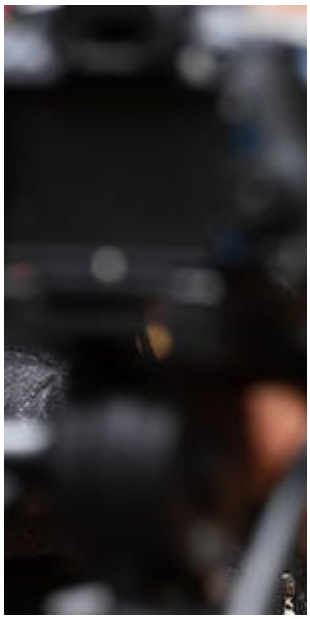
Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions



1 Project Title

Authors: Ryan Sajac

1.1 Overview

Microsoft is venturing into the exciting world of movi
risky investment. Microsoft needs to think about the
target the largest possible audience, or at least gene

Using the genre, runtime, budgets, and gross earnin
movie under the animation genre with a runtime betv
findings are detailed below.

1.2 Business Problem

**Microsoft must be most interested in two things,
task.**

For these two pain points, the questions we need to

1. What type of genres are most likely to be most p
2. What runtimes are most likely to produce higher
3. Which genre limits the risk on return of investme

Contents
1 Project Title
1.1 Overview
1.2 Business Prc
1.3 Data Unders
1.4 Data Prepara
1.5 Data Modelir
1.6 Evaluation
1.7 Conclusions

From a business perspective, these questions are im
either do that by taking a big risk and hitting it out of

1.3 Data Understanding

Target Variable - Profit

Microsoft's goal is to make money in this venture.

Data is from Rotten Tomatoes, and IMDB. These site
information to determine what are recommendations

I included the following files:

- imdb_title_basics_csv_gz - has genres, runtime
- bom_movie_gross_csv_gz - has the domestic ar
- tn.movie_budgets.csv.gz - has the budget, and t

I can join these three tables to get relationships amo
columns. The most relevant columns for our analysis

- Production Budget
- Runtime Minutes
- Worldwide Gross

I created two further columns:

- Profit
- Gross-to-cost-Ratio

While profit is our target variable, Gross-to-cost-ratic

With more time we could have narrowed down a sho

In [212]:

```
# Import standard packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('seaborn-talk')

%matplotlib inline
```

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [139]:
```
# Here we import glob to find the
import glob, os
fpath = 'zippedData/'
os.listdir(fpath)
```

Out[139]:
```
['imdb.title.crew.csv.gz',
 'tmdb.movies.csv.gz',
 'imdb.title.akas.csv.gz',
 'imdb.title.ratings.csv.gz',
 'imdb.name.basics.csv.gz',
 'rt.reviews.tsv.gz',
 'imdb.title.basics.csv.gz',
 'rt.movie_info.tsv.gz',
 'tn.movie_budgets.csv.gz',
 'bom.movie_gross.csv.gz',
 'imdb.title.principals.csv.gz']
```

In [140]:
```
# I made a list of all my files.
query = fpath+'*.gz'
file_list = glob.glob(query)
```

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [141]:

```
"""I looped through the list of fi
and display the head and tail so t
are most applicable for our analys

tables = {}
for file in file_list:
    print('---'*20)
    filename = file.replace('zippe
    print(filename)
    if '.tsv.gz' in file:
        temp_df = pd.read_csv(file
    else:
        temp_df = pd.read_csv(file
    display(temp_df.head(), temp_d

    tables[filename] = temp_df
```

```
------------------------------------------------------------
imdb_title_crew_csv_gz
```

	tconst	directors	
0	tt0285252	nm0899854	
1	tt0438973	NaN	nm0
2	tt0462036	nm1940585	
3	tt0835418	nm0151540	nm0
4	tt0878654	nm0089502,nm2291498,nm2292011	

	tconst	directors	wr
146139	tt8999974	nm10122357	nm1012
146140	tt9001390	nm6711477	nm671
146141	tt9001494	nm10123242,nm10123248	
146142	tt9004986	nm4993825	nm499
146143	tt9010172	NaN	nm835

```
------------------------------------------------------------
tmdb_movies_csv_gz
```

	Unnamed: 0	genre_ids	id	original_language
0	0	[12, 14, 10751]	12444	en
1	1	[14, 12, 16, 10751]	10191	en
2	2	[12, 28, 878]	10138	en
3	3	[16, 35, 10751]	862	en

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

	Unnamed: 0	genre_ids	id	original_language
4	4	[28, 878, 12]	27205	en

	Unnamed: 0	genre_ids	id	original_
26512	26512	[27, 18]	488143	
26513	26513	[18, 53]	485975	
26514	26514	[14, 28, 12]	381231	
26515	26515	[10751, 12, 28]	366854	
26516	26516	[53, 27]	309885	

---

imdb_title_akas_csv_gz

	title_id	ordering	
0	tt0369610	10	Джур
1	tt0369610	11	Jurashikl
2	tt0369610	12	Jurassic World: O Mundo dos Di
3	tt0369610	13	O Mundo dos Di
4	tt0369610	14	Jura

	title_id	ordering	title	regi
331698	tt9827784	2	Sayonara kuchibiru	Na
331699	tt9827784	3	Farewell Song	XW
331700	tt9880178	1	La atención	Na
331701	tt9880178	2	La atención	E
331702	tt9880178	3	The Attention	XW

---

imdb_title_ratings_csv_gz

	tconst	averagerating	numvotes
0	tt10356526	8.3	31
1	tt10384606	8.9	559
2	tt1042974	6.4	20
3	tt1043726	4.2	50352
4	tt1060240	6.5	21

	tconst	averagerating	numvotes

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

	tconst	averagerating	numvotes
73851	tt9805820	8.1	25
73852	tt9844256	7.5	24
73853	tt9851050	4.7	14
73854	tt9886934	7.0	5
73855	tt9894098	6.3	128

---

imdb_name_basics_csv_gz

	nconst	primary_name	birth_year	death_y
0	nm0061671	Mary Ellen Bauder	NaN	N
1	nm0061865	Joseph Bauer	NaN	N
2	nm0062070	Bruce Baum	NaN	N
3	nm0062195	Axel Baumann	NaN	N
4	nm0062798	Pete Baxter	NaN	N

	nconst	primary_name	birth_year	d
606643	nm9990381	Susan Grobes	NaN	
606644	nm9990690	Joo Yeon So	NaN	
606645	nm9991320	Madeline Smith	NaN	
606646	nm9991786	Michelle Modigliani	NaN	
606647	nm9993380	Pegasus Envoyé	NaN	

---

rt_reviews_tsv_gz

	id	review	ra
0	3	A distinctly gallows take on contemporary fina...	
1	3	It's an allegory in search of a meaning that n...	
2	3	... life lived in a bubble in financial dealin...	
3	3	Continuing along a line introduced in last yea...	
4	3	... a perverse twist on neorealism...	

	id	revie
54427	2000	The real charm of this trifle is the deadpan c
54428	2000	Na
54429	2000	Na
54430	2000	Na

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

	id	revie
54431	2000	Na

---

imdb_title_basics_csv_gz

	tconst	primary_title	
0	tt0063540	Sunghursh	
1	tt0066787	One Day Before the Rainy Season	
2	tt0069049	The Other Side of the Wind	The Oth
3	tt0069204	Sabse Bada Sukh	
4	tt0100275	The Wandering Soap Opera	La

	tconst	prim
146139	tt9916538	Kuambil La
146140	tt9916622	Rodolpho Teóphilo - O Legado de um
146141	tt9916706	Dankyav
146142	tt9916730	
146143	tt9916754	Chico Albuquerque - Re

---

rt_movie_info_tsv_gz

	id	synopsis	rating	
0	1	This gritty, fast-paced, and innovative police...	R	Actic Adventure|Classics|I
1	3	New York City, not-too-distant-future: Eric Pa...	R	Drama|Science I and F
2	5	Illeana Douglas delivers a superb performance ...	R	Drama|Music Performir
3	6	Michael Douglas runs afoul of a treacherous su...	R	Drama|Myste Sus
4	7	NaN	NR	Drama|Ror

	id	synopsis	rating	
1555	1996	Forget terrorists or hijackers -- there's a ha...	R	Adventure|Horror

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

	id	synopsis	rating	
1556	1997	The popular Saturday Night Live sketch was exp...	PG	Comedy|Scienc
1557	1998	Based on a novel by Richard Powell, when the l...	G	Classics|Comedy|D and Pe
1558	1999	The Sandlot is a coming-of-age story about a g...	PG	Comedy|Dra Family|Sport
1559	2000	Suspended from the force, Paris cop Hubert is ...	R	Action and Adventu and

---

`tn_movie_budgets_csv_gz`

	id	release_date	
0	1	Dec 18, 2009	
1	2	May 20, 2011	Pirates of the Caribbean: On Strange
2	3	Jun 7, 2019	Dark F
3	4	May 1, 2015	Avengers: Age o
4	5	Dec 15, 2017	Star Wars Ep. VIII: The La

	id	release_date	movie
5777	78	Dec 31, 2018	Red 11
5778	79	Apr 2, 1999	Following
5779	80	Jul 13, 2005	Return to the Land of Wonders
5780	81	Sep 29, 2015	A Plague So Pleasant
5781	82	Aug 5, 2005	My Date With Drew

`bom_movie_gross_csv_gz`

	title	studio	d
0	Toy Story 3	BV	
1	Alice in Wonderland (2010)	BV	
2	Harry Potter and the Deathly Hallows Part 1	WB	
3	Inception	WB	
4	Shrek Forever After	P/DW	

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

	title	studio	domestic_
3382	The Quake	Magn.	6
3383	Edward II (2018 re-release)	FM	4
3384	El Pacto	Sony	2
3385	The Swan	Synergetic	2
3386	An Actor Prepares	Grav.	1

---

imdb_title_principals_csv_gz

	tconst	ordering	nconst	category	j
0	tt0111414	1	nm0246005	actor	Na
1	tt0111414	2	nm0398271	director	Na
2	tt0111414	3	nm3739909	producer	produc
3	tt0323808	10	nm0059247	editor	Na
4	tt0323808	1	nm3579312	actress	Na

	tconst	ordering	nconst	category
1028181	tt9692684	1	nm0186469	actor
1028182	tt9692684	2	nm4929530	self
1028183	tt9692684	3	nm10441594	director
1028184	tt9692684	4	nm6009913	writer
1028185	tt9692684	5	nm10441595	producer

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [142]:

```
"""We will combine three datafram
We drop duplicates in our table wi
acknowledging that there are some
the same domestic gross."""

basics = tables['imdb_title_basics
gross = tables['bom_movie_gross_cs
budgets = tables['tn_movie_budgets

tempcombined = pd.merge(gross, bas
                        right_on =
tempcombined
```

Out[142]:

	title	studio	domestic_gross	foreign_gr
0	Toy Story 3	BV	415000000.0	652000
1	Inception	WB	292600000.0	535700
2	Shrek Forever After	P/DW	238700000.0	513900
3	The Twilight Saga: Eclipse	Sum.	300500000.0	398000
4	Iron Man 2	Par.	312400000.0	311500
...	...	...	...	
1661	The House That Jack Built	IFC	88000.0	
1662	Helicopter Eela	Eros	72000.0	
1663	Oolong Courtyard	CL	37700.0	
1664	The Workshop	Strand	22100.0	
1665	An Actor Prepares	Grav.	1700.0	

1666 rows × 11 columns

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [143]:

```
#The size of our movie info is 10(
#information about that movie.

movie_info_temp = pd.merge(tempcon
                           right_on = '

movie_info = movie_info_temp.drop_
                                'origi
movie_info.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1001 entries, 0 to 1063
Data columns (total 17 columns):
 #   Column             Non-Null Count
---  ------             --------------
 0   title              1001 non-null
 1   studio             1001 non-null
 2   domestic_gross_x   1000 non-null
 3   foreign_gross      901 non-null
 4   year               1001 non-null
 5   tconst             1001 non-null
 6   primary_title      1001 non-null
 7   original_title     1001 non-null
 8   start_year         1001 non-null
 9   runtime_minutes    1000 non-null
 10  genres             1001 non-null
 11  id                 1001 non-null
 12  release_date       1001 non-null
 13  movie              1001 non-null
 14  production_budget  1001 non-null
 15  domestic_gross_y   1001 non-null
 16  worldwide_gross    1001 non-null
dtypes: float64(2), int64(3), object(12
memory usage: 140.8+ KB
```

1.4 Data Preparation

Describe and justify the process for preparing the da

Questions to consider:

- Were there variables you dropped or created?
- How did you address missing values or outliers?
- Why are these choices appropriate given the dat

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [144]:

```
"""From our above info of the movi
domestic_gross_x, foreign gross ar
those inconsistencies here."""

for col in movie_info.columns:
    print(col, '\n', movie_info[co
          '\n\n')

```

```
title
 How to Train Your Dragon 2        0.000
Shame                              0.0009
Mandela: Long Walk to Freedom      0.0009
Little Boy                         0.0009
Dear White People                  0.0009
Name: title, dtype: float64


studio
 Uni.     0.105894
Fox       0.101898
WB        0.093906
BV        0.066933
Par.      0.065934
Name: studio, dtype: float64


domestic_gross_x
```

In [145]:

```
"""I notice that we have a missing
data since it took less than 2 mir
Down' is 90 minutes. """

movie_info[movie_info['runtime_mir
```

Out[145]:

	title	studio	domestic_gross_x	foreign_gross
509	Upside Down	MNE	105000.0	800000

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [146]:

```
1 movie_info['runtime_minutes'].iloc
2
```

```
/Users/ryansajac/opt/anaconda3/envs/lea
yWarning:
A value is trying to be set on a copy o

See the caveats in the documentation: h
-a-view-versus-a-copy (https://pandas.p
a-copy)
  iloc._setitem_with_indexer(indexer, v
<ipython-input-146-c74b2853e7c5>:1: Set
A value is trying to be set on a copy o

See the caveats in the documentation: h
-a-view-versus-a-copy (https://pandas.p
a-copy)
  movie_info['runtime_minutes'].iloc[48
```

In [147]:

```
1 """Here we need to write a functic
2 in production_budget, domestic_gro
3 to integers."""
4
5 def dollartonum(col, df):
6     df[col] = df[col].map(lambda x
7     return df[col]
```

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [148]:

```
"""Convert the three dollar amount
to see that it is correct. """
dollartonum('production_budget', m
dollartonum('domestic_gross_y', mo
dollartonum('worldwide_gross', mov
movie_info

```

```
<ipython-input-147-6e2934661e56>:6: Set
A value is trying to be set on a copy o
Try using .loc[row_indexer,col_indexer]

See the caveats in the documentation: h
-a-view-versus-a-copy (https://pandas.p
a-copy)
  df[col] = df[col].map(lambda x: int(x
```

Out[148]:

	title	studio	domestic_gross_x	foreig
0	Toy Story 3	BV	415000000.0	65
1	Inception	WB	292600000.0	53
2	Shrek Forever After	P/DW	238700000.0	51
3	The Twilight Saga: Eclipse	Sum.	300500000.0	39
4	Iron Man 2	Par.	312400000.0	31
...	...	...	...	
1059	Suspiria	Amazon	2500000.0	
1060	The Hurricane Heist	ENTMP	6100000.0	
1061	Destroyer	Annapurna	1500000.0	
1062	Gotti	VE	4300000.0	
1063	Mandy	RLJ	1200000.0	

1001 rows × 17 columns

In [149]:

```
"""Get rid of Eden which has no fo
it represents less than a 1/1000 o
movie_info[movie_info['worldwide_g

```

Out[149]:

	title	studio	domestic_gross_x	foreign_gross
756	Eden	BG	65500.0	NaN

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [150]:

```
#Dropped Eden and checked that ou
movie_info.drop(index = 756, inpla
movie_info[710:800]

```

```
/Users/ryansajac/opt/anaconda3/envs/lea
arning:
A value is trying to be set on a copy o

See the caveats in the documentation: h
-a-view-versus-a-copy (https://pandas.p
a-copy)
  return super().drop(
```

Out[150]:

	title	studio	domestic_gross_x	foreign_
754	Freeheld	LGF	573000.0	
755	Maggie	RAtt.	187000.0	
757	Knock Knock	LGP	36300.0	
758	Strangerland	Alc	17500.0	
759	Captain America: Civil War	BV	408100000.0	7452
...	...	...	...	
844	Free State of Jones	STX	20800000.0	42
845	Middle School: The Worst Years of My Life	LGF	20000000.0	33
846	Triple 9	ORF	12600000.0	105
847	Whiskey Tango Foxtrot	Par.	23100000.0	
848	Fifty Shades of Black	ORF	11700000.0	105

90 rows × 17 columns

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [151]:

```
1 #Making a list out of the genres s
2 movie_info['genres'] = movie_info[
3
```

```
<ipython-input-151-15810ef16470>:2: Set
A value is trying to be set on a copy o
Try using .loc[row_indexer,col_indexer]

See the caveats in the documentation: h
-a-view-versus-a-copy (https://pandas.p
a-copy)
  movie_info['genres'] = movie_info['ge
```

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [152]:

```
#Adding a column to get the gross
movie_info['gross_to_cost_ratio']
movie_info['profit'] = movie_info[
movie_info
```

```
<ipython-input-152-96034e63426c>:2: Set
A value is trying to be set on a copy o
Try using .loc[row_indexer,col_indexer]

See the caveats in the documentation: h
-a-view-versus-a-copy (https://pandas.p
a-copy)
  movie_info['gross_to_cost_ratio'] = m
<ipython-input-152-96034e63426c>:3: Set
A value is trying to be set on a copy o
Try using .loc[row_indexer,col_indexer]

See the caveats in the documentation: h
-a-view-versus-a-copy (https://pandas.p
a-copy)
  movie_info['profit'] = movie_info['wo
```

Out[152]:

	title	studio	domestic_gross_x	foreig
0	Toy Story 3	BV	415000000.0	65
1	Inception	WB	292600000.0	53
2	Shrek Forever After	P/DW	238700000.0	51
3	The Twilight Saga: Eclipse	Sum.	300500000.0	39
4	Iron Man 2	Par.	312400000.0	31
...	...	...	...	
1059	Suspiria	Amazon	2500000.0	
1060	The Hurricane Heist	ENTMP	6100000.0	
1061	Destroyer	Annapurna	1500000.0	
1062	Gotti	VE	4300000.0	

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

	title	studio	domestic_gross_x	foreig
1063	Mandy	RLJ	1200000.0	

1000 rows × 19 columns

In [153]:
```
# Exploding so that I can separate
exploded_movies = movie_info.explo
```

In [154]:
```
"""This is the crux of most of my
higher medians of profit. I sorted
also sort by profit. """
exploded_movies.groupby('genres').
```

Out[154]:

genres	domestic_gross_x	year	start_year
War	3800000.0	2012.0	2012.0
Documentary	6100000.0	2012.0	2012.0
Romance	25900000.0	2012.0	2012.0
Sport	30100000.0	2014.0	2014.0
Drama	25250000.0	2014.0	2014.0
Biography	27300000.0	2015.0	2015.0
Music	28650000.0	2013.5	2013.5
Crime	29700000.0	2014.0	2014.0
Musical	38500000.0	2013.0	2013.0
Horror	31850000.0	2014.0	2014.0
Thriller	35400000.0	2013.0	2013.0
History	45350000.0	2015.0	2015.0
Comedy	45300000.0	2013.0	2013.0
Mystery	32750000.0	2014.0	2014.0
Family	54900000.0	2012.0	2012.0
Fantasy	42600000.0	2013.5	2013.5
Western	47500000.0	2011.5	2011.5
Action	56300000.0	2014.0	2014.0
Sci-Fi	90600000.0	2014.0	2014.0
Adventure	94100000.0	2014.0	2014.0
Animation	126849999.5	2014.0	2014.0

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [155]:

```
"""Clean up exploded_movies so onl
analysis."""
exploded_movies.drop(['title', 'st
                      'primary_tit
                      'start_year'
                      'domestic_gr
                     axis=1, inpl
exploded_movies
```

Out[155]:

	runtime_minutes	genres	production_budget
0	103.0	Adventure	200000000
0	103.0	Animation	200000000
0	103.0	Comedy	200000000
1	148.0	Action	160000000
1	148.0	Adventure	160000000
...	...	...	...
1062	112.0	Crime	10000000
1062	112.0	Drama	10000000
1063	121.0	Action	6000000
1063	121.0	Fantasy	6000000
1063	121.0	Horror	6000000

2621 rows × 7 columns

1.5 Data Modeling

Describe and justify the process for analyzing or mod

Questions to consider:

- How did you analyze or model the data?
- How did you iterate on your initial approach to m
- Why are these choices appropriate given the dat

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [156]:

```
# Here you run your code to model
median_by_genre = exploded_movies.
    'profit')
median_by_genre
```

Out[156]:

genres	runtime_minutes	production_budget	c
Western	116.0	38500000.0	
War	115.0	19500000.0	
Documentary	92.0	12000000.0	
Sport	117.0	25000000.0	
Drama	112.0	20000000.0	
Romance	104.0	19000000.0	
Music	108.0	17750000.0	
Biography	120.0	20000000.0	
Crime	109.0	30000000.0	
Horror	95.5	10000000.0	
Musical	123.0	70000000.0	
History	126.0	26500000.0	
Thriller	106.0	26000000.0	
Fantasy	110.5	62500000.0	
Family	103.0	45000000.0	
Comedy	102.0	30000000.0	
Mystery	103.0	13250000.0	
Action	112.0	63000000.0	
Adventure	107.0	110000000.0	
Sci-Fi	114.0	95000000.0	
Animation	95.0	99500000.0	

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [157]:

```
"""After running the count below,
Westerns, War, and Documentary to
We can note that explains the disc
and median profit of Westerns. We
data points. In the next cell we r

exploded_movies.groupby('genres').
```

Out[157]:

genres	runtime_minutes	production_budget	c
Musical	3	3	
Western	6	6	
War	8	8	
Documentary	9	9	
Sport	23	23	
Music	30	30	
History	30	30	
Family	65	65	
Mystery	76	76	
Animation	82	82	
Fantasy	86	86	
Sci-Fi	94	94	
Horror	100	100	
Biography	101	101	
Romance	139	139	
Crime	154	154	
Thriller	169	169	
Adventure	268	268	
Action	313	313	
Comedy	378	378	
Drama	487	487	

In [158]:

```
genre_index = exploded_movies.set_
genre_index = genre_index.drop(['I
```

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [159]:
```
#Check our genre_index dataframe.
genre_index.groupby('genres').cour
```

Out[159]:

genres	runtime_minutes	production_budget	dom
Sport	23	23	
History	30	30	
Music	30	30	
Family	65	65	
Mystery	76	76	
Animation	82	82	
Fantasy	86	86	
Sci-Fi	94	94	
Horror	100	100	
Biography	101	101	
Romance	139	139	
Crime	154	154	
Thriller	169	169	
Adventure	268	268	
Action	313	313	
Comedy	378	378	
Drama	487	487	

In [160]:
```

genre_index.describe()
```

Out[160]:

	runtime_minutes	production_budget	domesti
count	2595.000000	2.595000e+03	2.59
mean	110.109056	5.592611e+07	7.24
std	17.523441	6.060162e+07	9.21
min	63.000000	5.000000e+04	0.00
25%	97.000000	1.400000e+07	1.53
50%	107.000000	3.400000e+07	4.10
75%	120.000000	7.500000e+07	9.08
max	180.000000	4.106000e+08	7.00

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [161]:
```
1 genre_index.groupby('genres').desc
```

Out[161]:

	runtime_minutes				
	count	mean	std	min	25%
genres					
Action	313.0	114.488818	17.968831	81.0	101
Adventure	268.0	111.619403	19.141615	63.0	96
Animation	82.0	95.000000	7.903742	63.0	90
Biography	101.0	119.366337	16.396172	85.0	106
Comedy	378.0	103.111111	12.925755	63.0	94
Crime	154.0	111.363636	15.613308	83.0	101
Drama	487.0	114.028747	17.435218	81.0	102
Family	65.0	104.338462	14.890514	81.0	93
Fantasy	86.0	112.488372	18.100857	81.0	99
History	30.0	124.800000	15.938243	100.0	107
Horror	100.0	98.590000	12.856704	80.0	89
Music	30.0	111.666667	15.322585	93.0	104
Mystery	76.0	106.986842	20.084152	80.0	92
Romance	139.0	106.136691	14.486976	83.0	97
Sci-Fi	94.0	117.851064	20.426084	83.0	102
Sport	23.0	120.608696	16.191297	96.0	110
Thriller	169.0	108.491124	16.797859	80.0	95

17 rows × 48 columns

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [162]:

```
"""Here we get the mean, median, a
at a bar graph of these values."""
median_by_genre = genre_index.grou
    'worldwide_gross')
max_by_genre = genre_index.groupby
    'worldwide_gross')
min_by_genre = genre_index.groupby
    'worldwide_gross')

median_by_genre
```

Out[162]:

genres	runtime_minutes	production_budget	dom
Romance	104.0	19000000.0	
Sport	117.0	25000000.0	
Drama	112.0	20000000.0	
Biography	120.0	20000000.0	
Music	108.0	17750000.0	
Crime	109.0	30000000.0	
Horror	95.5	10000000.0	
Thriller	106.0	26000000.0	
History	126.0	26500000.0	
Comedy	102.0	30000000.0	
Mystery	103.0	13250000.0	
Family	103.0	45000000.0	
Fantasy	110.5	62500000.0	
Action	112.0	63000000.0	
Sci-Fi	114.0	95000000.0	
Adventure	107.0	110000000.0	
Animation	95.0	99500000.0	

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [163]:
```
1 max_by_genre
```

Out[163]:

genres	runtime_minutes	production_budget	dom
Sport	170.0	65000000	
History	150.0	156000000	
Romance	176.0	106000000	
Mystery	172.0	185000000	
Horror	152.0	190000000	
Biography	180.0	135000000	
Music	160.0	55000000	
Drama	180.0	185000000	
Family	169.0	250000000	
Fantasy	169.0	410600000	
Thriller	164.0	300000000	
Crime	180.0	250000000	
Comedy	163.0	260000000	
Animation	118.0	260000000	
Adventure	169.0	410600000	
Sci-Fi	169.0	330600000	
Action	172.0	410600000	

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [164]:
```
1 min_by_genre
```

Out[164]:

genres	runtime_minutes	production_budget	dom
Action	81.0	3000000	
Adventure	63.0	1800000	
Romance	83.0	50000	
Comedy	63.0	50000	
Drama	81.0	50000	
Horror	80.0	100000	
Thriller	80.0	100000	
Fantasy	81.0	100000	
Mystery	80.0	100000	
Music	93.0	1000000	
Biography	85.0	270000	
Crime	83.0	270000	
Sci-Fi	83.0	175000	
Sport	96.0	3500000	
Family	81.0	3000000	
Animation	63.0	8000000	
History	100.0	6000000	

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [165]:

```
"""To make conclusions for Microsc
and profit at three different leve
This way we can determine the 'sai
most likely profitable. We can als
be a high risk high reward. """



#MedbygenGtC = median_by_genre.plc
                    # xlabel = 'Wc
                     #title = 'Mec
#plt.savefig('./images/MedbygenGtC
#plt.close('MedbygenGtC.png')

"""While Mystery looks good here,
Mysteries move back toward the mic
```

Out[165]:

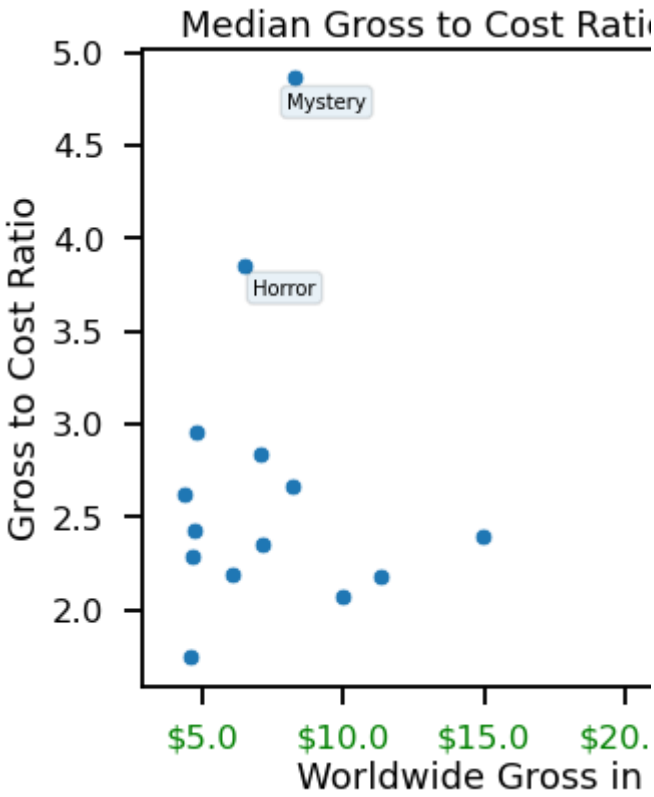
```
'While Mystery looks good here, when we
e pack. '
```

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [255]:

```
fig, ax = plt.subplots(figsize = (
MedbygenGtC = sns.scatterplot(x =
          y = median_by_genre['gr
ax.set(title = "Median Gross to Co
       xlabel ='Worldwide Gross ir
       ylabel = 'Gross to Cost Rat
fig.tight_layout();

ax.annotate("Mystery", xy = (8,4.7
            bbox=dict(boxstyle="rc
ax.annotate("Horror", xy = (6.8,3.
            bbox=dict(boxstyle="rc
ax.annotate("Sci-Fi", xy = (26, 3.
            bbox=dict(boxstyle="rc
ax.annotate("Animation", xy = (32,
            bbox=dict(boxstyle="rc
            color = 'red');

ax.xaxis.set_major_formatter('${x}

ax.xaxis.set_tick_params(which='ma
plt.savefig('./images/MedbygenGtC.
```



In [167]:

```
median_by_genre.index
```

Out[167]:

```
Index(['Romance', 'Sport', 'Drama', 'Bi
       'Thriller', 'History', 'Comedy',
       'Action', 'Sci-Fi', 'Adventure',
      dtype='object', name='genres')
```

Contents

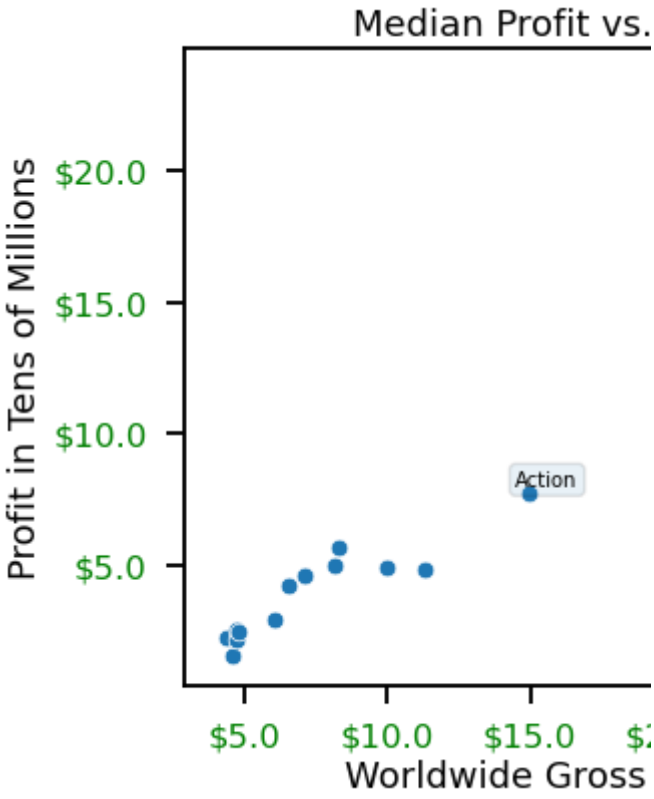
- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [168]:

```
#medbygenprof = median_by_genre.pl
 #                      xlabel = 'Wo
  #                     title = 'Mec
#plt.savefig('./images/Medbygenpro
#plt.close('Medbygenprof.png')
#"""We can start to conclude that
#the top 5 of both graphs."""
```

In [254]:

```
fig, ax = plt.subplots(figsize = (
medbygenprof = sns.scatterplot(x =
           y = median_by_genre['pr
ax.set(title = "Median Profit vs.
       xlabel ='Worldwide Gross in
       ylabel = 'Profit in Tens of
fig.tight_layout();

ax.annotate("Action", xy = (14.5,8
            bbox=dict(boxstyle="ro
ax.annotate("Sci-Fi", xy = (27,14.
            bbox=dict(boxstyle="ro
ax.annotate("Adventure", xy = (27,
            bbox=dict(boxstyle="ro
ax.annotate("Animation", xy = (32,
            bbox=dict(boxstyle="ro
            color = 'red');
ax.yaxis.set_major_formatter('${x}
ax.xaxis.set_major_formatter('${x}
ax.yaxis.set_tick_params(which='ma
ax.xaxis.set_tick_params(which='ma
plt.savefig('./images/medbygenprof
```

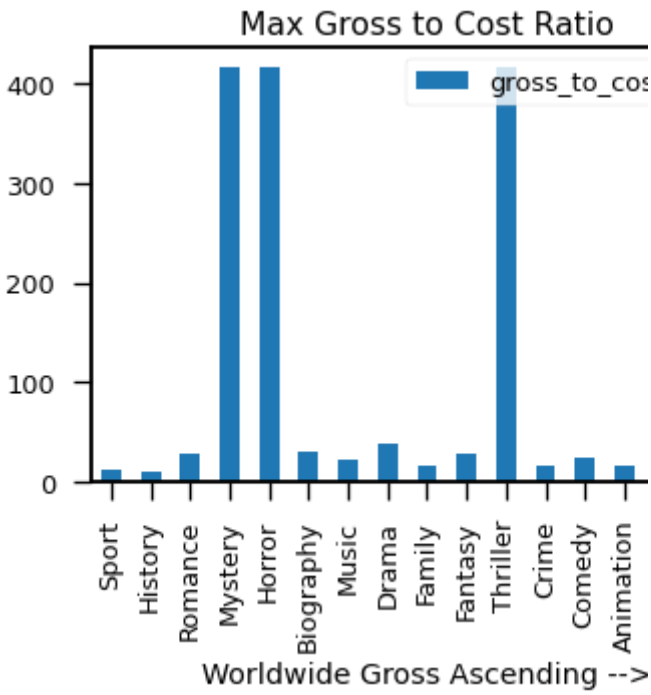


Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [215]:
```
max_by_genre.plot(y = 'gross_to_co
                  xlabel = 'Wor
                  title = 'Max
plt.savefig('./images/MaxbygenGtC.
plt.close('MaxbygenGtC.png')
"""Outliers in the Mystery, Horror
gross-to-cost-ratio. These are all
```

Out[215]:
```
'Outliers in the Mystery, Horror, and T
t of one movie, explored below.'
```



In [171]:
```
movie_info[movie_info['gross_to_co

"""Seeing as how the Gallows is th
_cost_ratio, we will throw away th
missing a zero, causing a factor o
```

Out[171]:
```
'Seeing as how the Gallows is the only
lier. Likely the production budget is\n
```

Contents

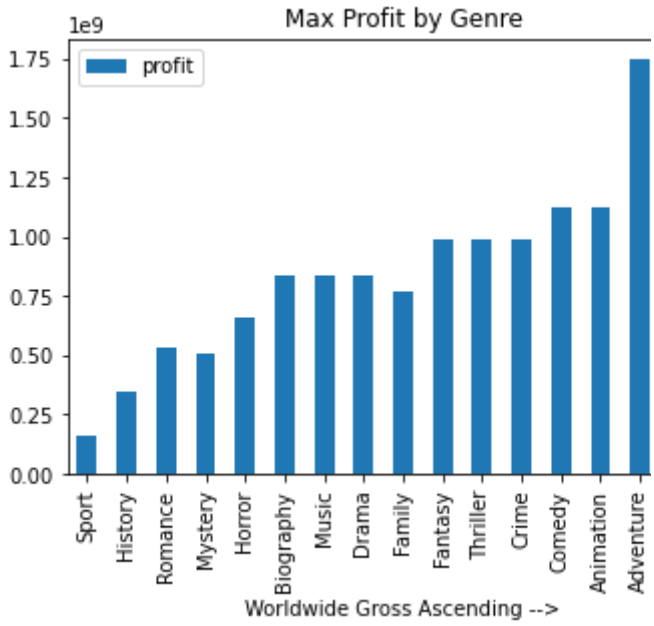
- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [172]:

```
max_by_genre.plot(y = 'profit', ki
                  xlabel = 'Wor
                  title = 'Max
plt.savefig('./images/Maxbygenprof
plt.close('Maxbygenprof.png')
"""Again we see Sci-fi, Adventure,
same height because the movie with
in this case the max profit was ar
```

Out[172]:

```
'Again we see Sci-fi, Adventure, Animat
max will have more than one genre tag\n
```



Contents

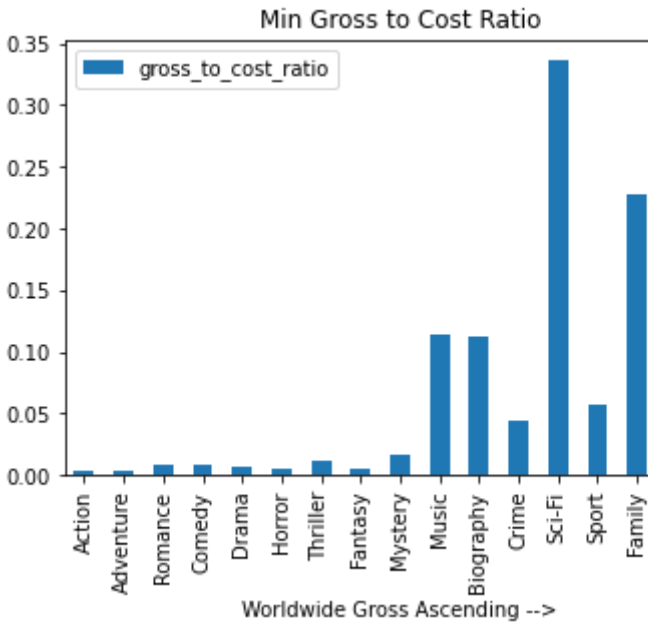
- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [173]:

```
min_by_genre.plot(y = 'gross_to_cc
                  xlabel = 'Wor
                  title = 'Min
plt.savefig('./images/MinbygenGtC.
plt.close('MinbygenGtC.png')
"""This graph tells us about what
genre. While anything under a rati
that some flops are worse than oth
in the event of a flop, are only g
our data. """
```

Out[173]:

```
"This graph tells us about what would h
means we didn't profit, we can see \nth
in the event of a flop, are only going
```

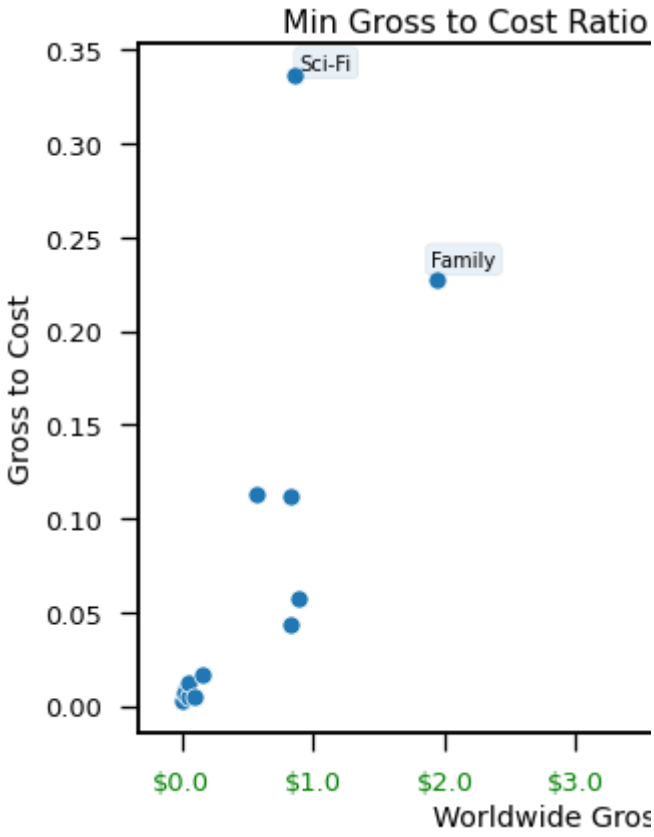


Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [225]:

```
fig, ax = plt.subplots(figsize = (
minbygenregtc = sns.scatterplot(x
           y = min_by_genre['gross
ax.set(title = "Min Gross to Cost
       xlabel ='Worldwide Gross in
       ylabel = 'Gross to Cost');
fig.tight_layout();

#ax.annotate("Action", xy = (14.5,
 #           bbox=dict(boxstyle="r
ax.annotate("Sci-Fi", xy = (.9,.34
            bbox=dict(boxstyle="ro
ax.annotate("Family", xy = (1.9, .
            bbox=dict(boxstyle="ro
ax.annotate("Animation", xy = (5.7
           bbox=dict(boxstyle="rou
            color = 'red');
#ax.yaxis.set_major_formatter('${x
ax.xaxis.set_major_formatter('${x}
#ax.yaxis.set_tick_params(which='r
ax.xaxis.set_tick_params(which='ma
plt.savefig('./images/minbygenregt
```



Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [226]:

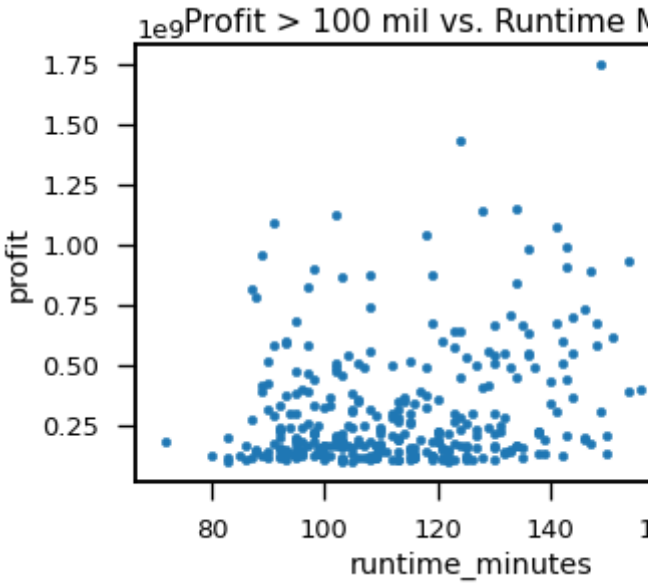
```
prof1 = movie_info[movie_info['prc
                           y = 'profi
                           title = 'I
plt.savefig('./images/Prof1.png')
plt.close('Prof1.png')

prof2 = movie_info[movie_info['prc
                           y = 'profi
                           title = 'I
plt.savefig('./images/Prof2.png')
plt.close('Prof2.png')

prof3 = movie_info[movie_info['prc
                           y = 'profi
                           title = 'I
plt.savefig('./images/Prof3.png')
plt.close('Prof3.png')

prof4 = movie_info[movie_info['prc
                           y = 'profi
                           title = 'I
plt.savefig('./images/Prof4.png')
plt.close('Prof4.png')
prof5 = movie_info[movie_info['prc
                           y = 'profi
                           title = 'I

plt.savefig('./images/Prof5.png')
plt.close('Prof5.png')

"""I wanted to compare movies with
any significant difference in the
determine that most movies that ge
150 minutes, but also most movies
also fall in that time frame. """
```

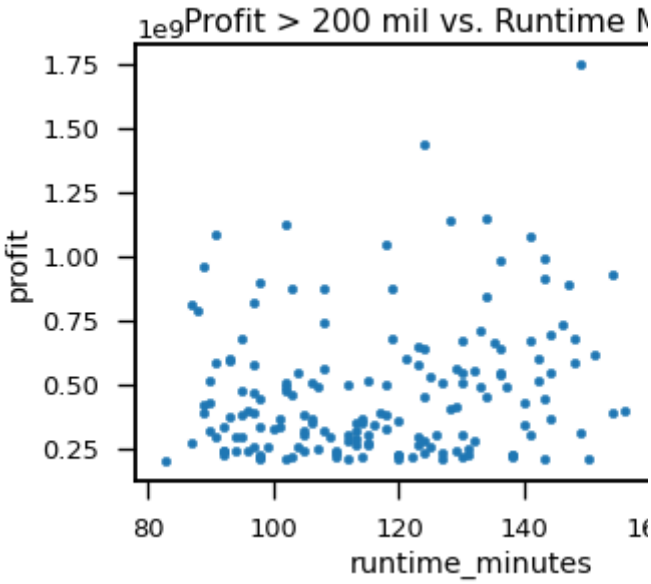
Out[226]:

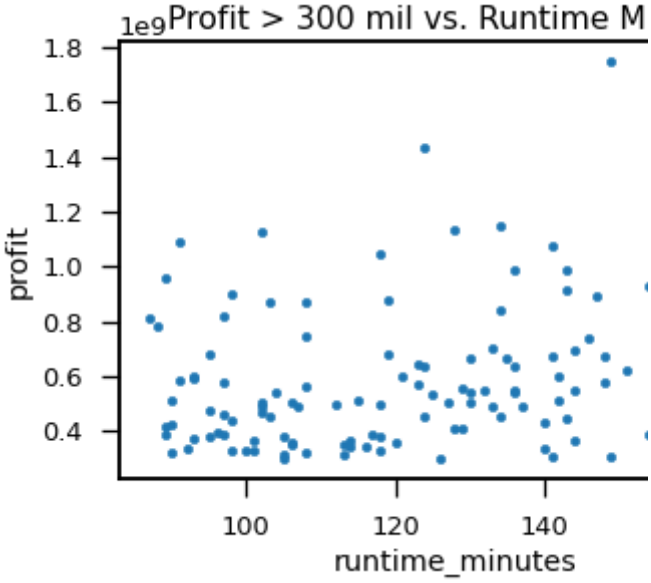
```
'I wanted to compare movies with high p
untimes of the movies. We \ndetermine t
s, but also most movies that generate l
```

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

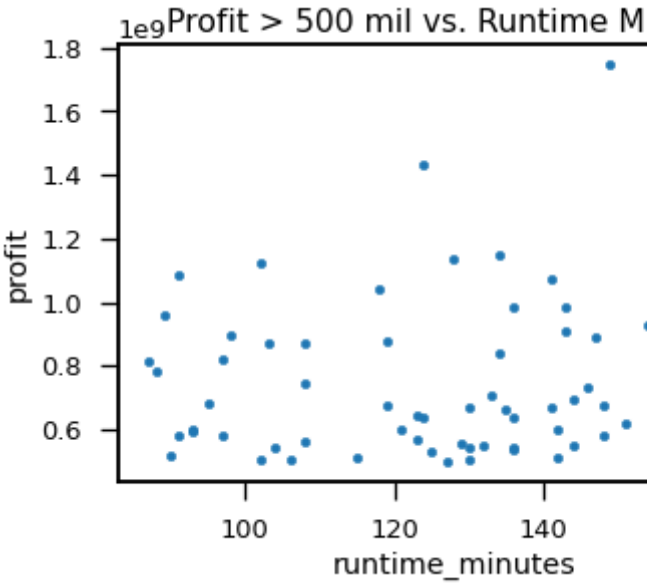


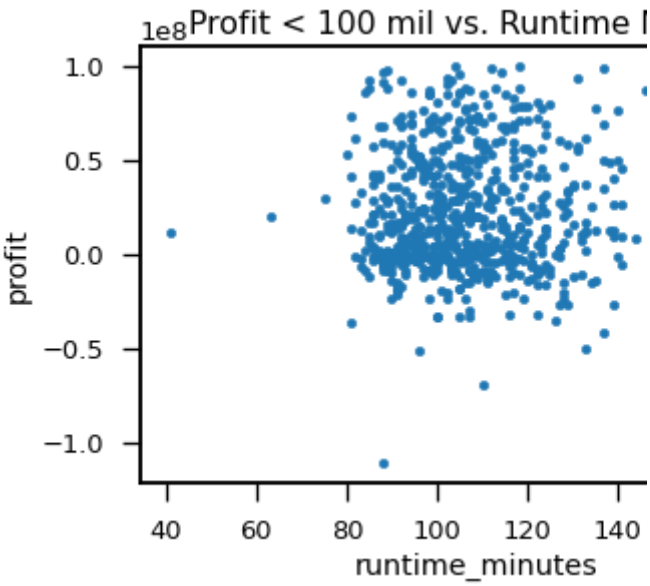




Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions





Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [175]:

```
"""Check to see what profit correl
most with gross either domestic or
profit has a somewhat significant
also surprisingly has almost not c
means that the more Microsoft sper
the profit will go. """
movie_info.corr()


```

Out[175]:

	domestic_gross_x	year	sta
domestic_gross_x	1.000000	0.103828	0.
year	0.103828	1.000000	1.
start_year	0.103828	1.000000	1.
runtime_minutes	0.276856	0.133445	0.
id	-0.044136	0.041758	0.
production_budget	0.688917	0.055248	0.
domestic_gross_y	0.999987	0.104887	0.
worldwide_gross	0.938842	0.116309	0.
gross_to_cost_ratio	0.065553	0.030864	0.
profit	0.934438	0.124435	0.

In [176]:

```
spec_box = movie_info[movie_info['
spec_box2 = exploded_movies[explod
spec_box3 = spec_box2[spec_box2['g
```

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

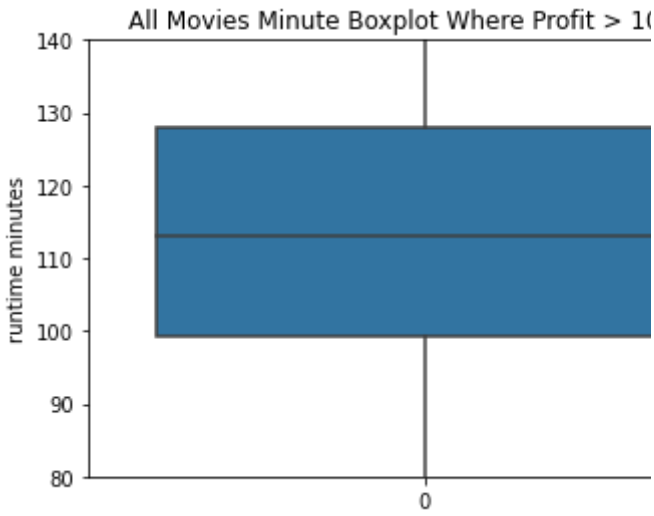
In [177]:
```
1 genre_index
```

Out[177]:

genres	runtime_minutes	production_budget	dom
Adventure	103.0	200000000	
Animation	103.0	200000000	
Comedy	103.0	200000000	
Action	148.0	160000000	
Adventure	148.0	160000000	
...	...	...	
Crime	112.0	10000000	
Drama	112.0	10000000	
Action	121.0	6000000	
Fantasy	121.0	6000000	
Horror	121.0	6000000	

2595 rows × 6 columns

In [183]:
```
1 runtimebox = sns.boxplot(data = sp
2     ylabel = 'runtime minutes', yl
3     title = 'All Movies Minute Box
4
5 plt.savefig('./images/runtimebox.p
6 plt.close('runtimebox.png')
```

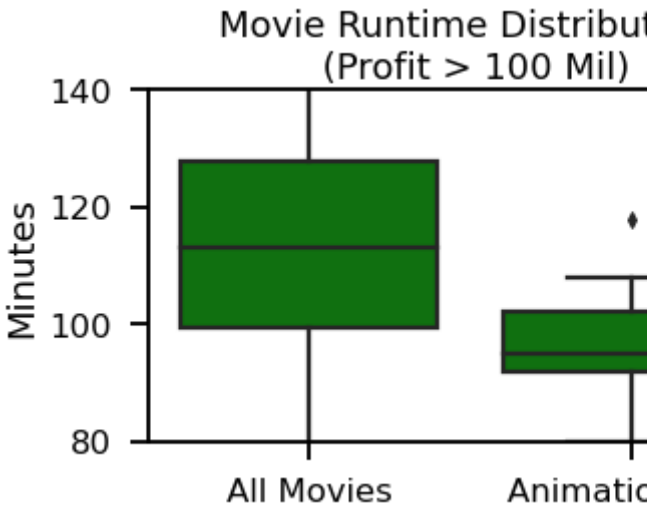


Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Prc
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

In [248]:

```
sns.set_context("talk")
runtimebox2 = sns.boxplot(data = |

    ylabel = 'Minutes', ylim = (8(
    title = 'Movie Runtime Distrik

plt.xticks([0,1],['All Movies','Ar
plt.tight_layout()

plt.savefig('./images/runtimebox2.
```



1.6 Evaluation

The model fits the data well. I am confident that the r
genre. A major new factor in this industry is the effec
ultimately conclude that Microsoft should go with wc
who know the industry well.

My work solves the business problem of which movi
risk.

1.7 Conclusions

Based on the results, my top recommendation for Mi
median genre that both profit and gross to cost ratio
than most alternatives and a safe venture with likely

Microsoft also has software and a team that can eas

Contents

- 1 Project Title
  - 1.1 Overview
  - 1.2 Business Pro
  - 1.3 Data Unders
  - 1.4 Data Prepara
  - 1.5 Data Modelir
  - 1.6 Evaluation
  - 1.7 Conclusions

Based on the profit vs runtime minute graphs, movie
However, the animation median is 95 minutes, so the
minutes.

Profit is most correlated with budget. So we recogniz
movie we recommend an approximate 10-15 million

We are limited in our results, and for future movies w

Limiting factors in our results included throwing out \
movies within an approximate 10 year team period a
and we could create a short list of those to make a d
services have had on the industry and whether Micro

Suggestions for our future exploration include:

1. Expand our data to more than the most recent 1
2. Do a similar data analysis but separate movies b
3. Connect author, producer, and director to our da

I have confidence in my initial suggestion, but would
studio.