(people_names[["nconst", "primary_name"]], on = "nconst") # Identify names for people list
merged2 = merged.merge(people_names0, on = "tconst", how = "inner")
merged2
```

Out[63]:

	tconst	primary_title	original_title	start_year	runtime_minutes	genres	title	studio	domestic_gross	foreign_gross	...	wor
0	tt0249516	Foodfight	Foodfight!	2012	91	Action,Animation,Comedy	Foodfight	NaN	0	NaN	...	
1	tt0249516	Foodfight	Foodfight!	2012	91	Action,Animation,Comedy	Foodfight	NaN	0	NaN	...	
2	tt0249516	Foodfight	Foodfight!	2012	91	Action,Animation,Comedy	Foodfight	NaN	0	NaN	...	
3	tt0249516	Foodfight	Foodfight!	2012	91	Action,Animation,Comedy	Foodfight	NaN	0	NaN	...	
4	tt0249516	Foodfight	Foodfight!	2012	91	Action,Animation,Comedy	Foodfight	NaN	0	NaN	...	
...	...	...	...	...	...	...	...	...	...	...	...	
23917	tt9225192	Unstoppable	Seongnan hwangso	2018	116	Action,Crime	Unstoppable	WGUSA	101,000	NaN	...	
23918	tt9225192	Unstoppable	Seongnan hwangso	2018	116	Action,Crime	Unstoppable	WGUSA	101,000	NaN	...	
23919	tt9225192	Unstoppable	Seongnan hwangso	2018	116	Action,Crime	Unstoppable	WGUSA	101,000	NaN	...	
23920	tt9225192	Unstoppable	Seongnan hwangso	2018	116	Action,Crime	Unstoppable	WGUSA	101,000	NaN	...	
23921	tt9225192	Unstoppable	Seongnan hwangso	2018	116	Action,Crime	Unstoppable	WGUSA	101,000	NaN	...	

23922 rows × 25 columns

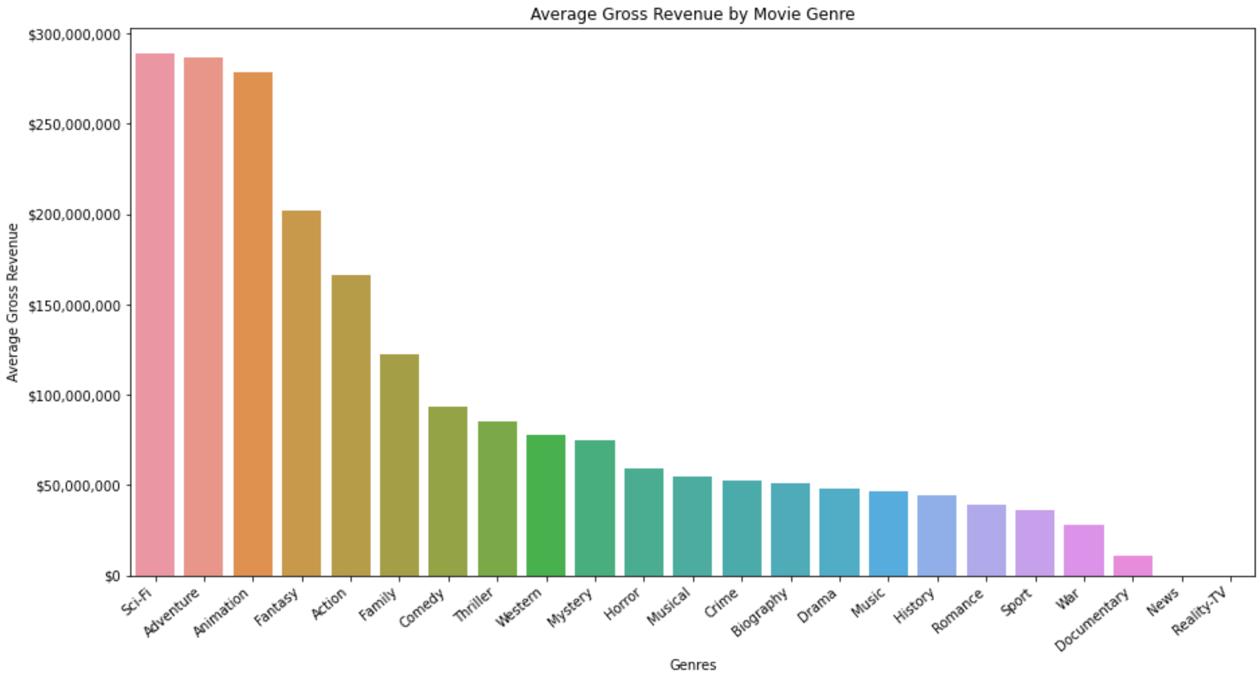
The following is a custom function for generating a bar chart.

In [64]:
```
# Barplot function

def bplot(var1, var2, title, xlabel, ylabel):
    fig, ax = plt.subplots(figsize = [15, 7.5])
    sns.barplot(x = var1, y = var2, data = chart_data, ax = ax)
    ax.yaxis.set_major_formatter('${x:,.0f}')
    ax.set_xticklabels(ax.get_xticklabels(), rotation=40, ha="right")
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.show()
```

V. Results

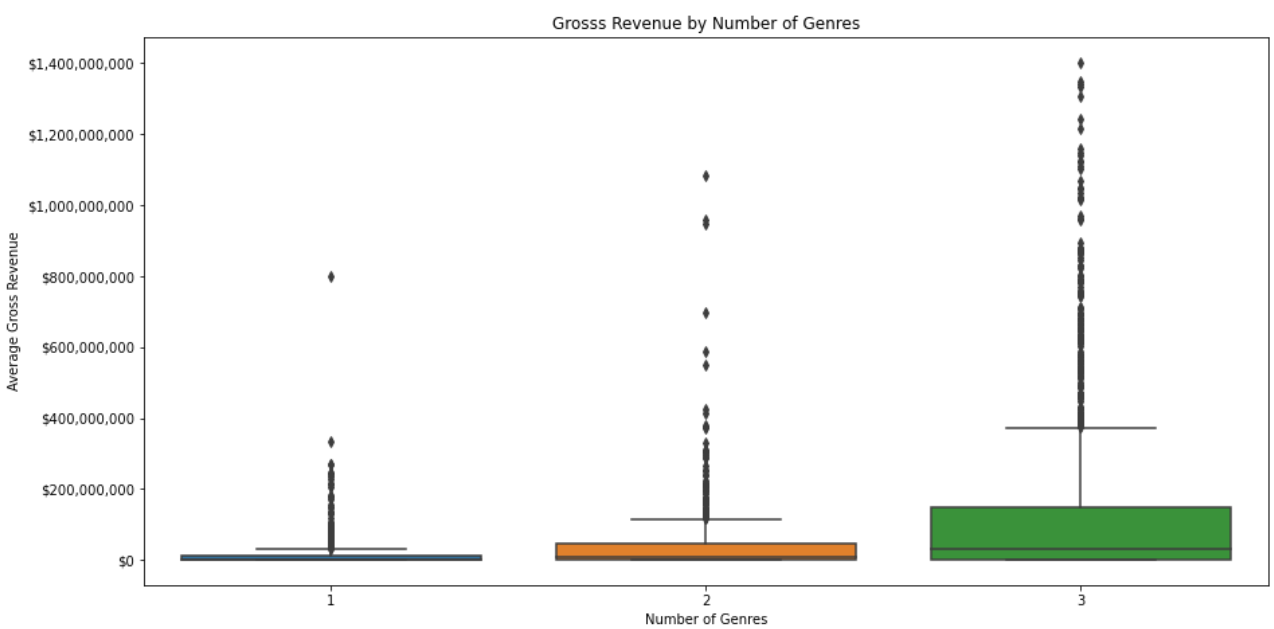
To begin with, below shows the average gross revenue by unique movie genre. We can see that Animation, Sci-Fi, and Adventure are the top 3 movies with the biggest revenue, close to $300M. Therefore, we want to pay attention to them.

In [65]:
```
# Calculate and plot average gross revenue by movie genre

chart_data = merged1.groupby("value")["total_gross"].agg({"size", "mean"}).sort_values(by = "mean", ascending = False)

bplot("value", "mean", "Average Gross Revenue by Movie Genre", "Genres", "Average Gross Revenue")
```



However, we already know that many movies are associated with multiple genres. Is a movie more likely to succeed when it has more unique genres? Yes, it is seen below that gross revenue tends to be bigger for multi-genre movies.

In [66]:
```
# Calculate and plot average gross revenue by number of genres

merged["num_genres"] = merged["genres"].str.count(",").fillna(0)+1

fig, ax = plt.subplots(figsize = [15, 7.5])
sns.boxplot(x = "num_genres", y = "total_gross", data = merged, ax = ax)
ax.yaxis.set_major_formatter('${x:,.0f}')
ax.xaxis.set_major_formatter('{x:,.0f}')
ax.set_title("Grosss Revenue by Number of Genres")
ax.set_xlabel("Number of Genres")
ax.set_xticklabels([1, 2, 3])
ax.set_ylabel("Average Gross Revenue")
plt.show()
```



More numerically, the average gross revenue for a single-genre movie is $26M, and average gross revenue for a movie with two genres is $47M. Lastly, the average gross revenue for multi-genre movies is much higher, reaching approximately $132M.

In [67]:
```
# Show average gross revenue by number of unique genres in a table

table_data = merged.groupby("num_genres")["total_gross"].mean().reset_index()
table_data.columns = ["Number of Genres", "Average Gross Revenue"]
table_data
```

Out[67]:

	Number of Genres	Average Gross Revenue
0	1	24,107,705
1	2	44,375,522
2	3	129,321,740

Therefore, it is advisable to invest in a movie that is associated with multiple genres. Then, which combination of genres is most lucrative? Below, we can see that "Adventure, Fantasy" yielded approximately $700M on average. However, there are only three data points. The next one with more data points is "Action, Adventure, Sci-Fi" with an average gross revenue of $566M. I propose that the firm create a movie in the genre.

In [68]:
```
# Show average gross revenue by combination of genres in a table

table_data = merged.groupby("genres")["total_gross"].agg({"size", "mean"}).sort_values(by = "mean", ascending = False)
table_data.columns = ["Genre(s)", "Number of Movies", "Average Gross Revenue"]
table_data
```

Out[68]:

	Genre(s)	Number of Movies	Average Gross Revenue
0	Adventure,Fantasy,Mystery	1	960,300,000
1	Adventure,Fantasy	3	700,555,074
2	Adventure,Drama,Sci-Fi	2	648,239,688
3	Action,Adventure,Sci-Fi	54	566,346,391
4	Action,Comedy,Mystery	1	544,100,000
5	Adventure,Drama,Fantasy	10	468,106,705
6	Action,Adventure,Fantasy	33	416,692,584
7	Adventure,Mystery,Sci-Fi	1	402,448,265
8	Biography,Drama,Musical	1	386,665,550
9	Adventure,Family,Fantasy	13	377,654,807
10	Action,Adventure,Thriller	18	370,897,668
11	Adventure,Animation,Comedy	76	353,066,180
12	Action,Adventure,Animation	21	352,529,711
13	Action,Mystery,Sci-Fi	1	348,300,000
14	Animation,Comedy,Family	6	347,942,098
15	Action,Fantasy,War	1	330,780,051
16	Action,Adventure,Comedy	33	303,314,301
17	Action,Sci-Fi	2	291,200,000
18	Action,Drama,Family	1	263,880,341
19	Action,Adventure,Family	6	255,506,507

It is consistently proven that the movie genre "Action, Adventure, Sci-Fi" results in substantial gross revenue. The table below shows that 25th percentile of the gross revenue is over $100M.

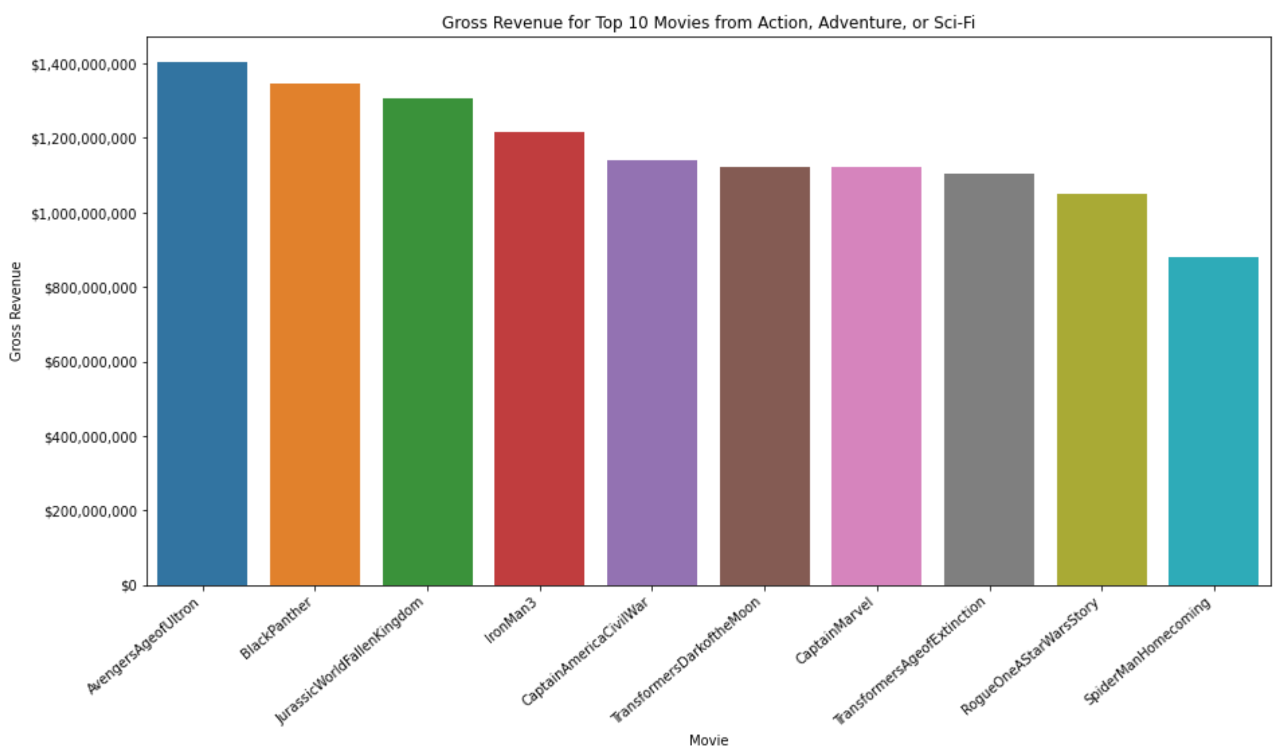
In [69]:
```
# Distribution of gross revenue for movies from Action, Adventure, or Sci-Fi

genres = ["Action", "Adventure", "Sci-Fi", "Action,Adventure", "Adventure,Sci-Fi", "Action,Adventure,Sci-Fi"]

table_data = merged[merged["genres"].isin(genres) == True]["total_gross"].describe().reset_index()
table_data.columns = ["Statistic", "Gross Revenue"]
table_data
```

Out[69]:

	Statistic	Gross Revenue
0	count	69
1	mean	445,512,731
2	std	395,546,376
3	min	0
4	25%	94,763,758
5	50%	375,700,000
6	75%	678,801,370
7	max	1,403,013,963

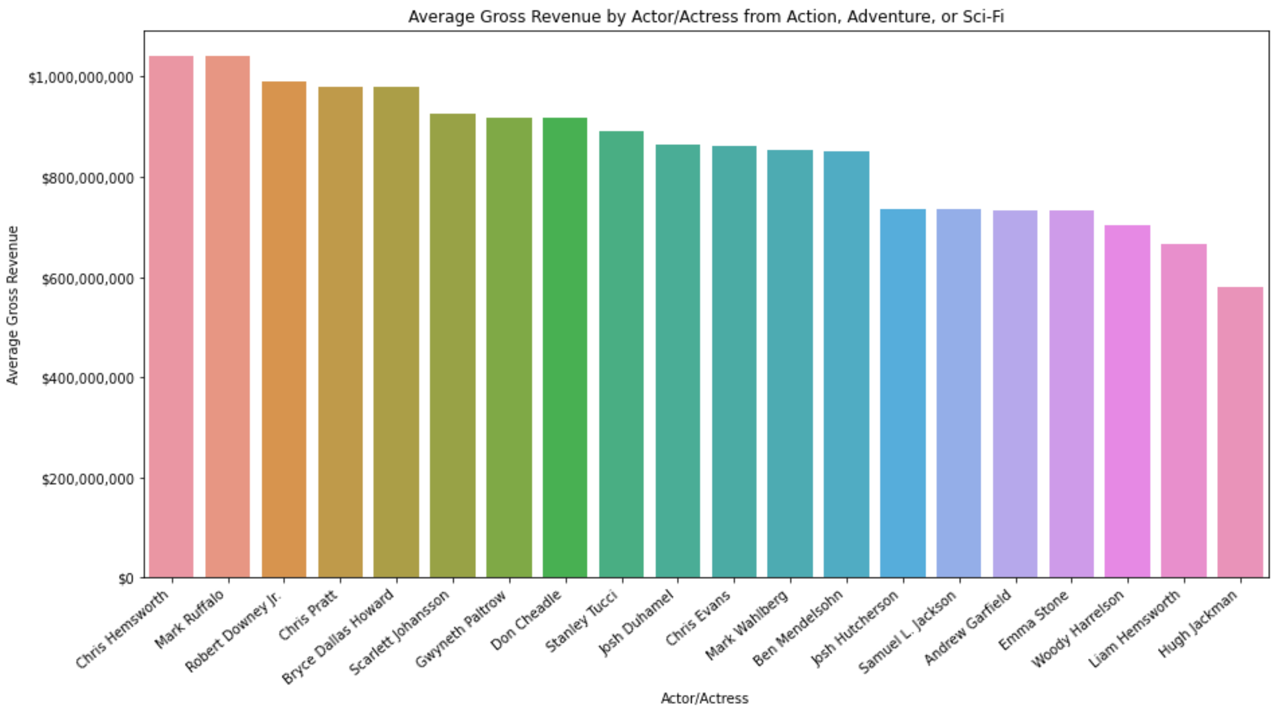
For reference, below are top 10 movies in that genre.

In [70]:
```
# Calculate and plot gross revenue for top 10 movies from Action, Adventure, or Sci-Fi

# merged["title_genres"] = merged["title"]+"\n("+merged["genres"]+")"
chart_data = merged[merged["genres"].isin(genres) == True].sort_values(by = "total_gross", ascending = False)[:10]

bplot("title", "total_gross", "Gross Revenue for Top 10 Movies from Action, Adventure, or Sci-Fi", "Movie", "Gross Reve
```



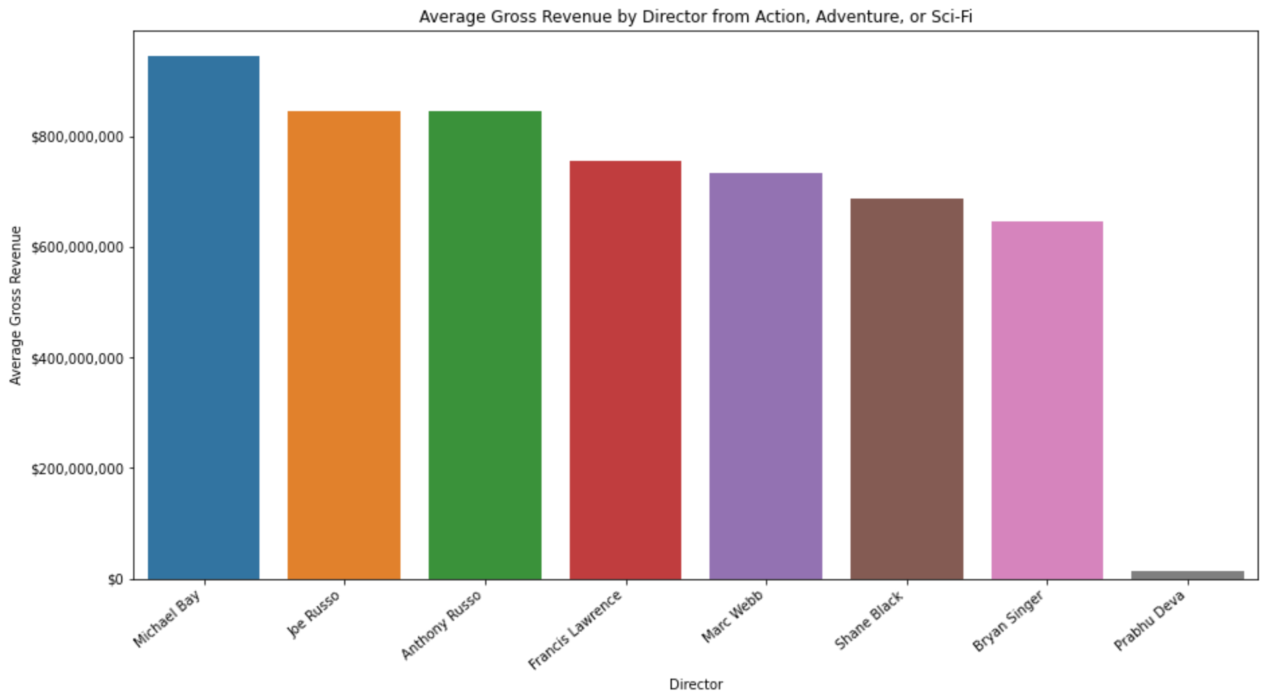
The next step is to identify which actors/actresses are associated with higher gross revenue. The chart below demonstrates that Chris Hemsworth ("Avengers: Age of Ultron", "Avengers: Infinity War", etc.), Mark Ruffalo ("Avengers: Age of Ultron", "Thor: Ragnarok", etc.), and Robert Downey Jr. ("Avengers: Age of Ultron", "Iron Man 3", etc.) are the top 3 actors. Here, I excluded actors/actresses who acted in only one movie with high gross revenue because it is unclear if the actor/actress contributed to the movie's success, or it was because of luck.

In [79]:
```
# Identify actors/actresses from Action, Adeventure, or Sci-Fi who acted in more than one movie
# Calculate and plot average gross revenue by actor/actress

merged2a = merged2[(merged2["category"].str.startswith("act") == True) & (merged2["genres"].isin(genres) == True)]
actors = merged2a.groupby("primary_name")["primary_name"].count()
actors_with_mult_movies = actors[actors>1].keys()
merged2a = merged2a[merged2a["primary_name"].isin(actors_with_mult_movies)]
chart_data = merged2a.groupby(["primary_name", "category"])["total_gross"].mean().sort_values(ascending = False)[:20].

bplot("primary_name", "total_gross", "Average Gross Revenue by Actor/Actress from Action, Adventure, or Sci-Fi", "Acto
```



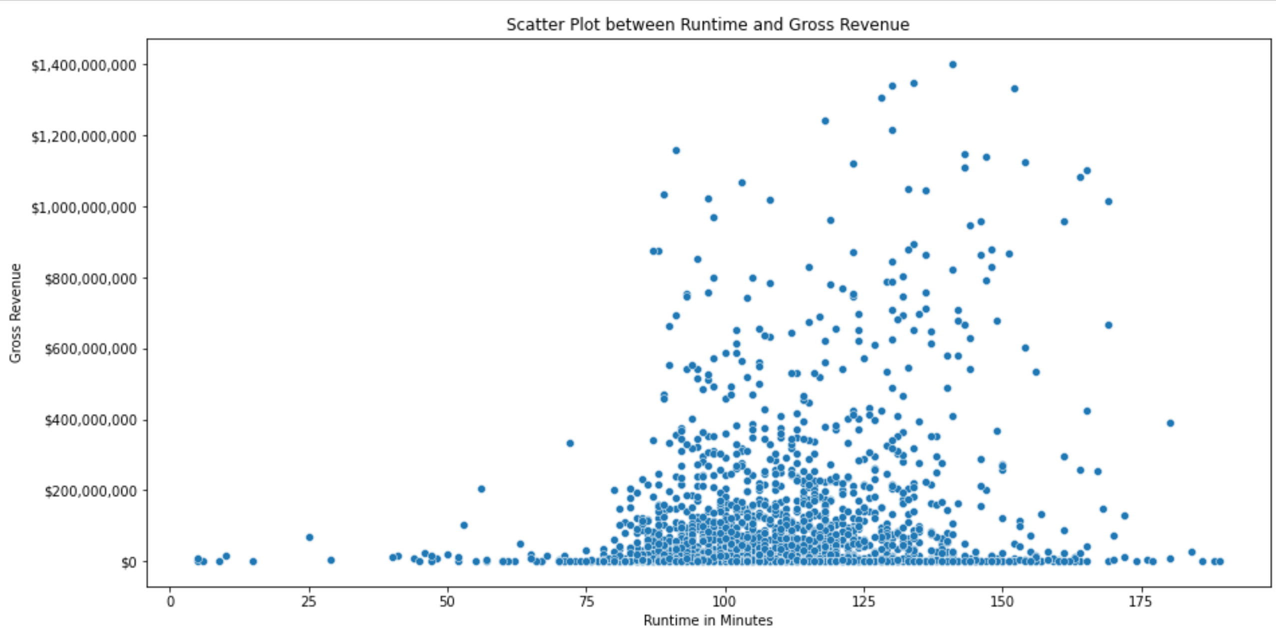
I also identified which directors are associated with higher gross revenue. The chart below demonstrates that Michael Bay ("Transformers: Dark of the Moon", "Transformers: Age of Extinction", etc.), Joe Russo ("Captain America: Civil War", "Captain America: The Winter Soldier", etc.), and Anthony Russo ("Captain America: Civil War", "Captain America: The Winter Soldier", etc.) are the top 3 directors. Here, I excluded directors who directed only one movie with high gross revenue because it is unclear if the director contributed to the movie's success, or it was because of luck.

In [73]:
```
# Identify directors from Action, Adeventure, or Sci-Fi who directed more than one movie
# Calculate and plot average gross revenue by director

merged2b = merged2[(merged2["category"].str.startswith("director") == True) & (merged2["genres"].isin(genres) == True)
directors = merged2b.groupby("primary_name")["primary_name"].count()
directors_with_mult_movies = directors[directors>1].keys()
merged2b = merged2b[merged2b["primary_name"].isin(directors_with_mult_movies)]
chart_data = merged2b.groupby(["primary_name", "category"])["total_gross"].mean().sort_values(ascending = False)[:30].

bplot("primary_name", "total_gross", "Average Gross Revenue by Director from Action, Adventure, or Sci-Fi", "Director"
```

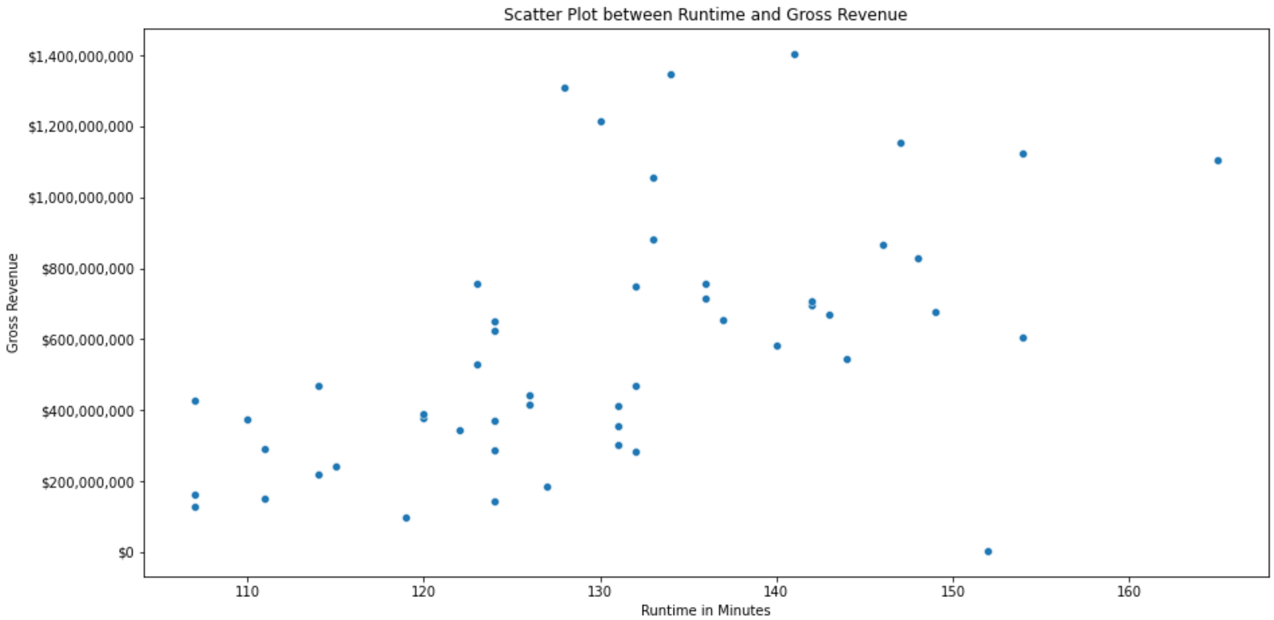


Lastly, I plotted the movie runtime and gross revenue in all genres. It can be generally seen that successful movies are not under 75 minutes and over 175 minutes. Therefore, the optimal range of runtime would be 75-175 minutes.

In [77]:
```
# Plot gross revenue by runtime

fig, ax = plt.subplots(figsize = [15, 7.5])
sns.scatterplot(x = "runtime_minutes", y = "total_gross", data = merged, ax = ax)
ax.yaxis.set_major_formatter('${x:,.0f}')
plt.title("Scatter Plot between Runtime and Gross Revenue")
plt.xlabel("Runtime in Minutes")
plt.ylabel("Gross Revenue")
plt.show()
```



In [98]:
```
# Plot gross revenue by runtime

fig, ax = plt.subplots(figsize = [15, 7.5])
sns.scatterplot(x = "runtime_minutes", y = "total_gross", data = merged[merged["genres"] == "Action,Adventure,Sci-Fi"]
ax.yaxis.set_major_formatter('${x:,.0f}')
plt.title("Scatter Plot between Runtime and Gross Revenue")
plt.xlabel("Runtime in Minutes")
plt.ylabel("Gross Revenue")
plt.show()
```



Final Proposal

Based on the findings, I propose the following three solutions for the new movie business:

1. Create a multi-genre movie, especially one in "Action, Adventure, and Sci-Fi"
2. Hire actors and directors generating higher gross revenue
    - Such actors include: Chris Hemsworth, Mark Ruffalo, Robert Downey Jr., etc.
    - Such directors include: Michael Bay, Joe Russo, Anthony Russo, etc.
3. Runtime should be between 120 to 150 minutes

Future Improvement Ideas

1. It was not readily verifiable if the genre "Action, Adventure, Sci-Fi" is produced by only a handful of studios or not. If this is the case, it could mean that a small number of studios with appropriate expertise can produce the successful movies, and the barrier entry is high for a new player like Microsoft. Therefore, additional research or a more reliable dataset in this regard can be useful for future improvements.
2. In identifying actors/actresses with high gross revenue, I identified them regardless of whether they are main characters or supporting characters. If we decide to proceed with the genre "Action, Adventure, Sci-Fi", it makese sense to refine the list based on their roles.