MICROSOFT MOVIES ANALYSIS

- Student name: Frida Oyucho
- Student pace: full time
- Instructor name: William Okombo/Noah Kandie

Data Preparation ¶

Steps taken

- Importing of merged data
- Dropped unnecessary columns
- Removed special characters and whitespace from Production_budget & worldwide_gross columns
- Changed Data Types for Production_budget & worldwide_gross columns
- Dropped rows in runtime_minutes column
- Replaced Null values with MISSING for Movies column
- Removed the missing values which were denoted by zero for Production_budget & worldwide_gross columns
- Identified and removed duplicates
- Removed extreme outliers using IQR

In [2]:

```
# Importing necessary Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:

```
# Importing them merged dataset and converting it to a dataframe

df = pd.read_csv('FinalMergedData.csv', index_col=0)
df.head()
```

Out[3]:

	genre_ids	id_x	original_language	original_title	popularity	release_date_x	title
0	[12, 14, 10751]	12444	en	Harry Potter and the Deathly Hallows: Part 1	33.533	2010-11-19	Harry Potter and the Deathly Hallows: Part 1
1	[14, 12, 16, 10751]	10191	en	How to Train Your Dragon	28.734	2010-03-26	How to Train Your Dragon
2	[12, 28, 878]	10138	en	Iron Man 2	28.515	2010-05-07	Iron Man 2
3	[28, 878, 12]	27205	en	Inception	27.920	2010-07-16	Inception
4	[12, 14, 10751]	32657	en	Percy Jackson & the Olympians: The Lightning T...	26.691	2010-02-11	Percy Jackson & the Olympians: The Lightning T...

In [4]:

```
# Preview column names
df.columns
```

Out[4]:

```
Index(['genre_ids', 'id_x', 'original_language', 'original_title',
       'popularity', 'release_date_x', 'title', 'vote_average', 'vote_coun
t',
       'tconst', 'primary_title', 'start_year', 'runtime_minutes', 'genre
s',
       'id_y', 'release_date_y', 'movie', 'production_budget',
       'domestic_gross', 'worldwide_gross'],
      dtype='object')
```

In [5]:

```
# Dropping unnecessary Columns
df.drop(['genre_ids','id_x','original_title','release_date_x','id_y','relea
df.shape
```

Out[5]:

```
(21078, 12)
```

In [7]:

```
# Checking information about the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21078 entries, 0 to 21077
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   original_language  21078 non-null  object 
 1   popularity         21078 non-null  float64
 2   title              21078 non-null  object 
 3   vote_average       21078 non-null  float64
 4   tconst             21078 non-null  object 
 5   primary_title      21078 non-null  object 
 6   start_year         21078 non-null  int64  
 7   runtime_minutes    19452 non-null  float64
 8   genres             20779 non-null  object 
 9   movie              5782 non-null   object 
 10  production_budget  5782 non-null   object 
 11  worldwide_gross    5782 non-null   object 
dtypes: float64(3), int64(1), object(8)
memory usage: 2.1+ MB
```

In [8]:

```
# Removing $ sign from  Production_budget & worldwide_gross
df['production_budget']=df['production_budget'].str.replace('$', '')
df['production_budget']

df['worldwide_gross']=df['worldwide_gross'].str.replace('$','')
```

In [9]:

```
# Removing ',' sign from  Production_budget & worldwide_gross
df['production_budget']=df['production_budget'].str.replace(',', '')
df['production_budget']

df['worldwide_gross']=df['worldwide_gross'].str.replace(',','')
```

In [10]:

```
df['production_budget'].dtype
```

Out[10]:

```
dtype('O')
```

In [11]:

```
# Changing objects to int for Production Budget & Worldwide_gross
# Replace NaN with 0
df['production_budget'] = df['production_budget'].fillna(0)
df['production_budget'] = df['production_budget'].astype(int)

# Changing objects to int for Production Budget & Worldwide_gross
# Replace NaN with 0
df['worldwide_gross'] = df['worldwide_gross'].fillna(0)
df['worldwide_gross'] = df['worldwide_gross'].astype(float)
```

In [12]:
```
# Handling missing Values
# Checking the proportion of missing data
df.isnull().mean()
```

Out[12]:
```
original_language    0.000000
popularity           0.000000
title                0.000000
vote_average         0.000000
tconst               0.000000
primary_title        0.000000
start_year           0.000000
runtime_minutes      0.077142
genres               0.014185
movie                0.725686
production_budget    0.000000
worldwide_gross      0.000000
dtype: float64
```

In [13]:
```
# Dropping rows in runtime_minutes column because of the low proportion  of
df = df.dropna(subset=['runtime_minutes','genres'])
# df.isnull().mean()
```

In [14]:
```
# Replacing Null values with the word MISSING for Movies
df['movie']=df['movie'].fillna('missing')
df.isnull().mean()
```

Out[14]:
```
original_language    0.0
popularity           0.0
title                0.0
vote_average         0.0
tconst               0.0
primary_title        0.0
start_year           0.0
runtime_minutes      0.0
genres               0.0
movie                0.0
production_budget    0.0
worldwide_gross      0.0
dtype: float64
```

In [15]:
```
df.shape
```

Out[15]:
```
(19329, 12)
```

In [16]:
```
#Removing the missing values which are denoted by zero
df=df[(df['production_budget']!=0)]
df=df[(df['worldwide_gross']!=0)]
df.shape
```

Out[16]:
```
(4853, 12)
```

In [17]:
```
# Handling duplicate Records
# Checking for duplicates in the dataset
df.duplicated().any()

# Dropping the duplicates
df.drop_duplicates(inplace=True,keep = 'first')
```

In [18]:
```
df.shape
```

Out[18]:
```
(4853, 12)
```

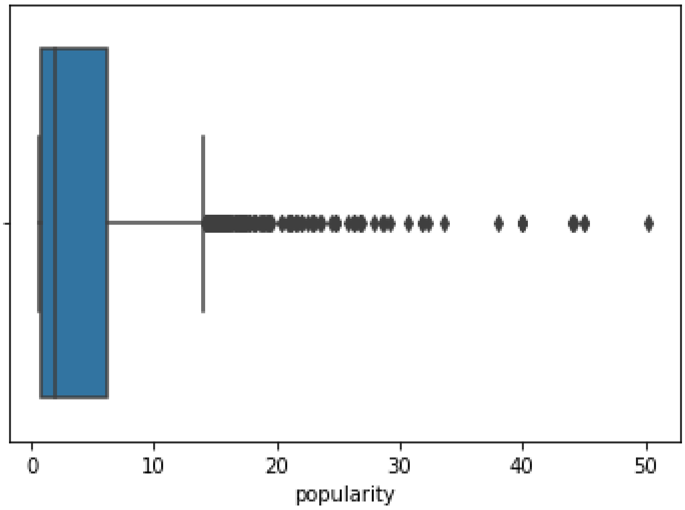
In [19]:
```
# Handling Outliers
df.describe()
```

Out[19]:
```
         popularity  vote_average   start_year  runtime_minutes  production_budget  worldwi
count   4853.000000   4853.000000  4853.000000      4853.000000       4.853000e+03     4.85
mean       4.014960      5.696023  2012.128992        91.926437       3.338989e+07     9.89
std        4.740305      1.570870     2.454837        23.707368       4.328410e+07     1.83
min        0.600000      0.000000  2010.000000         3.000000       5.000000e+03     2.60
25%        0.840000      5.000000  2010.000000        83.000000       6.000000e+06     6.60
50%        1.985000      5.800000  2011.000000        91.000000       1.800000e+07     3.29
75%        6.136000      6.600000  2013.000000       101.000000       4.000000e+07     1.04
max       50.289000     10.000000  2020.000000       495.000000       4.250000e+08     2.77
```

In [20]:
```
import seaborn as sns
```

In [23]:
```
# Visualizing Outliers
sns.boxplot(df['popularity']);
```

```
C:\Users\Fridah.Oyucho\AppData\Local\anaconda3\envs\learn-env\lib\site-pac
kages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: x. From version 0.12, the only valid positional argume
nt will be `data`, and passing other arguments without an explicit keyword
will result in an error or misinterpretation.
  warnings.warn(
```



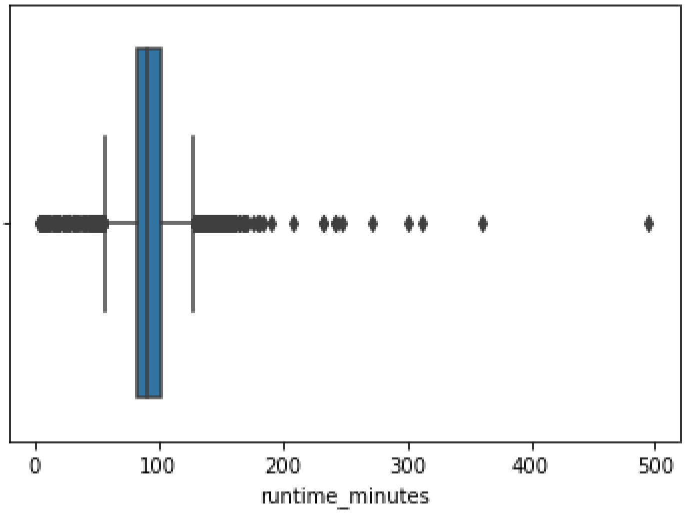
In [24]:
```
# Position of the Outlier values
x = np.where(df['runtime_minutes']>150)
print(x)
```

```
(array([ 261,  322,  391,  409,  473,  474,  478,  479,  604,  876,  919,
         953,  957,  961, 1515, 1520, 1522, 1700, 1796, 1839, 1909, 1914,
        1919, 1924, 1929, 1979, 2009, 2074, 2075, 2096, 2098, 2107, 2271,
        2364, 2492, 2605, 2612, 2683, 2687, 2786, 2828, 2832, 2834, 2939,
        2943, 2946, 2950, 3007, 3129, 3390, 3408, 3492, 3506, 3510, 3568,
        3844, 3849, 3853, 3882, 3946, 4135, 4361, 4392, 4764, 4770, 4775,
        4835], dtype=int64),)
```

In [21]:
```
sns.boxplot(df['runtime_minutes'])
```

```
C:\Users\Fridah.Oyucho\AppData\Local\anaconda3\envs\learn-env\lib\site-pac
kages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: x. From version 0.12, the only valid positional argume
nt will be `data`, and passing other arguments without an explicit keyword
will result in an error or misinterpretation.
  warnings.warn(
```

Out[21]:
```
<AxesSubplot:xlabel='runtime_minutes'>
```



In [25]:
```
#Creating a function to remove the outliers
def remove_outliers_usingIQR(df,column):
    q1 = df[column].quantile(0.25)
    q3 = df[column].quantile(0.75)
    iqr = q3 - q1
    lower_bound = q1 - 1.5*iqr
    upper_bound = q3 + 1.5*iqr
    return df[(df[column]>= lower_bound) & (df[column] <= upper_bound)]
```

In [26]:
```
#removing outliers using the created function
df = remove_outliers_usingIQR(df,'runtime_minutes')
df = remove_outliers_usingIQR(df,'production_budget')
df = remove_outliers_usingIQR(df,'worldwide_gross')
```

In [27]:
```
df = remove_outliers_usingIQR(df,'popularity')
```

In [28]:
```
df.describe()
```

Out[28]:

	popularity	vote_average	start_year	runtime_minutes	production_budget	worldwi
count	3454.000000	3454.000000	3454.000000	3454.000000	3.454000e+03	3.45
mean	2.803139	5.559873	2012.102490	91.269543	2.002213e+07	3.89
std	2.749238	1.621166	2.355448	12.667780	1.990075e+07	4.55
min	0.600000	0.000000	2010.000000	56.000000	5.000000e+03	2.60
25%	0.665250	4.800000	2011.000000	84.000000	5.000000e+06	4.02
50%	1.552000	5.600000	2011.000000	90.000000	1.400000e+07	2.01
75%	3.836000	6.600000	2012.000000	99.000000	3.000000e+07	5.89
max	11.571000	10.000000	2020.000000	128.000000	9.000000e+07	1.87

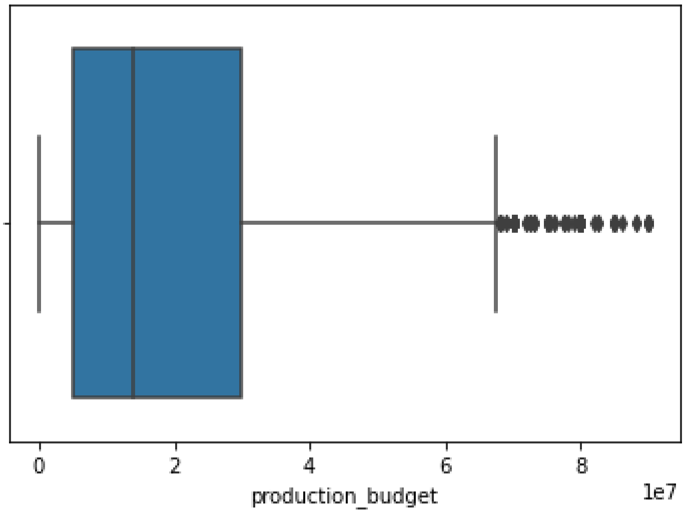
In [29]:
```
df.shape
```

Out[29]:
```
(3454, 12)
```

In [30]:
```
# Visualizing boxplots after removing Outliers
sns.boxplot(df['production_budget'])
```

```
C:\Users\Fridah.Oyucho\AppData\Local\anaconda3\envs\learn-env\lib\site-pac
kages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: x. From version 0.12, the only valid positional argume
nt will be `data`, and passing other arguments without an explicit keyword
will result in an error or misinterpretation.
  warnings.warn(
```

Out[30]:
```
<AxesSubplot:xlabel='production_budget'>
```



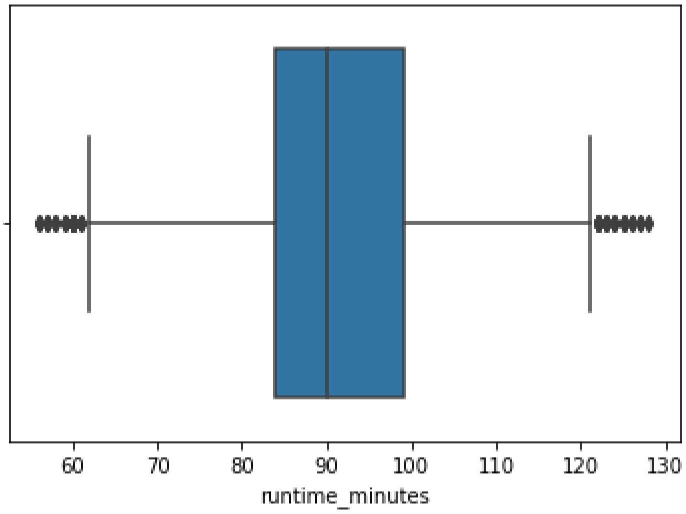
In [28]:

```
# Visualizing boxplots after removing Outliers
sns.boxplot(df['runtime_minutes'])
```

```
C:\Users\Fridah.Oyucho\AppData\Local\anaconda3\envs\learn-env\lib\site-pac
kages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: x. From version 0.12, the only valid positional argume
nt will be `data`, and passing other arguments without an explicit keyword
will result in an error or misinterpretation.
  warnings.warn(
```

Out[28]:

```
<AxesSubplot:xlabel='runtime_minutes'>
```



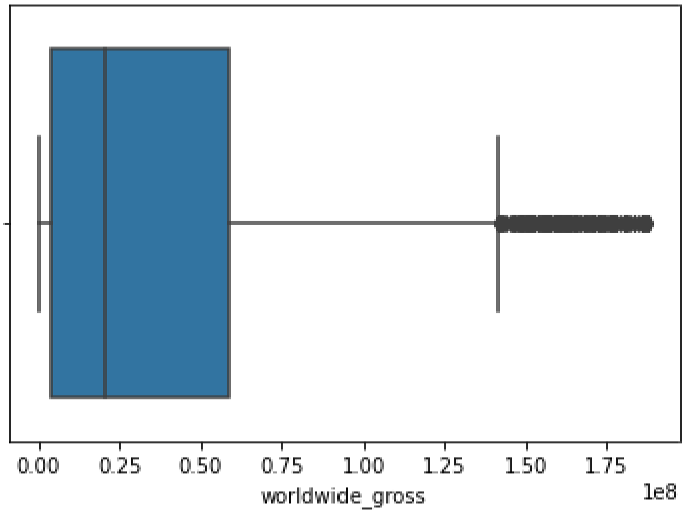
In [31]:

```
# Visualizing boxplots after removing Outliers
sns.boxplot(df['worldwide_gross'])
```

```
C:\Users\Fridah.Oyucho\AppData\Local\anaconda3\envs\learn-env\lib\site-pac
kages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: x. From version 0.12, the only valid positional argume
nt will be `data`, and passing other arguments without an explicit keyword
will result in an error or misinterpretation.
  warnings.warn(
```

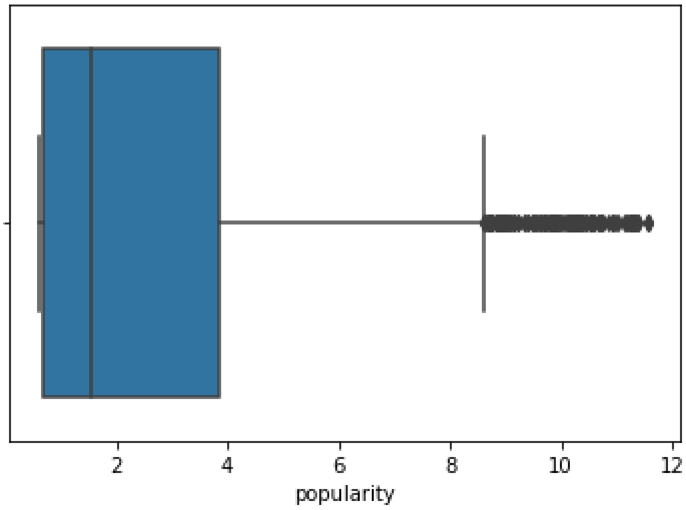
Out[31]:

```
<AxesSubplot:xlabel='worldwide_gross'>
```



In [32]:
```
# Visualizing boxplots after removing Outliers
sns.boxplot(df['popularity']);
```

```
C:\Users\Fridah.Oyucho\AppData\Local\anaconda3\envs\learn-env\lib\site-pac
kages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: x. From version 0.12, the only valid positional argume
nt will be `data`, and passing other arguments without an explicit keyword
will result in an error or misinterpretation.
  warnings.warn(
```



In [33]:
```
df.describe()
```

Out[33]:

	popularity	vote_average	start_year	runtime_minutes	production_budget	worldwi
count	3454.000000	3454.000000	3454.000000	3454.000000	3.454000e+03	3.45
mean	2.803139	5.559873	2012.102490	91.269543	2.002213e+07	3.89
std	2.749238	1.621166	2.355448	12.667780	1.990075e+07	4.55
min	0.600000	0.000000	2010.000000	56.000000	5.000000e+03	2.60
25%	0.665250	4.800000	2011.000000	84.000000	5.000000e+06	4.02
50%	1.552000	5.600000	2011.000000	90.000000	1.400000e+07	2.01
75%	3.836000	6.600000	2012.000000	99.000000	3.000000e+07	5.89
max	11.571000	10.000000	2020.000000	128.000000	9.000000e+07	1.87

In [34]:
```
df.shape
```

Out[34]:
```
(3454, 12)
```

In [35]:
```
# Exporting Output to CSV file
df.to_csv('CleanedData.csv', index=False)
```

Conclusion on Handling Outliers

We managed to remove extreme but not all outliers