PROJECT ANALYSIS OF IDMB MOVIES DATA BETWEEN 2010-2018

Zahra & Hellen

Our Objective

Determining whether it is profitable for Microsoft to invest in the Film Industry

In [27]:
```
#importing python packages
import pandas as pd
import numpy as np
import seaborn as sns
from scipy import stats
```

In [29]:
```
movies_data = pd.read_csv(r'C:\Users\LENOVO THINKPAD L540\Desktop\ZAHRA\movies_data\movies_
movies_data
```

0	Toy Story 3	BV	415000000.0	652000000	2010	47	Jun 18, 2010	Toy Story 3	$200,000,000
1	Inception	WB	292600000.0	535700000	2010	38	Jul 16, 2010	Inception	$160,000,000
2	Shrek Forever After	P/DW	238700000.0	513900000	2010	27	May 21, 2010	Shrek Forever After	$165,000,000
3	The Twilight Saga: Eclipse	Sum.	300500000.0	398000000	2010	53	Jun 30, 2010	The Twilight Saga: Eclipse	$68,000,000
4	Iron Man 2	Par.	312400000.0	311500000	2010	15	May 7, 2010	Iron Man 2	$170,000,000
...	...	...	...	...	...	...	...	...	...
1242	Gotti	VE	4300000.0	NaN	2018	64	Jun 15, 2018	Gotti	$10,000,000
1243	Ben is Back	RAtt.	3700000.0	NaN	2018	95	Dec 7, 2018	Ben is Back	$13,000,000

Missing values

In [ ]:
```
# check for missing values
#NaN - Not a number or null
```

In [ ]:
```
# check for missing values

movies_data.isna().sum()
```

In [ ]:
```
# treatment for missing values
```

In [ ]:
```
# dropping single columns

movies_data.drop(['id'],axis = 1)
```

In [ ]:
```
# drop all rows with missing values

movies_data.dropna(inplace=True)
```

In [ ]:
```
# dropping multiple columns

movies_data.drop(['title', 'id'], axis = 1, inplace=True)
```

Duplicates

In [ ]:
```
# check for duplicates

movies_data.duplicated().value_counts()
```

In [ ]:
```
# drop duplicates
movies_data.drop_duplicates()
```

Fixing data structure

In [ ]:
```
import pandas as pd
```

In [ ]:
```
#turn_data = pd.read_csv(r'C:\Users\LENOVO THINKPAD L540\Desktop\movies_data\movies_data.cs
#turn_data
```

In [ ]:
```
# call the dataframe

#turn_data
```

In [ ]:
```
# change data types
#turn_data['movies'] = turn_data['movies'].astype(str)
```

In [ ]:
```
#turn_data.info()
```

In [ ]:
```
#turn_data.info()
```

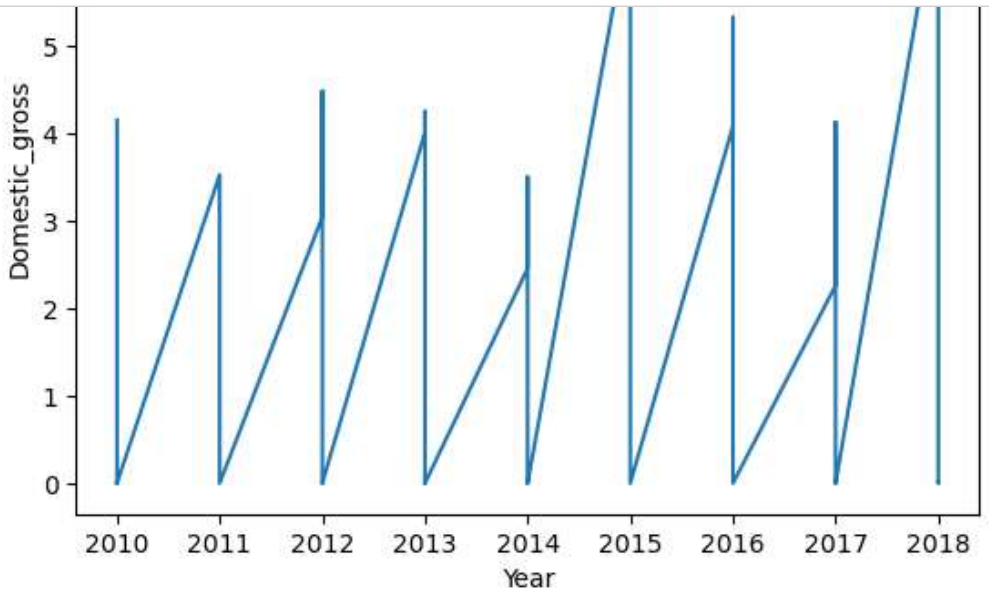
Data visualization

Investigating the Relationship Between our Variables

In [ ]:
```
# import neccessary libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Line Plot

In [30]:
```
# studio
plt.plot(movies_data['year'], movies_data['domestic_gross'])

plt.title('Domestic gross by Year in USD')
plt.xlabel('Year')
plt.ylabel('Domestic_gross')
plt.show()
```



In [31]:
```
movies_data.dtypes
```

Out[31]:
```
title                object
studio               object
domestic_gross      float64
foreign_gross        object
year                  int64
id                    int64
release_date         object
movie                object
production_budget    object
worldwide_gross      object
dtype: object
```

In [32]:
```
movies_data['foreign_gross']
```

Out[32]:
```
0       652000000
1       535700000
2       513900000
3       398000000
4       311500000
          ...
1242          NaN
1243          NaN
1244      1700000
1245          NaN
1246          NaN
Name: foreign_gross, Length: 1247, dtype: object
```

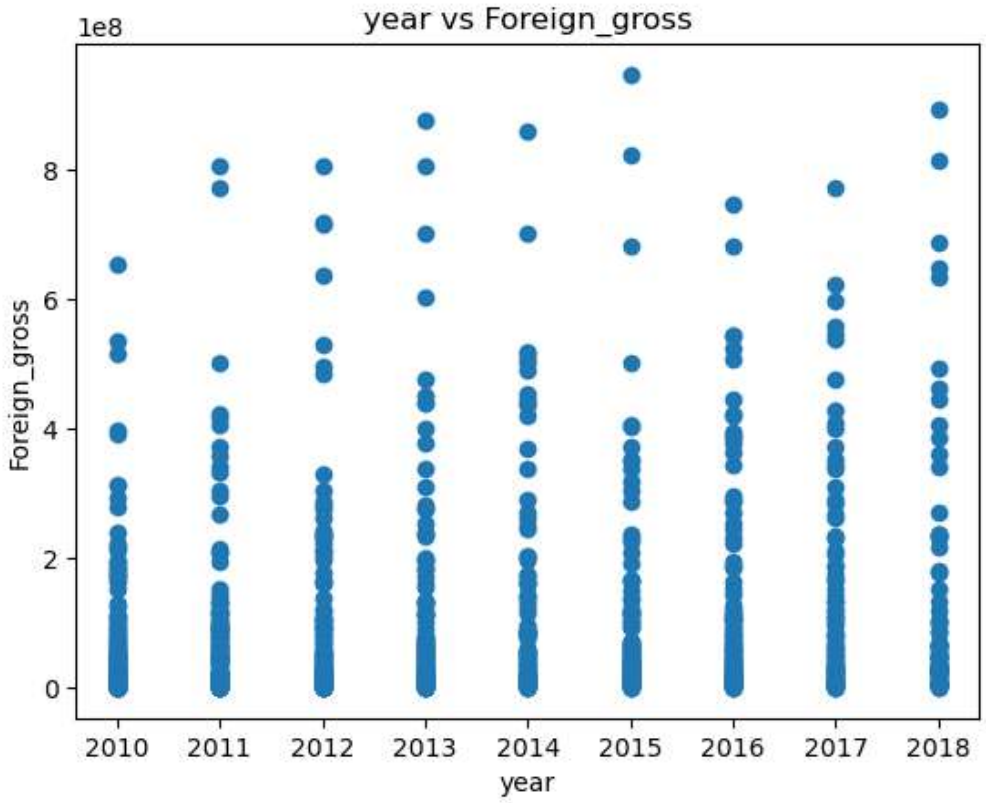
In [37]:
```
movies_data['foreign_gross'] = pd.to_numeric(movies_data['foreign_gross'], errors="coerce"
```

In [38]:
```
movies_data.dtypes
```

Out[38]:
```
title                 object
studio                object
domestic_gross       float64
foreign_gross        float64
year                   int64
id                     int64
release_date          object
movie                 object
production_budget     object
worldwide_gross       object
dtype: object
```

In [ ]:
```
# change data types
movies_data['foreign_gross'] = movies_data['foreign_gross'].astype(int)
```

In [35]:
```
##scatter
x =movies_data['year']
y =movies_data['foreign_gross']
plt.scatter(x, y)

plt.title('Scatter Plot')
plt.ylabel('Foreign_gross')
plt.xlabel('year')
plt.title('year vs Foreign_gross')
plt.show()
```



Plotting Charts

Bar Charts

```
movies_data.dtypes
```

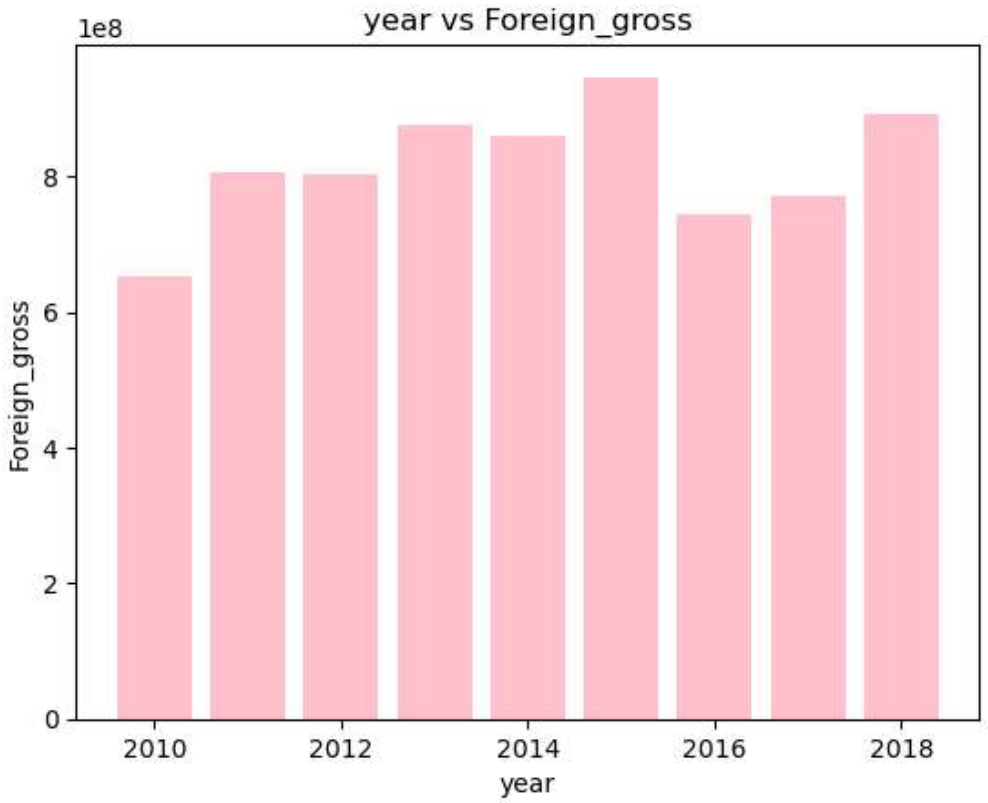
In [ ]:
```
movies_data['foreign_gross']
```

In [ ]:
```
movies_data['foreign_gross'] = pd.to_numeric(movies_data['foreign_gross'], errors="coerce"
```

In [ ]:
```
movies_data.dtypes
```

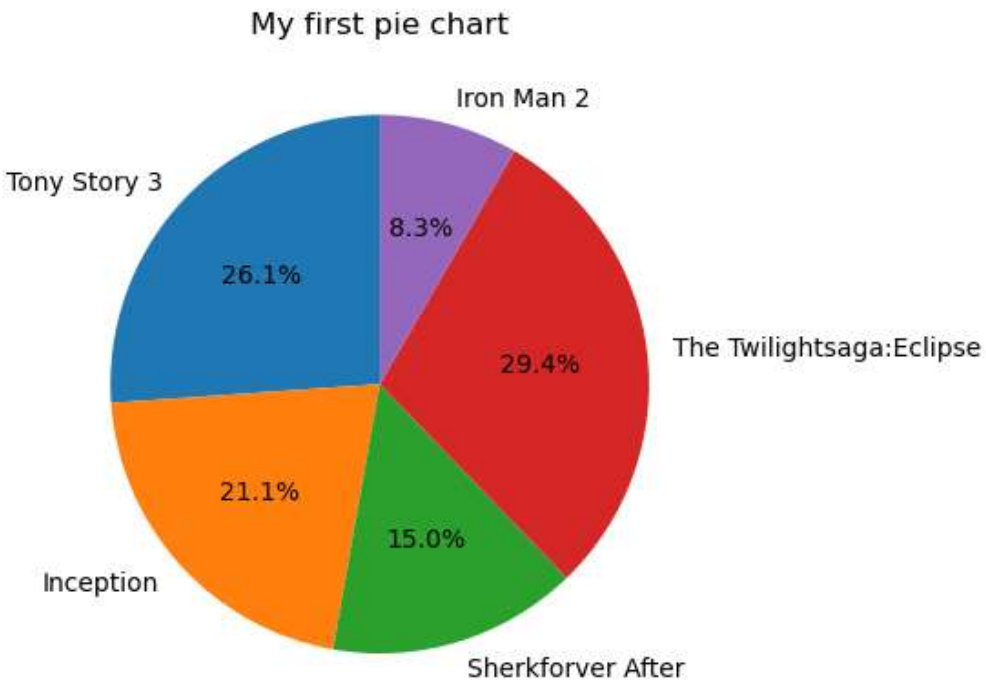
In [ ]:
```
# change data types
movies_data['foreign_gross'] = movies_data['foreign_gross'].astype(int)
```

In [45]:
```
##bar chart
x =movies_data['year']
y =movies_data['foreign_gross']
plt.bar(x, y , color='pink')
plt.ylabel('Foreign_gross')
plt.xlabel('year')
plt.title('year vs Foreign_gross')
plt.show()
```



Pie Charts

In [43]:
```
import pandas as pd

##pie chart
slices = [47, 38, 27, 53, 15]
labels = ['Tony Story 3', 'Inception', 'Sherkforver After', 'The Twilightsaga:Eclipse', 'I
plt.pie(slices, labels = labels,startangle = 90, autopct = '%.1f%%')
plt.title('My first pie chart')
plt.show()
```



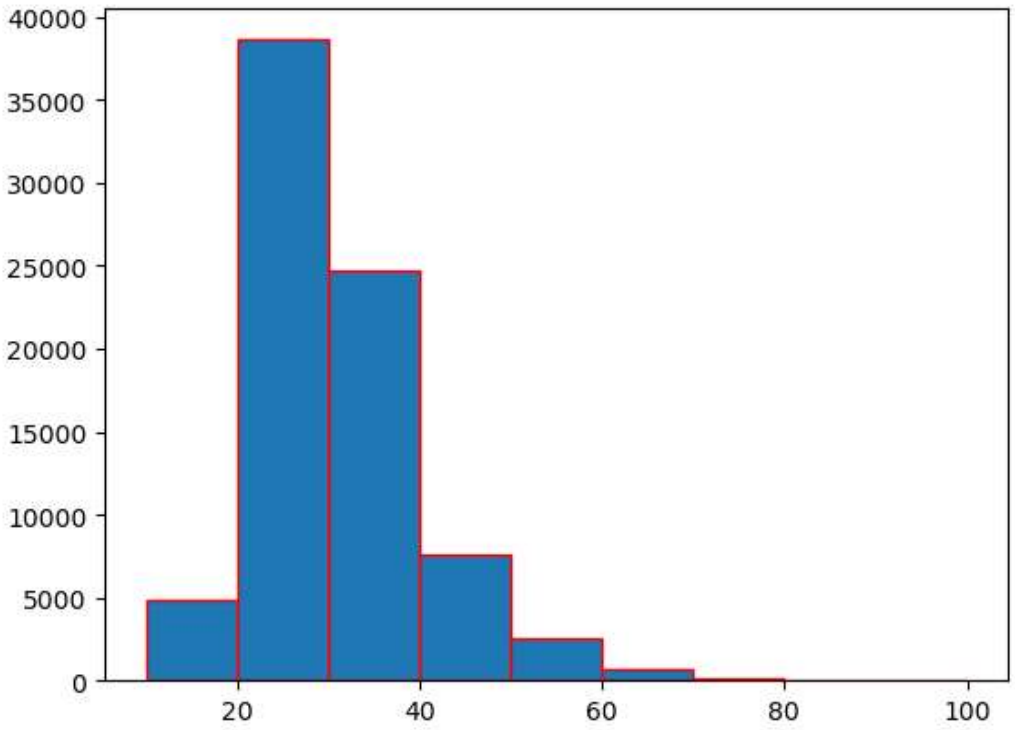
In [40]:
```
#Lood data

viz_data = pd.read_csv(r'C:\Users\LENOVO THINKPAD L540\Desktop\data visualization\data.csv
viz_data
```

Out[40]:

	Responder_id	Age
0	1	14
1	2	19
2	3	28
3	4	22
4	5	30
...	...	...
79205	87352	59
79206	87386	21
79207	87739	25
79208	88212	40
79209	88863	18

79210 rows × 2 columns

In [48]:
```
import pandas as pd
bins = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
plt.hist(viz_data['Age'], bins = bins, edgecolor ='red')
plt.show()
```



In [ ]:

In [ ]: