Importing important libraries

In [ ]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Adding our dataset

In [ ]:
```
dataframe = pd.read_csv('imbd.csv')
```

List of our dataset

In [ ]:
```
dataframe
```

Out[ ]:

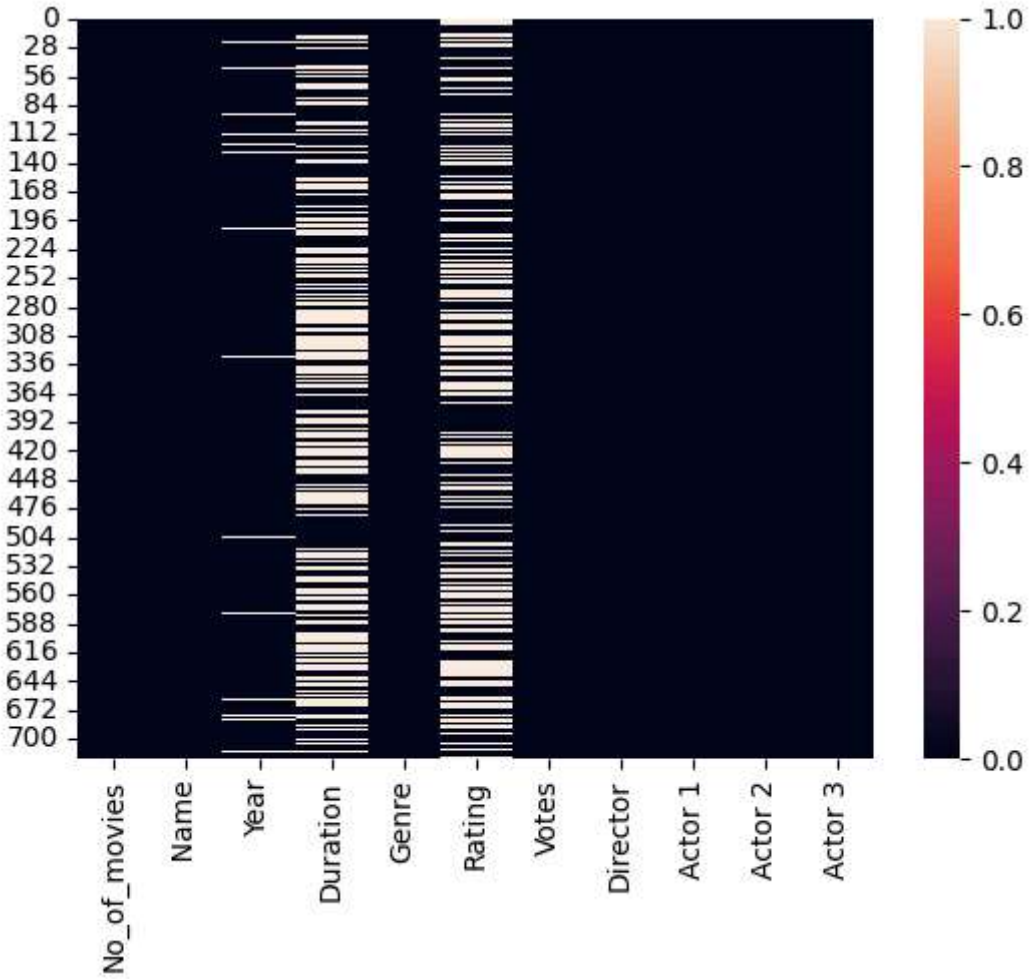
	No_of_movies	Name	Year	Duration	Genre	Rating	Votes	Director	
0	1		NaN	NaN	Drama	NaN	2	J.S. Randhawa	
1	2	#Gadhvi (He thought he was G...	2019	109.0	Drama	NaN	7	Gaurav Bakshi	
2	3	#Homecoming	2021	90.0	Drama, Musical	7.0	8	Soumyajit Majumdar	
3	4	#Yaaram	2019	110.0	Comedy, Romance	NaN	50	Ovais Khan	
4	5	...And Once Again	2010	105.0	Drama	4.4	35	Amol Palekar	
...	...	...	...	...	...	...	...	...	
715	716	Agyaat	2009	130.0	Drama, Horror, Mystery	6.2	0	Ram Gopal Varma	
716	717	Ahaan	2019	81.0	Comedy, Drama	2.9	620	Nikhil Pherwani	A
717	718	Aham Brahmasmi	2021	106.0	Action	7.2	213	Megastar Maharishi Aazaad	
718	719	Ahankaar	1995	145.0	Drama	NaN	0	Ashim S. Samanta	Ch
719	720	Ahankar	1999	75.0	-------	5.9	53	Raja Mitra	

720 rows × 11 columns

There are so many NaN items(not a number)

In [ ]:
```
sns.heatmap(dataframe.isnull())
```

Out[ ]: <Axes: >



In [ ]:
```
dataframe.isnull().sum()
```

Out[ ]:
```
No_of_movies      0
       Name       0
Year             23
Duration        348
Genre             0
Rating          298
Votes             0
Director          0
Actor 1           0
Actor 2           0
Actor 3           0
dtype: int64
```

In [ ]:
```
dataframe.shape
```

Out[ ]:
```
(720, 11)
```

In [ ]:
```
dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 720 entries, 0 to 719
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   No_of_movies  720 non-null    int64  
 1          Name   720 non-null    object 
 2   Year          697 non-null    object 
 3   Duration      372 non-null    float64
 4   Genre         720 non-null    object 
 5   Rating        422 non-null    float64
 6   Votes         720 non-null    int64  
 7   Director      720 non-null    object 
 8   Actor 1       720 non-null    object 
 9   Actor 2       720 non-null    object 
 10  Actor 3       720 non-null    object 
dtypes: float64(2), int64(2), object(7)
memory usage: 62.0+ KB
```

So Many errors

Lets try to remove errors(NaN values)

I have made 'Year' column clear here

In [ ]:
```
dataframe['Year'] = pd.to_numeric(dataframe['Year'], errors='coerce')
```

Here we will make a new variable which is tdf (True Dataframe)

In [ ]:
```
true_dataframe = dataframe.dropna()
```

I removed the error by using [dropna]

In [ ]:
```
true_dataframe
```

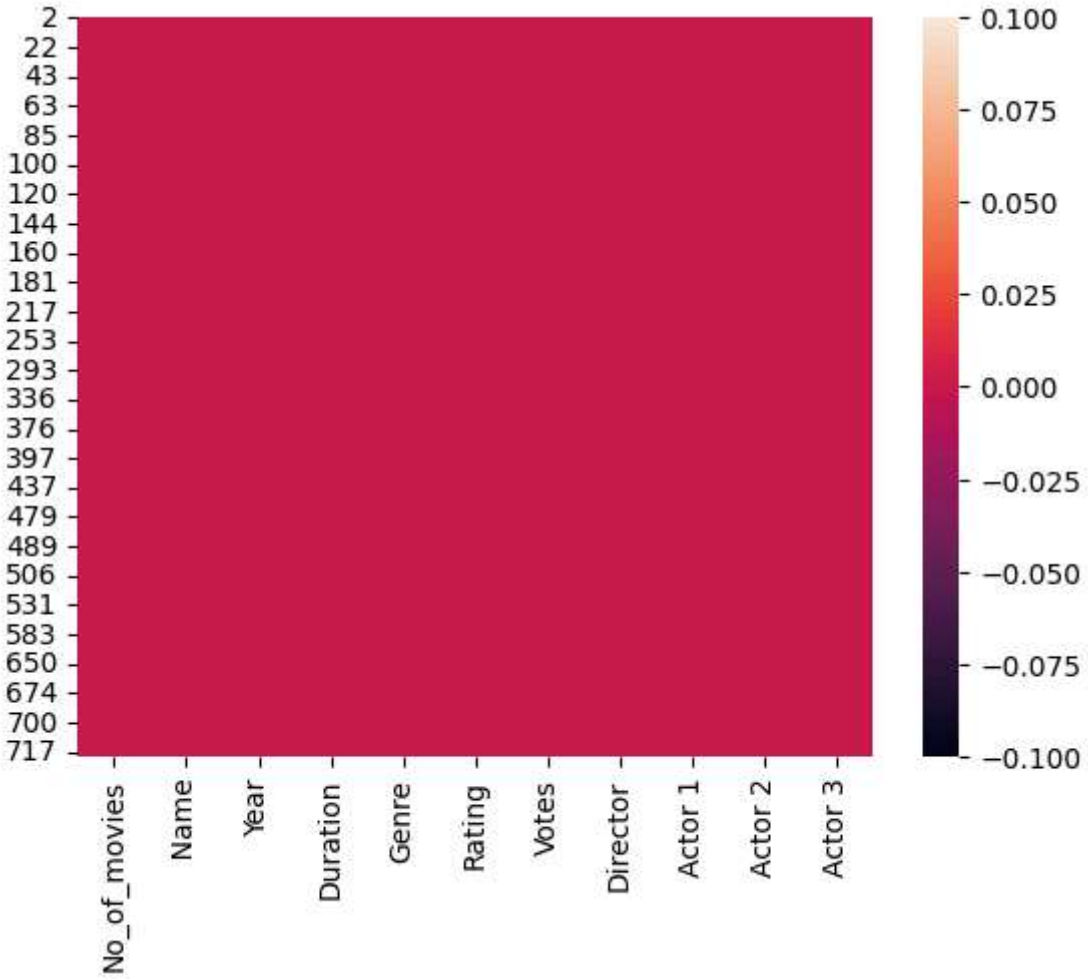
Out[ ]:

	No_of_movies	Name	Year	Duration	Genre	Rating	Votes	Directo
2	3	#Homecoming	2021.0	90.0	Drama, Musical	7.0	8	Soumyaj Majumda
4	5	...And Once Again	2010.0	105.0	Drama	4.4	35	Amo Paleka
6	7	...Yahaan	2005.0	142.0	Drama, Romance, War	4.7	827	Shooj Sirca
7	8	.in for Motion	2008.0	59.0	Documentary	7.4	1086	Anirba Datt
9	10	@Andheri	2014.0	116.0	Action, Crime, Thriller	5.6	326	Bij Bhaska Na
...	...	...	...	...	...	...	...	
714	715	Agreement	1980.0	124.0	Musical, Comedy, Drama	4.8	0	An Gangul
715	716	Agyaat	2009.0	130.0	Drama, Horror, Mystery	6.2	0	Ran Gopa Varm
716	717	Ahaan	2019.0	81.0	Comedy, Drama	2.9	620	Nikh Pherwar
717	718	Aham Brahmasmi	2021.0	106.0	Action	7.2	213	Megasta Maharish Aazaa
719	720	Ahankar	1999.0	75.0	-------	5.9	53	Raja Mitr

227 rows × 11 columns

In [ ]:
```
sns.heatmap(true_dataframe.isnull())
```

Out[ ]:
```
<Axes: >
```



In [ ]:
```
true_dataframe.isnull().sum()
```

Out[ ]:
```
No_of_movies    0
       Name     0
Year            0
Duration        0
Genre           0
Rating          0
Votes           0
Director        0
Actor 1         0
Actor 2         0
Actor 3         0
dtype: int64
```

In [ ]:
```
true_dataframe.shape
```

Out[ ]:
```
(227, 11)
```

In [ ]:
```
true_dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 227 entries, 2 to 719
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   No_of_movies  227 non-null    int64
 1          Name   227 non-null    object
 2   Year          227 non-null    float64
 3   Duration      227 non-null    float64
 4   Genre         227 non-null    object
 5   Rating        227 non-null    float64
 6   Votes         227 non-null    int64
 7   Director      227 non-null    object
 8   Actor 1       227 non-null    object
 9   Actor 2       227 non-null    object
 10  Actor 3       227 non-null    object
dtypes: float64(3), int64(2), object(6)
memory usage: 21.3+ KB
```

Now we will check that are there any duplicate values or not

In [ ]:
```
dup_true_dataframe=true_dataframe.duplicated().any()
```

In [ ]:
```
print("is there any duplicate value? : ",dup_true_dataframe)
```

```
is there any duplicate value? :  False
```

This means our dataset is clear now

I am making another variable for this [ddf(duplicated dataframe)]

In [ ]:
```
duplicate_dataframe = true_dataframe.drop_duplicates()
duplicate_dataframe
```

Out[ ]:

	No_of_movies	Name	Year	Duration	Genre	Rating	Votes	Directo
2	3	#Homecoming	2021.0	90.0	Drama, Musical	7.0	8	Soumyaj Majumda
4	5	...And Once Again	2010.0	105.0	Drama	4.4	35	Amo Paleka
6	7	...Yahaan	2005.0	142.0	Drama, Romance, War	4.7	827	Shooj Sirca
7	8	.in for Motion	2008.0	59.0	Documentary	7.4	1086	Anirba Datt
9	10	@Andheri	2014.0	116.0	Action, Crime, Thriller	5.6	326	Bij Bhaska Na
...	...	...	...	...	...	...	...	
714	715	Agreement	1980.0	124.0	Musical, Comedy, Drama	4.8	0	An Gangul
715	716	Agyaat	2009.0	130.0	Drama, Horror, Mystery	6.2	0	Ran Gopa Varm
716	717	Ahaan	2019.0	81.0	Comedy, Drama	2.9	620	Nikh Pherwar
717	718	Aham Brahmasmi	2021.0	106.0	Action	7.2	213	Megasta Maharish Aazaa
719	720	Ahankar	1999.0	75.0	-------	5.9	53	Raja Mitr

227 rows × 11 columns

In [ ]:
```
duplicate_dataframe.describe()
```

Out[ ]:

	No_of_movies	Year	Duration	Rating	Votes
count	227.000000	227.000000	227.000000	227.000000	227.000000
mean	331.215859	1996.964758	128.251101	5.661233	930.118943
std	220.755966	19.679930	28.405151	1.329413	3392.521840
min	3.000000	1938.000000	45.000000	2.500000	0.000000
25%	127.500000	1985.000000	110.000000	4.700000	14.500000
50%	329.000000	2001.000000	130.000000	5.700000	61.000000
75%	502.500000	2013.500000	148.000000	6.700000	338.000000
max	720.000000	2021.000000	240.000000	8.100000	27357.000000

In [ ]:
```
duplicate_dataframe.describe(include='all')
```

Out[ ]:

	No_of_movies	Name	Year	Duration	Genre	Rating	Votes
count	227.000000	227	227.000000	227.000000	227	227.000000	227.000000
unique	NaN	209	NaN	NaN	89	NaN	NaN
top	NaN		NaN	NaN	Drama	NaN	NaN
freq	NaN	6	NaN	NaN	27	NaN	NaN
mean	331.215859	NaN	1996.964758	128.251101	NaN	5.661233	930.118943
std	220.755966	NaN	19.679930	28.405151	NaN	1.329413	3392.521840
min	3.000000	NaN	1938.000000	45.000000	NaN	2.500000	0.000000
25%	127.500000	NaN	1985.000000	110.000000	NaN	4.700000	14.500000
50%	329.000000	NaN	2001.000000	130.000000	NaN	5.700000	61.000000
75%	502.500000	NaN	2013.500000	148.000000	NaN	6.700000	338.000000
max	720.000000	NaN	2021.000000	240.000000	NaN	8.100000	27357.000000

Our data is clear now

In [ ]:
```
clear_data = duplicate_dataframe
```

Now movies which are less than 90 minute

In [ ]:
```
duplicate_dataframe[duplicate_dataframe['Duration']<=90]
```

Out[ ]:

	No_of_movies	Name	Year	Duration	Genre	Rating	Votes	Direct
2	3	#Homecoming	2021.0	90.0	Drama, Musical	7.0	8	Soumya Majumd
7	8	.in for Motion	2008.0	59.0	Documentary	7.4	1086	Anirba Dat
50	51	2 Nights in Soul Valley	2012.0	80.0	Adventure, Horror, Mystery	5.6	14	Hari Sharn
95	96	417 Miles	2016.0	75.0	Drama	7.3	2548	Main Dh
107	108	6387 Meters Black Peak	2020.0	85.0	Documentary	5.4	5	Kov Mitt
134	135	99.9 FM	2005.0	90.0	Crime, Drama	7.4	1901	Sanj Bhat
135	136	A Ballad of Maladies	2016.0	86.0	Documentary, Music	4.8	38	Sarvr Ka
144	145	A Dream Document	2016.0	70.0	Documentary	7.5	4637	Rup D
154	155	A Mermaid Called Aida	1996.0	52.0	Documentary	7.3	17	Riy Vir Wac
168	169	A Suitable Girl	2017.0	90.0	Documentary	3.8	627	Sari Khura
192	193	Aadamkhor	2018.0	72.0	Horror	5.5	6	Kshi Sharn
218	219	Aadmi Ki Aurat Aur Anya Kahaniya	2009.0	78.0	Drama	5.0	142	An Dut
253	254	Aage Maut Peeche Maut	2001.0	78.0	Horror	4.8	30	A.T. J
273	274	Aaina	2019.0	60.0	Horror	3.2	5	Ra Bharg
280	281	Aaj Ka Andha Kanoon	2003.0	46.0	Action	6.0	5	An Chand Sah
377	378	Aamna Saamna	2013.0	62.0	Animation	7.7	10862	Ra Chila
494	495	Aasmaan Se Gira	1992.0	88.0	Adventure, Comedy, Romance	5.5	6	Pank Parash

	No_of_movies	Name	Year	Duration	Genre	Rating	Votes	Direct
530	531	Ab Hoga Dharna Unlimited	2012.0	78.0	Comedy	7.6	80	Deep Tanw
621	622	Adhiktam	2020.0	45.0	Action, Comedy	4.9	21	Prasha Sing
656	657	Adwait Sangeet	2011.0	82.0	Documentary, Biography, Musical	4.6	140	Makarar Brahn
716	717	Ahaan	2019.0	81.0	Comedy, Drama	2.9	620	Nikl Pherwa
719	720	Ahankar	1999.0	75.0	-------	5.9	53	Raja Mit

This shows that total 23 movies are less than 90 minute

Lets check for greater than 200 minute

In [ ]:
```
duplicate_dataframe[duplicate_dataframe['Duration']>=200]
```

Out[ ]:

	No_of_movies	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1
490	491	Aasma: The Sky Is the Limit	2009.0	240.0	Drama	5.2	8	Rohit Krishnakant Nayyar	Hrishitaa Bhatt

This shows that total 1 movie is equal or greater than 200 minute

This means that 204 movies are between 90 and 200 minuts

Lets find out that how many movies were made between 1940 to 2023

In [ ]:
```
duplicate_dataframe[duplicate_dataframe['Year']<=1940]
```

Out[ ]:

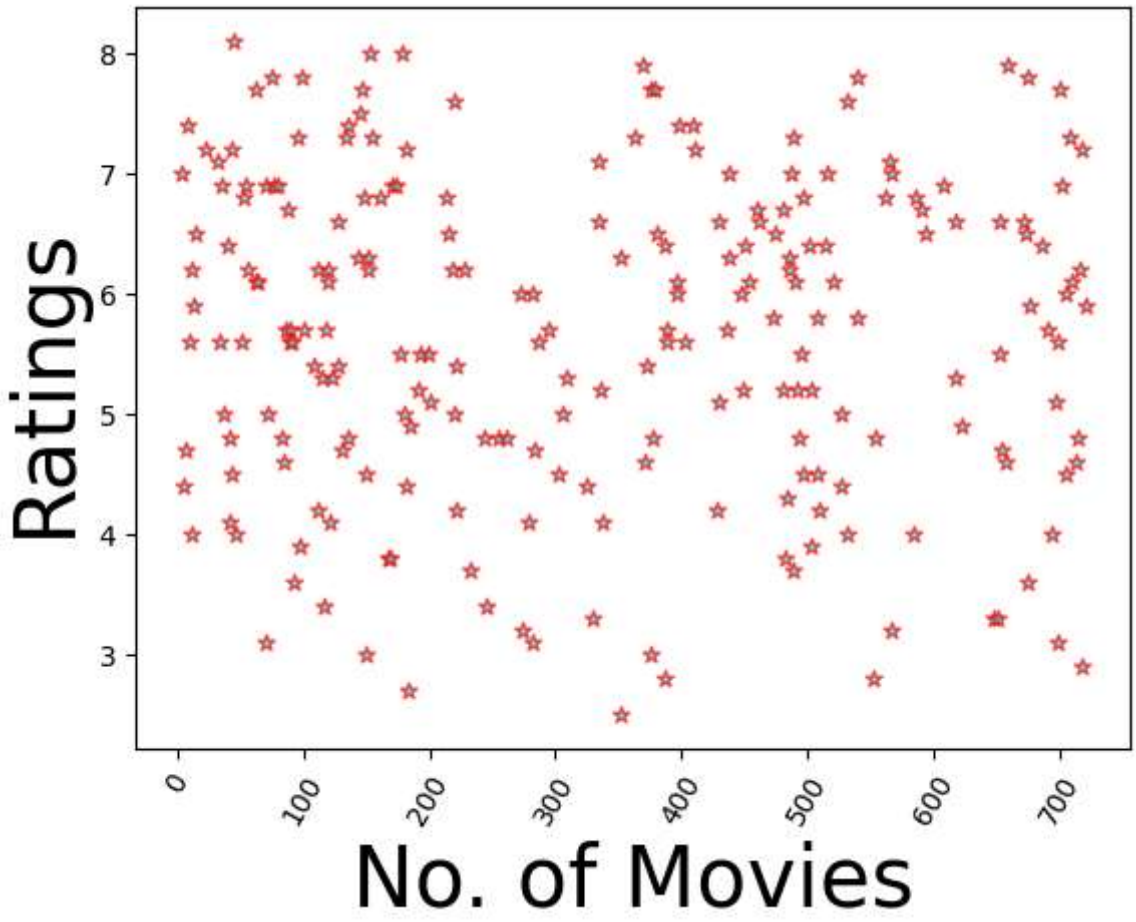
	No_of_movies	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1
552	553	Abhagin	1938.0	151.0	-------	4.8	21	Prafulla Roy	Molina Devi
561	562	Abhilasha	1938.0	134.0	-------	6.8	0	Zia Sarhadi	Mahendra Thakore
590	591	Achhut	1940.0	140.0	Drama	6.7	25	Gohar	Motilal
617	618	Adhikar	1938.0	132.0	-------	6.6	28	P.C. Barua	P.C. Barua

This shows that the oldest movie was made in 1938

Now we will make a graph

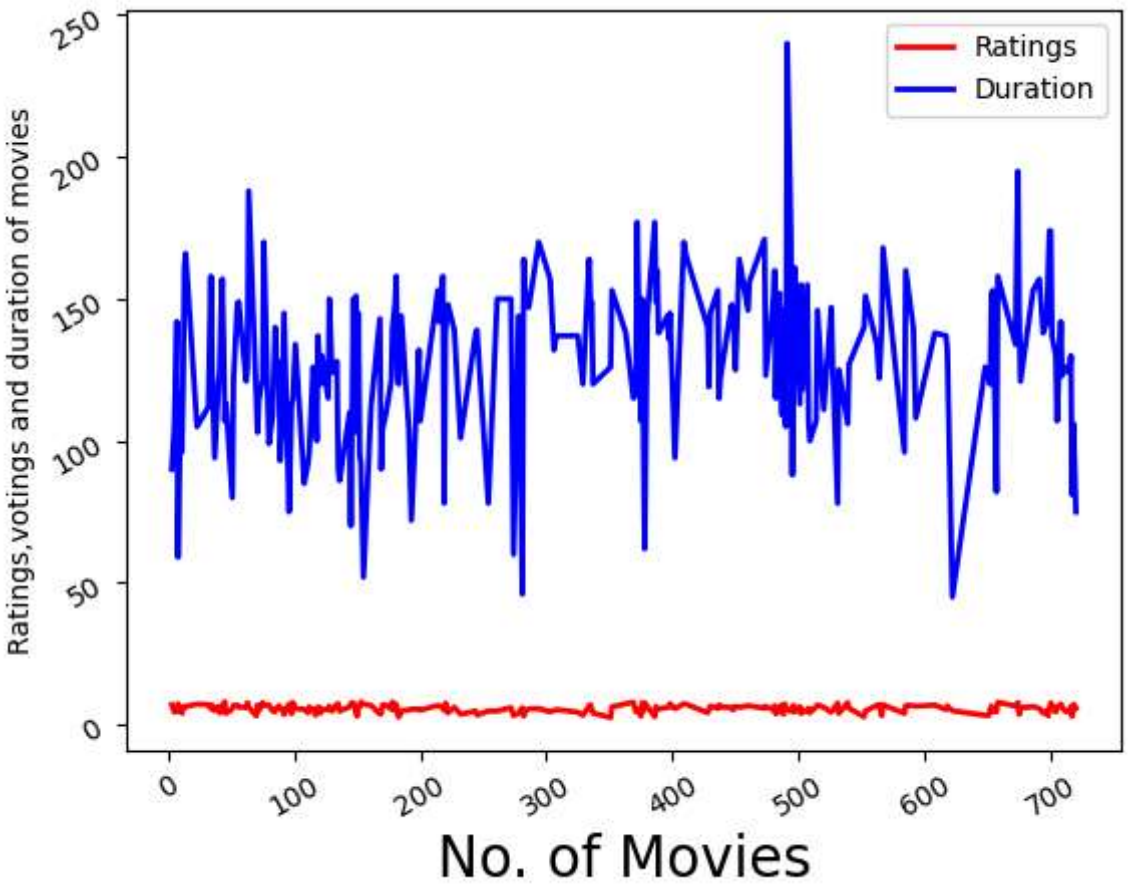
In [ ]:
```
w = clear_data.Votes
x = clear_data.No_of_movies
y = clear_data.Rating
z = clear_data.Duration
```

In [ ]:
```
plt.scatter(x,y,marker='*',c='cyan',ec='red',alpha=0.7)
plt.xticks(rotation=60)
plt.xlabel("No. of Movies",fontsize='30')
plt.ylabel("Ratings",fontsize="30")
plt.show()
```

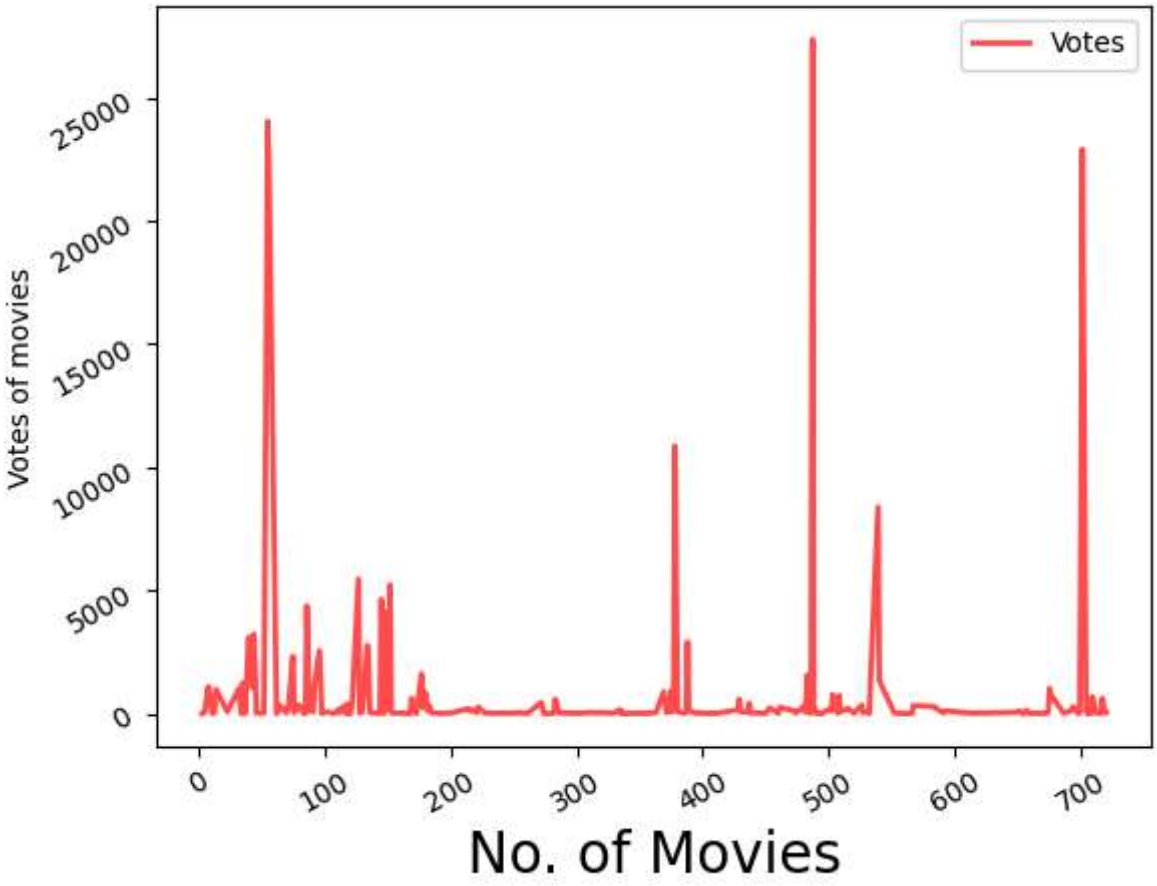


Now Line Plots

In [ ]:
```
plt.plot(x,y,label='Ratings',c='red',lw=2)
plt.plot(x,z,label='Duration',c='blue',lw=2)
plt.xlabel('No. of Movies',fontsize='20')
plt.ylabel("Ratings,votings and duration of movies",fontsize='10')
plt.xticks (rotation=30)
plt.yticks (rotation=30)
plt.legend()
plt.show()
```

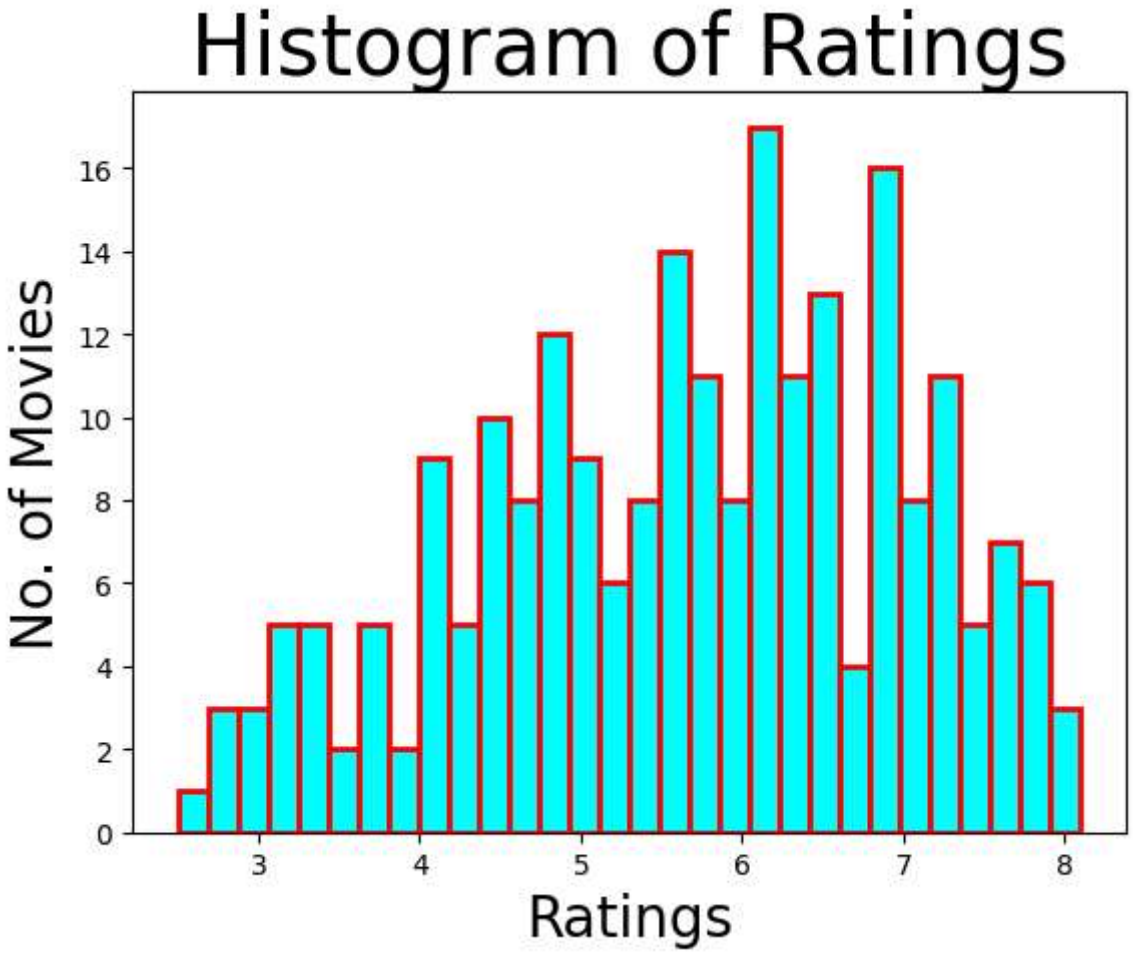


In [ ]:
```
plt.plot(x,w,label='Votes',c='red',alpha=0.7,lw=2)
plt.xlabel('No. of Movies',fontsize='20')
plt.ylabel("Votes of movies",fontsize='10')
plt.xticks (rotation=30)
plt.yticks (rotation=30)
plt.legend()
plt.show()
```



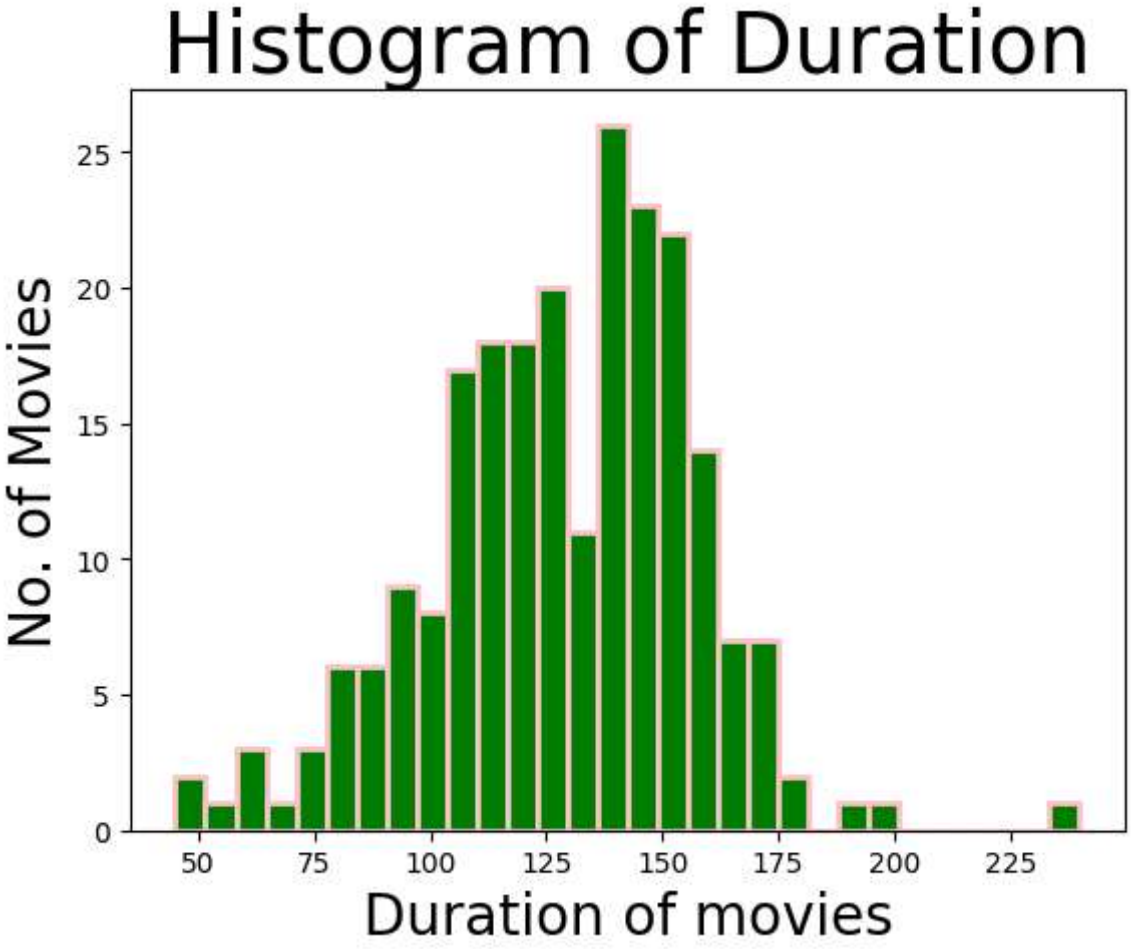
Now Histogram of Ratings

In [ ]:
```
plt.hist(y,bins=30, color="cyan",ec="red", lw=2,)
plt.title("Histogram of Ratings", fontsize=30)
plt.xlabel("Ratings", fontsize=20)
plt.ylabel("No. of Movies",fontsize=20)
plt.show()
```



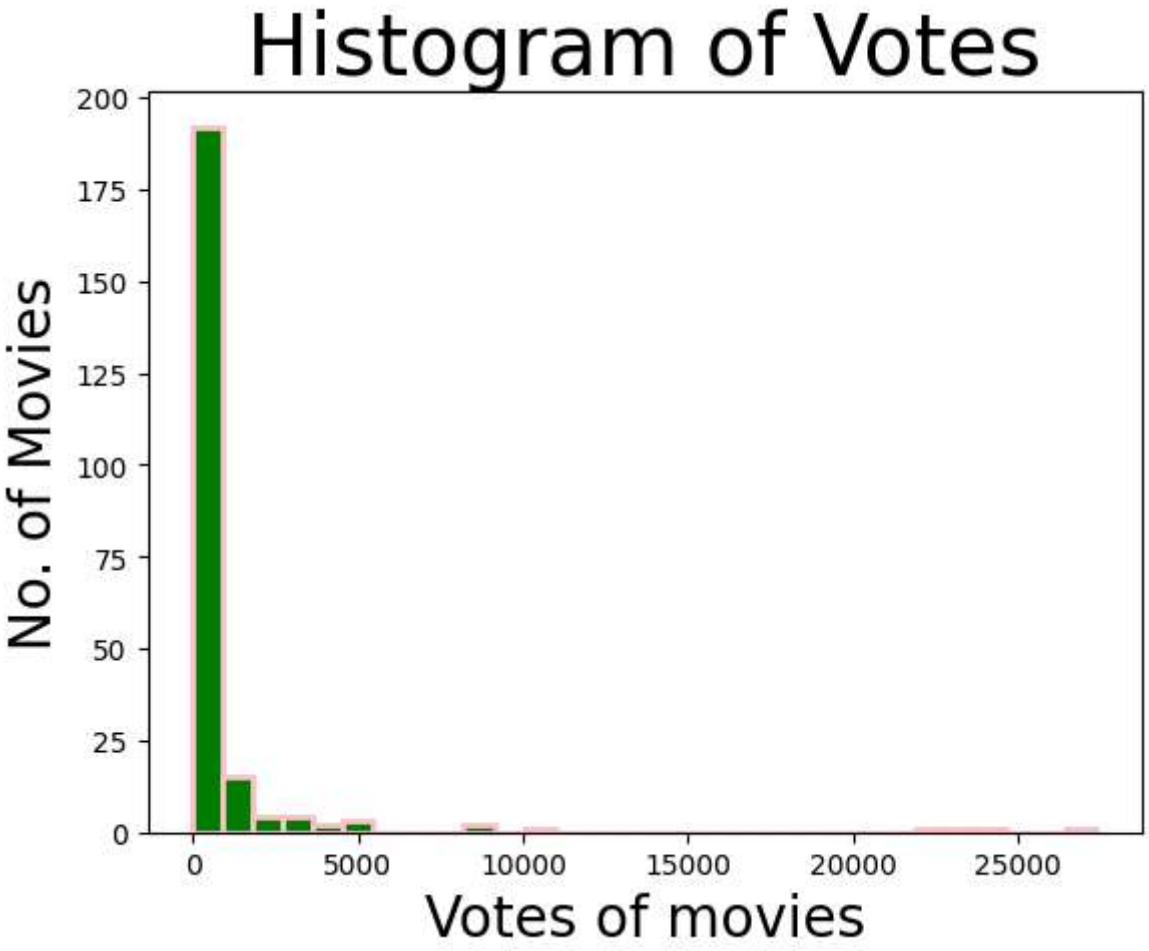
Now Histogram of Duration

In [ ]:
```
plt.hist(z,bins=30, color="green",ec="pink", lw=2,)
plt.title("Histogram of Duration", fontsize=30)
plt.ylabel("No. of Movies",fontsize=20)
plt.xlabel("Duration of movies", fontsize=20)
plt.show()
```



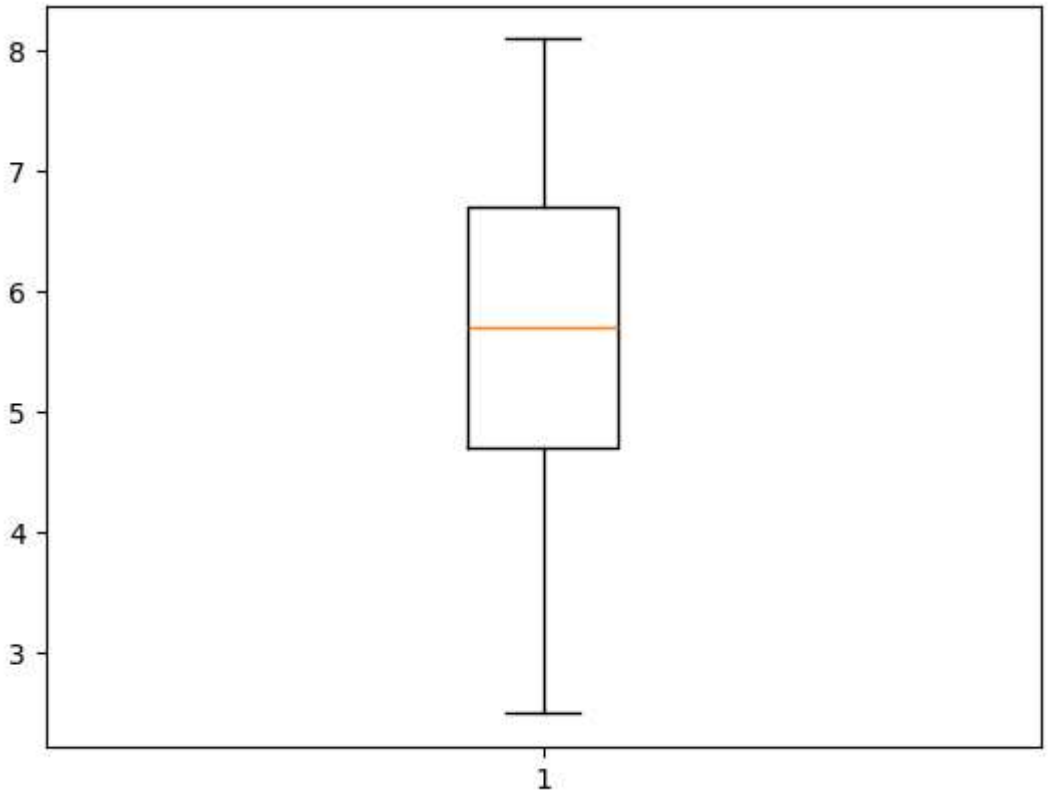
Now Histogram of Votes

In [ ]:
```
plt.hist(w,bins=30, color="green",ec="pink", lw=2,)
plt.title("Histogram of Votes", fontsize=30)
plt.ylabel("No. of Movies",fontsize=20)
plt.xlabel("Votes of movies", fontsize=20)
plt.show()
```

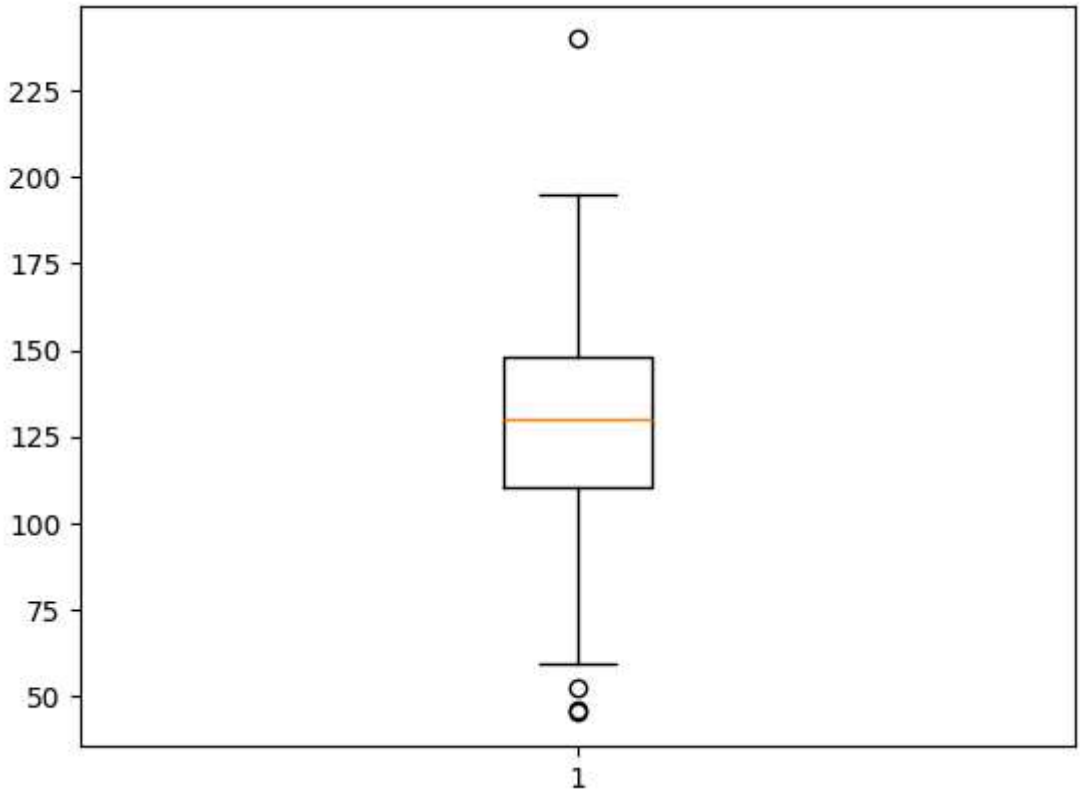


Now Box plot

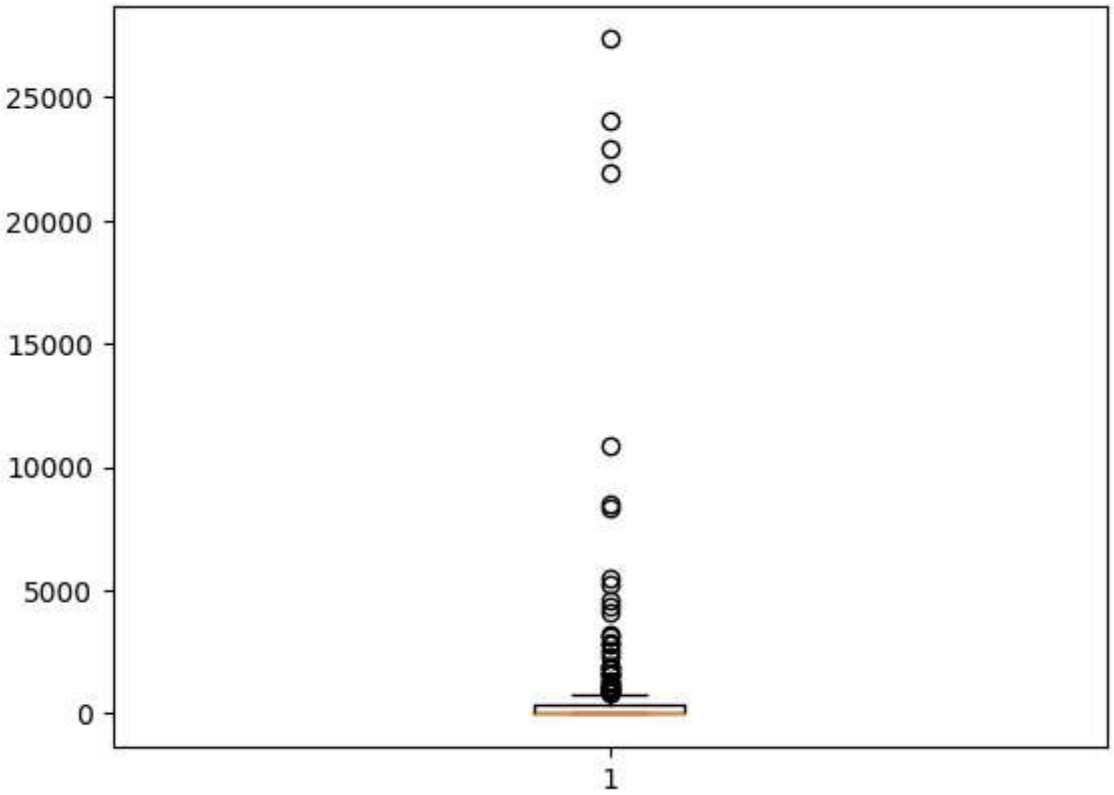
In [ ]:
```
plt.boxplot(y)
plt.show()
```



In [ ]:
```
plt.boxplot(z)
plt.show()
```



In [ ]:
```
plt.boxplot(w)
plt.show()
```


25000
20000
15000
10000
5000
0
1