Final Project Submission

Please fill out:

- Student name: Ian Ngugi
- Student pace: full time
- Scheduled project review date/time:
- Instructor name: Nikita Njoroge and Lucille Kaleha
- Blog post URL:

DESCRIBING THE QUESTION

1. SPECIFYING THE QUESTION

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.

1.1 DEFINING THE METRIC FOR SUCCESS

In order for Microsoft to be able to pull off movie production successfully they require to understand the average budget required for production , most popular genre and the average return on interest.The return on interest can be calculated by taking the budget and subtracting it from the worldwide gross.

DATA PREPARATION

In [1]:

```
# importing the required libraries
import pandas as pd
import sqlite3
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
conn = sqlite3.connect(r"C:\Users\PC\Documents\course_content\Phase_1_project\dsc-phase-1-p
```

In [57]:

```
#checking where the filepath exists before executing
import os
fn=(r"C:\Users\PC\Documents\course_content\Phase_1_project\dsc-phase-1-project\zippedData\b
os.path.exists(fn)
fn2=(r"C:\Users\PC\Documents\course_content\Phase_1_project\dsc-phase-1-project\zippedData\
os.path.exists(fn2)
fn3=(r"C:\Users\PC\Documents\course_content\Phase_1_project\dsc-phase-1-project\zippedData\
os.path.exists(fn3)
```

Out[57]:

```
True
```

In [3]:

```
#description of the datasets we will be working on
#the first data set will help us determine which films did the best in terms of foreign and
box_office=pd.read_csv(r"C:\Users\PC\Documents\course_content\Phase_1_project\dsc-phase-1-p
box_office
```

Out[3]:

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010
3	Inception	WB	292600000.0	535700000	2010
4	Shrek Forever After	P/DW	238700000.0	513900000	2010
...	...	...	...	...	...
3382	The Quake	Magn.	6200.0	NaN	2018
3383	Edward II (2018 re-release)	FM	4800.0	NaN	2018
3384	El Pacto	Sony	2500.0	NaN	2018
3385	The Swan	Synergetic	2400.0	NaN	2018
3386	An Actor Prepares	Grav.	1700.0	NaN	2018

3387 rows × 5 columns

In [4]:

```
#this dataset will help us get the average budget in movie production
the_numbers=pd.read_csv(r"C:\Users\PC\Documents\course_content\Phase_1_project\dsc-phase-1-
the_numbers.head()
```

Out[4]:

	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

In [5]:

```
#the movie basics and movie ratings tables will help us understand more on what genres are
#this will help shape our recommendation on what genre  microsoft should film
pd.read_sql("""SELECT * FROM movie_basics;""",conn)
```

Out[5]:

	movie_id	primary_title	original_title	start_year	runtime_minutes	genres
0	tt0063540	Sunghursh	Sunghursh	2013	175.0	Action,Crime,Drama
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114.0	Biography,Drama
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122.0	Drama
3	tt0069204	Sabse Bada Sukh	Sabse Bada Sukh	2018	NaN	Comedy,Drama
4	tt0100275	The Wandering Soap Opera	La Telenovela Errante	2017	80.0	Comedy,Drama,Fantasy
...	...	...	...	...	...	...
146139	tt9916538	Kuambil Lagi Hatiku	Kuambil Lagi Hatiku	2019	123.0	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	116.0	None
146143	tt9916754	Chico Albuquerque - Revelações	Chico Albuquerque - Revelações	2013	NaN	Documentary

146144 rows × 8 columns

In [6]:

```
pd.read_sql("""SELECT* FROM movie_ratings;""",conn)
```

Out[6]:

	movie_id	averagerating	numvotes	high_rated	highly_rated	ratings
0	tt10356526	8.3	31	None	None	None
1	tt10384606	8.9	559	None	None	None
2	tt1042974	6.4	20	None	None	None
3	tt1043726	4.2	50352	None	None	None
4	tt1060240	6.5	21	None	None	None
...	...	...	...	...	...	...
73851	tt9805820	8.1	25	None	None	None
73852	tt9844256	7.5	24	None	None	None
73853	tt9851050	4.7	14	None	None	None
73854	tt9886934	7.0	5	None	None	None
73855	tt9894098	6.3	128	None	None	None

73856 rows × 6 columns

In [7]:

```
# getting more information on the data sets
box_office.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   object 
 4   year            3387 non-null   int64  
dtypes: float64(1), int64(1), object(3)
memory usage: 132.4+ KB
```

In [8]:

```
the_numbers.info()
```

```
<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
```

DATA CLEANING

BOX OFFICE DATASET

checking for incomplete,missing or NaN values

In [9]:

```
#checking the number of null values
box_office.isnull().sum()
#foreign_gross and domestic_gross have  null values we can therefore use the most frequent
```

Out[9]:

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

In [10]:

```
box_office.describe(include='all')
```

Out[10]:

	title	studio	domestic_gross	foreign_gross	year
count	3387	3382	3.359000e+03	2037	3387.000000
unique	3386	257	NaN	1204	NaN
top	Bluebeard	IFC	NaN	1200000	NaN
freq	2	166	NaN	23	NaN
mean	NaN	NaN	2.874585e+07	NaN	2013.958075
std	NaN	NaN	6.698250e+07	NaN	2.478141
min	NaN	NaN	1.000000e+02	NaN	2010.000000
25%	NaN	NaN	1.200000e+05	NaN	2012.000000
50%	NaN	NaN	1.400000e+06	NaN	2014.000000
75%	NaN	NaN	2.790000e+07	NaN	2016.000000
max	NaN	NaN	9.367000e+08	NaN	2018.000000

In [11]:

```
# filling in null values for the column domestic_gross using the mean
box_office['domestic_gross'].fillna(28745850,inplace=True)
box_office['domestic_gross'].isnull().sum()
```

Out[11]:

```
0
```

In [12]:

```
#we should then procced to finding the highest foreign gross
#however since the values are stored as dtype object we need to convert them to integers
#we can achieve this by removing the ',' from the integers in the list
box_office['foreign_gross']=box_office['foreign_gross'].str.replace(',','')
box_office['foreign_gross']
```

Out[12]:

```
0       652000000
1       691300000
2       664300000
3       535700000
4       513900000
          ...
3382          NaN
3383          NaN
3384          NaN
3385          NaN
3386          NaN
Name: foreign_gross, Length: 3387, dtype: object
```

In [13]:

```
#getting the most frequent value which we will use to fill in for the NaN values in foreign
box_office['foreign_gross'].value_counts().max()
box_office.loc[box_office['foreign_gross'].value_counts().max()]
```

Out[13]:

```
title                  Salt
studio                 Sony
domestic_gross    1.183e+08
foreign_gross     175200000
year                   2010
Name: 23, dtype: object
```

In [14]:

```
#using the most frequent value to fill in for the NaN values
box_office['foreign_gross'].fillna(175200000,inplace=True)
box_office['foreign_gross']
```

Out[14]:

```
0       652000000
1       691300000
2       664300000
3       535700000
4       513900000
          ...
3382    175200000
3383    175200000
3384    175200000
3385    175200000
3386    175200000
Name: foreign_gross, Length: 3387, dtype: object
```

In [15]:

```
#we then procced with our initial task of converting the dtype objct to integer
box_office['foreign_gross'] = pd.to_numeric(box_office['foreign_gross']).astype(int)
box_office['foreign_gross']
```

Out[15]:

```
0       652000000
1       691300000
2       664300000
3       535700000
4       513900000
          ...
3382    175200000
3383    175200000
3384    175200000
3385    175200000
3386    175200000
Name: foreign_gross, Length: 3387, dtype: int32
```

In [16]:

```
box_office.describe(include='all')
```

Out[16]:

	title	studio	domestic_gross	foreign_gross	year
count	3387	3382	3.387000e+03	3.387000e+03	3387.000000
unique	3386	257	NaN	NaN	NaN
top	Bluebeard	IFC	NaN	NaN	NaN
freq	2	166	NaN	NaN	NaN
mean	NaN	NaN	2.874585e+07	1.148615e+08	2013.958075
std	NaN	NaN	6.670497e+07	1.173333e+08	2.478141
min	NaN	NaN	1.000000e+02	6.000000e+02	2010.000000
25%	NaN	NaN	1.225000e+05	1.160000e+07	2012.000000
50%	NaN	NaN	1.400000e+06	1.205000e+08	2014.000000
75%	NaN	NaN	2.874585e+07	1.752000e+08	2016.000000
max	NaN	NaN	9.367000e+08	9.605000e+08	2018.000000

THE NUMBERS DATASET

In [17]:

```
#checking for null values
the_numbers.isnull().any()
```

Out[17]:

```
id                   False
release_date         False
movie                False
production_budget    False
domestic_gross       False
worldwide_gross      False
dtype: bool
```

In [18]:

```
#inorder to do arithmetic calculations with the columns we need to perfom some operations i
the_numbers['production_budget']=the_numbers['production_budget'].str.replace(',','').str.r
the_numbers['production_budget']=pd.to_numeric(the_numbers['production_budget']).astype(int
```

In [19]:

```
the_numbers['domestic_gross']=the_numbers['domestic_gross'].str.replace(',','').str.replace
the_numbers['domestic_gross']=pd.to_numeric(the_numbers['domestic_gross']).astype(int)
```

In [20]:

```
the_numbers['worldwide_gross']=the_numbers['worldwide_gross'].str.replace(',','').str.repla
the_numbers['worldwide_gross']=pd.to_numeric(the_numbers['worldwide_gross']).astype(int)
```

MOVIE RATINGS AND MOVIE BASICS TABLES

In [21]:

```
the_numbers.describe(include='all')
```

Out[21]:

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross
count	5782.000000	5782	5782	5.782000e+03	5.782000e+03	5.782000e+03
unique	NaN	2418	5698	NaN	NaN	NaN
top	NaN	Dec 31, 2014	King Kong	NaN	NaN	NaN
freq	NaN	24	3	NaN	NaN	NaN
mean	50.372363	NaN	NaN	3.158776e+07	4.187333e+07	9.000183e+07
std	28.821076	NaN	NaN	4.181208e+07	6.824060e+07	1.725459e+08
min	1.000000	NaN	NaN	1.100000e+03	0.000000e+00	-2.086759e+09
25%	25.000000	NaN	NaN	5.000000e+06	1.429534e+06	4.102274e+06
50%	50.000000	NaN	NaN	1.700000e+07	1.722594e+07	2.794748e+07
75%	75.000000	NaN	NaN	4.000000e+07	5.234866e+07	9.758278e+07
max	100.000000	NaN	NaN	4.250000e+08	9.366622e+08	2.053311e+09

In [22]:

```
#here we have searched  for genre with the highest rating and highest number of votes
pd.read_sql("""SELECT MAX( movie_basics.genres) AS count
FROM movie_basics
INNER JOIN movie_ratings
ON movie_basics.movie_id=movie_ratings.movie_id
WHERE averagerating>7.0 AND numvotes>100000;""",conn)
```

Out[22]:

	count
0	Romance,Sci-Fi,Thriller

DATA ANALYSIS

From the box office dataset we learn that the average domestic gross is 28,745,850 and the average foreign gross is 114,861,500.The average production budget was 31,587,760. From the numbers dataset we observe that a low production budget leads to little or no return , on the other hand, the higher the production budget the higher the worldwide gross.

The movie ratings and movie basics table identifies the most popular movie genres as; Romance,Sci-Fi and Thriller.

The data also demonsrated a high correlation between the worldwide gross and production budget as indicated in figure 1.1 below. In addition,as shown in figure 1.2 not all films that had a high production budget produced high profit margins as well as the inverse.

Average return on interest was: 59,682,180 based on the average difference between the worldwide gross and production budget.

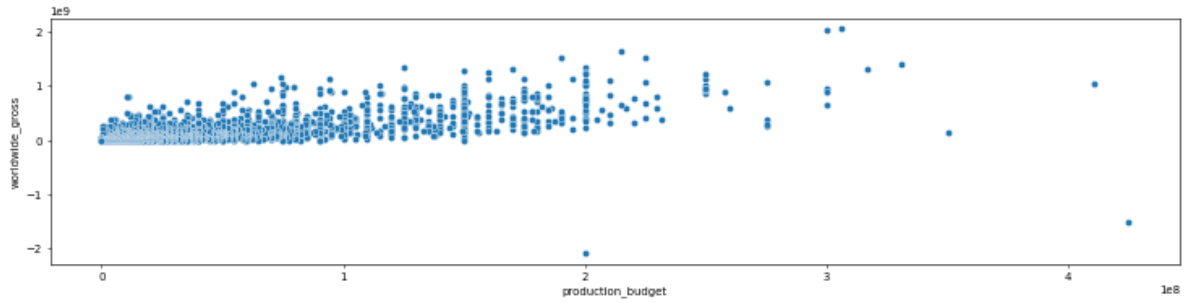
FIGURE 1.1

In [23]:

```
#first we begin by looking at the correlation between production budget and the worldwide g
plt.figure(figsize=(18,4),dpi=50)
sns.scatterplot(x='production_budget',y='worldwide_gross',data=the_numbers )
```

Out[23]:

```
<AxesSubplot:xlabel='production_budget', ylabel='worldwide_gross'>
```



In [47]:

```
#preparing a dataframe based on the worldwide gross and production budget column
#we will use this new dataframe to plot a histogram
combined_dataframe=the_numbers[['worldwide_gross','production_budget']].copy()
combined_dataframe.at[0,'worldwide_gross']=1518622017
big_dataframe=combined_dataframe.sort_values(by=['production_budget'],ascending=False).head
big_dataframe.head(10)
```

Out[47]:

	worldwide_gross	production_budget
0	1518622017	425000000
1	1045663875	410600000
2	149762350	350000000
3	1403013963	330600000
4	1316721747	317000000
5	2053311220	306000000
6	2048134200	300000000
7	963420425	300000000
8	655945209	300000000
9	879620923	300000000

In [25]:

```
combined_dataframe2=the_numbers[['worldwide_gross','production_budget']].copy()
big_dataframe2=combined_dataframe.sort_values(by=['production_budget'],ascending=True).head
big_dataframe2.head(20)
```

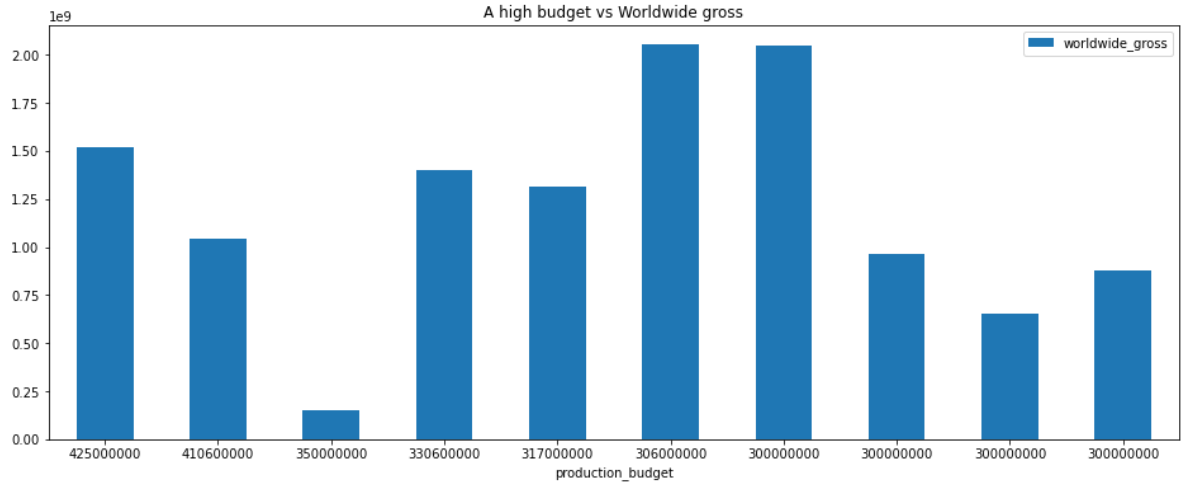
Out[25]:

	worldwide_gross	production_budget
5781	181041	1100
5780	0	1400
5779	1338	5000
5778	240495	6000
5776	900	7000
5775	71644	7000
5774	841926	7000
5773	2041928	7000
5777	0	7000
5772	4584	9000

FIGURE 1.2

In [30]:

```
#a bar graph representation of the data
big_dataframe.plot.bar(x='production_budget',y='worldwide_gross',rot=0)
plt.title('A high budget vs Worldwide gross')
plt.rcParams['figure.figsize']=[16,6]
```

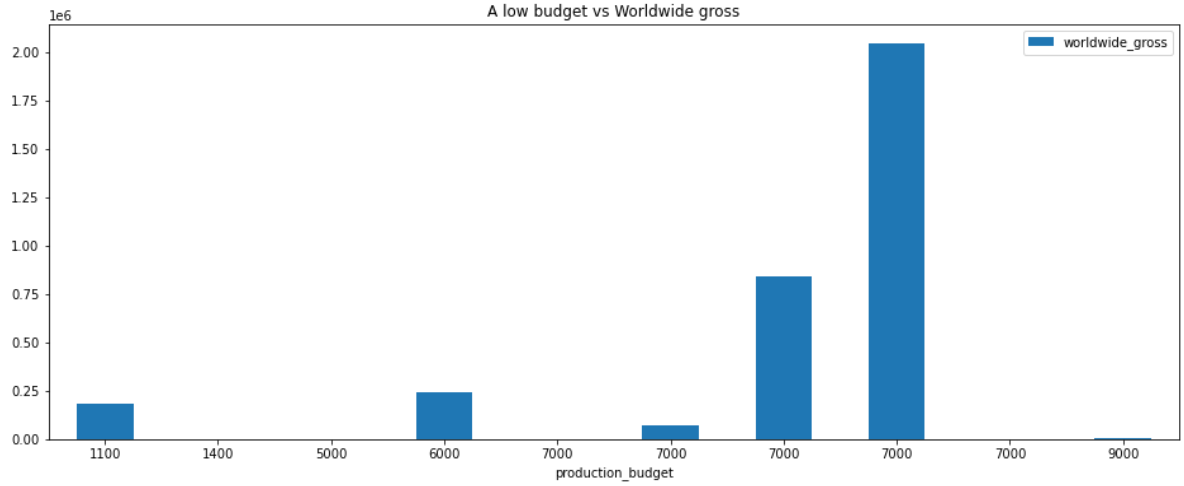


In [27]:

```
big_dataframe2.plot.bar(x='production_budget',y='worldwide_gross',rot=0)
plt.rcParams['figure.figsize']=[10,10]
plt.title('A low budget vs Worldwide gross')
```

Out[27]:

```
Text(0.5, 1.0, 'A low budget vs Worldwide gross')
```



In [56]:

```
#now we calculate the average return on profit
#this is obtained by subtracting the production budget from the worldwide gross
return_on_interest=combined_dataframe['worldwide_gross']-combined_dataframe['production_bud
return_on_interest.mean()
print(f'the average return on interest was:',return_on_interest.mean().round())
```

```
the average return on interest was: 59682180.0
```

CONCLUSION

In conclusion , it can be deduced that a high production budget does not necessarily lead to a higher rate of interest. Secondly even though the worldwide gross increases with the production budget , it can be concluded that the return on interest will not be affected by increasing the production budget.

RECOMMENDATION

In order to create a successful film I recommend sourcing for technical expertise in production of the three highly rated genres.I would also recommend working with the average production budget in order to maximise the return on interest.