Mini Project

Aim : Implement operations on Movies Dataset (Dataset link : https://github.com/rashida048/Some-NLP-Projects/blob/master/movie_dataset.csv)

Code :

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pandas import DataFrame, Series
```

In [2]:
```
df1 = pd.read_csv("movies.csv")
df1
```

Out[2]:

	index	budget	genres	homepage	
0	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	
2	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	2
3	3	250000000	Action Crime Drama Thriller	http://www.thedarkknightrises.com/	
4	4	260000000	Action Adventure Science Fiction	http://movies.disney.com/john-carter	
...	...	...	...	...	
4798	4798	220000	Action Crime Thriller	NaN	
4799	4799	9000	Comedy Romance	NaN	
4800	4800	0	Comedy Drama Romance TV Movie	http://www.hallmarkchannel.com/signedsealeddel...	2
4801	4801	0	NaN	http://shanghaicalling.com/	1

	index	budget	genres	homepage
4802	4802	0	Documentary	NaN

4803 rows × 24 columns

In [3]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   index                 4803 non-null   int64
 1   budget                4803 non-null   int64
 2   genres                4775 non-null   object
 3   homepage              1712 non-null   object
 4   id                    4803 non-null   int64
 5   keywords              4391 non-null   object
 6   original_language     4803 non-null   object
 7   original_title        4803 non-null   object
 8   overview              4800 non-null   object
 9   popularity            4803 non-null   float64
 10  production_companies  4803 non-null   object
 11  production_countries  4803 non-null   object
 12  release_date          4802 non-null   object
 13  revenue               4803 non-null   int64
 14  runtime               4801 non-null   float64
 15  spoken_languages      4803 non-null   object
 16  status                4803 non-null   object
 17  tagline               3959 non-null   object
 18  title                 4803 non-null   object
 19  vote_average          4803 non-null   float64
 20  vote_count            4803 non-null   int64
 21  cast                  4760 non-null   object
 22  crew                  4803 non-null   object
 23  director              4773 non-null   object
dtypes: float64(3), int64(5), object(16)
memory usage: 900.7+ KB
```

In [4]:
```
df1.describe()
```

Out[4]:

	index	budget	id	popularity	revenue	runtin
count	4803.000000	4.803000e+03	4803.000000	4803.000000	4.803000e+03	4801.0000
mean	2401.000000	2.904504e+07	57165.484281	21.492301	8.226064e+07	106.8758
std	1386.651002	4.072239e+07	88694.614033	31.816650	1.628571e+08	22.6119
min	0.000000	0.000000e+00	5.000000	0.000000	0.000000e+00	0.0000
25%	1200.500000	7.900000e+05	9014.500000	4.668070	0.000000e+00	94.0000
50%	2401.000000	1.500000e+07	14629.000000	12.921594	1.917000e+07	103.0000
75%	3601.500000	4.000000e+07	58610.500000	28.313505	9.291719e+07	118.0000
max	4802.000000	3.800000e+08	459488.000000	875.581305	2.787965e+09	338.0000

In [5]:
```
df1.head()
```

Out[5]:

	index	budget	genres	homepage	id	key
0	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	19995	spa
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	285	eas
2	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	206647	spy or
3	3	250000000	Action Crime Drama Thriller	http://www.thedarkknightrises.com/	49026	dc te
4	4	260000000	Action Adventure Science Fiction	http://movies.disney.com/john-carter	49529	ba me

5 rows × 24 columns

In [6]:
```
df1.tail()
```

Out[6]:

	index	budget	genres	homepage	
4798	4798	220000	Action Crime Thriller	NaN	93
4799	4799	9000	Comedy Romance	NaN	727
4800	4800	0	Comedy Drama Romance TV Movie	http://www.hallmarkchannel.com/signedsealeddel...	2316
4801	4801	0	NaN	http://shanghaicalling.com/	1261
4802	4802	0	Documentary	NaN	259

5 rows × 24 columns

In [12]:
```
top_left_corner_df = df1.iloc[:4, :4]
print(top_left_corner_df)
```

```
   index     budget                                genres  \
0      0  237000000  Action Adventure Fantasy Science Fiction
1      1  300000000                  Adventure Fantasy Action
2      2  245000000                    Action Adventure Crime
3      3  250000000                Action Crime Drama Thriller

                                     homepage
0                   http://www.avatarmovie.com/
1  http://disney.go.com/disneypictures/pirates/
2   http://www.sonypictures.com/movies/spectre/
3            http://www.thedarkknightrises.com/
```

In [13]:
```
df1.to_csv()
```

Out[13]:

',index,budget,genres,homepage,id,keywords,original_language,original_title,ove
rview,popularity,production_companies,production_countries,release_date,revenu
e,runtime,spoken_languages,status,tagline,title,vote_average,vote_count,cast,cr
ew,director\r\n0,0,237000000,Action Adventure Fantasy Science Fiction,http://ww
w.avatarmovie.com/,19995,culture clash future space war space colony society,e
n,Avatar,"In the 22nd century, a paraplegic Marine is dispatched to the moon Pa
ndora on a unique mission, but becomes torn between following orders and protec
ting an alien civilization.",150.437577,"[{""name"": ""Ingenious Film Partner
s"", ""id"": 289}, {""name"": ""Twentieth Century Fox Film Corporation"", ""i
d"": 306}, {""name"": ""Dune Entertainment"", ""id"": 444}, {""name"": ""Lights
torm Entertainment"", ""id"": 574}]","[{""iso_3166_1"": ""US"", ""name"": ""Uni
ted States of America""}, {""iso_3166_1"": ""GB"", ""name"": ""United Kingdo
m""}]",2009-12-10,2787965087,162.0,"[{""iso_639_1"": ""en"", ""name"": ""Englis
h""}, {""iso_639_1"": ""es"", ""name"": ""Espa\\u00f1ol""}]",Released,Enter the
World of Pandora.,Avatar,7.2,11800,Sam Worthington Zoe Saldana Sigourney Weaver
Stephen Lang Michelle Rodriguez,"[{\'name\': \'Stephen E. Rivkin\', \'gender\':
0, \'department\': \'Editing\', \'job\': \'Editor\', \'credit_id\': \'52fe48009
251416c750aca23\', \'id\': 1721}, {\'name\': \'Rick Carter\', \'gender\': 2,
\'department\': \'Art\', \'job\': \'Production Design\', \'credit_id\': \'539c4
7ecc3a36810e3001f87\', \'id\': 496}, {\'name\': \'Christopher Boyes\', \'gender
\': 0, \'department\': \'Sound\', \'job\': \'Sound Designer\', \'credit_id\':
\'54491c89c3a3680fb4001cf7\', \'id\': 900}, {\'name\': \'Christopher Boyes\',
\'gender\': 0, \'department\': \'Sound\', \'job\': \'Supervising Sound Editor
\', \'credit_id\': \'54491cb70e0a267480001bd0\', \'id\': 900}, {\'name\': \'Mal
i Finn\', \'gender\': 1, \'department\': \'Production\', \'job\': \'Casting\',
\'credit_id\': \'539c4a4cc3a36810c9002101\', \'id\': 1262}, {\'name\': \'James
Horner\', \'gender\': 2, \'department\': \'Sound\', \'job\': \'Original Music C
omposer\', \'credit_id\': \'5544ee3b925141499f0008fc\', \'id\': 1729}, {\'name
\': \'James Cameron\', \'gender\': 2, \'department\': \'Directing\', \'job\':
\'Director\', \'credit_id\': \'52fe48009251416c750ac9c3\', \'id\': 2710}, {\'na
me\': \'James Cameron\', \'gender\': 2, \'department\': \'Writing\', \'job\':
\'Writer\', \'credit_id\': \'52fe48009251416c750ac9d9\', \'id\': 2710}, {\'name
\': \'James Cameron\', \'gender\': 2, \'department\': \'Editing\', \'job\': \'E
ditor\', \'credit_id\': \'52fe48009251416c750aca17\', \'id\': 2710}, {\'name\':
\'James Cameron\', \'gender\': 2, \'department\': \'Production\', \'job\': \'Pr
oducer\', \'credit_id\': \'52fe48009251416c750aca29\', \'id\': 2710}, {\'name
\': \'James Cameron\', \'gender\': 2, \'department\': \'Writing\', \'job\': \'S
creenplay\', \'credit_id\': \'52fe48009251416c750aca3f\', \'id\': 2710}, {\'nam
e\': \'Andrew Menzies\', \'gender\': 2, \'department\': \'Art\', \'job\': \'Art
Direction\', \'credit_id\': \'539c4987c3a36810ba0021a4\', \'id\': 7236}, {\'nam
e\': \'Jill Brooks\', \'gender\': 0, \'department\': \'Visual Effects\', \'job
\': \'Visual Effects Producer\', \'credit_id\': \'549598c3c3a3686ae9004383\',
\'id\': 6690}, {\'name\': \'Margery Simkin\', \'gender\': 1, \'department\':
\'Production\', \'job\': \'Casting\', \'credit_id\': \'52fe48009251416c750aca4b
\', \'id\': 6347}, {\'name\': \'Kevin Ishioka\', \'gender\': 2, \'department\':
\'Art\', \'job\': \'Supervising Art Director\', \'credit_id\': \'570b6f41925141
7da70032fe\', \'id\': 6878}, {\'name\': \'Dick Bernstein\', \'gender\': 0, \'de
partment\': \'Sound\', \'job\': \'Music Editor\', \'credit_id\': \'5495a0fac3a3
686ae9004468\', \'id\': 6883}, {\'name\': \'Shannon Mills\', \'gender\': 0, \'d
epartment\': \'Sound\', \'job\': \'Sound Effects Editor\', \'credit_id\': \'549
59706c3a3686af3003e81\', \'id\': 8159}, {\'name\': \'Dennie Thorpe\', \'gender
\': 0, \'department\': \'Sound\', \'job\': \'Foley\', \'credit_id\': \'54491d58
c3a3680fb1001ccb\', \'id\': 8160}, {\'name\': \'Jana Vance\', \'gender\': 0,
\'department\': \'Sound\', \'job\': \'Foley\', \'credit_id\': \'54491d6cc3a3680
fa5001b2c\', \'id\': 8163}, {\'name\': \'Deborah Lynn Scott\', \'gender\': 1,
\'department\': \'Costume & Make-Up\', \'job\': \'Costume Design\', \'credit_id
\': \'52fe48009251416c750aca57\', \'id\': 8527}, {\'name\': \'Jon Landau\', \'g
ender\': 2, \'department\': \'Production\', \'job\': \'Producer\', \'credit_id
\': \'52fe48009251416c750aca2f\', \'id\': 8529}, {\'name\': \'Sean Haworth\',
\'gender\': 0, \'department\': \'Art\', \'job\': \'Art Direction\', \'credit_id

lm"", ""id"": 446}]","[{""iso_3166_1"": ""US"", ""name"": ""United States of Am
erica""}]",2004-10-08,424760,77.0,"[{""iso_639_1"": ""en"", ""name"": ""Englis
h""}]",Released,What happens if it actually works?,Primer,6.9,658,Shane Carruth
David Sullivan Casey Gooden Anand Upadhyaya Carrie Crawford,"[{\'name\': \'Shan
e Carruth\', \'gender\': 2, \'department\': \'Directing\', \'job\': \'Director
\', \'credit_id\': \'52fe45e79251416c75066791\', \'id\': 76624}, {\'name\': \'S
hane Carruth\', \'gender\': 2, \'department\': \'Editing\', \'job\': \'Editor
\', \'credit_id\': \'560a6cfcc3a368681f001dfe\', \'id\': 76624}, {\'name\': \'S
hane Carruth\', \'gender\': 2, \'department\': \'Production\', \'job\': \'Produ
cer\', \'credit_id\': \'52fe45e79251416c75066797\', \'id\': 76624}, {\'name\':
\'Shane Carruth\', \'gender\': 2, \'department\': \'Writing\', \'job\': \'Write
r\', \'credit_id\': \'52fe45e79251416c750667af\', \'id\': 76624}, {\'name\':
\'Shane Carruth\', \'gender\': 2, \'department\': \'Sound\', \'job\': \'Music
\', \'credit_id\': \'52fe45e79251416c7506679d\', \'id\': 76624}, {\'name\': \'S
hane Carruth\', \'gender\': 2, \'department\': \'Crew\', \'job\': \'Cinematogra
phy\', \'credit_id\': \'560a6ba2c3a3686824001e85\', \'id\': 76624}]",Shane Carr
uth\r\n4797,4797,0,Foreign Thriller,,67238,,en,Cavite,"Adam, a security guard,
travels from California to the Philippines, his native land, for his father\'s
funeral. He arrives in Manila. As he waits, a phone rings in his backpack; he a
nswers it, and a male voice tells him that his mother and sister are captives a
nd will be killed if Adam doesn\'t cooperate. Over the next hour, the voice sen
ds Adam by bus, taxi, motorized tricycle, and on foot through an urban landscap
e of busy streets, cramped apartments, a fetid squatters\' camp, a bank, a cock
fighting arena, and a church. Adam\'s conversations with the voice cover murde
r, Islam, jihad, rebellion in Mindanao, and his family. What is it Adam will be
commanded to do?",0.022173,[],[],2005-03-12,0,80.0,[],Released,,Cavite,7.5,2,,"
[{\'name\': \'Neill Dela Llana\', \'gender\': 0, \'department\': \'Directing\',
\'job\': \'Director\', \'credit_id\': \'52fe475dc3a368484e0c319f\', \'id\': 117
4437}, {\'name\': \'Ian Gamazon\', \'gender\': 0, \'department\': \'Directing
\', \'job\': \'Director\', \'credit_id\': \'52fe475dc3a368484e0c31a5\', \'id\':
1174438}]",Neill Dela Llana\r\n4798,4798,220000,Action Crime Thriller,,9367,uni
ted states\\u2013mexico barrier legs arms paper knife guitar case,es,El Mariach
i,"El Mariachi just wants to play his guitar and carry on the family tradition.
Unfortunately, the town he tries to find work in has another visitor...a killer
who carries his guns in a guitar case. The drug lord and his henchmen mistake E
l Mariachi for the killer, Azul, and chase him around town trying to kill him a
nd get his guitar case.",14.269792,"[{""name"": ""Columbia Pictures"", ""id"":
5}]","[{""iso_3166_1"": ""MX"", ""name"": ""Mexico""}, {""iso_3166_1"": ""US"",
""name"": ""United States of America""}]",1992-09-04,2040920,81.0,"[{""iso_639_
1"": ""es"", ""name"": ""Espa\\u00f1ol""}]",Released,"He didn\'t come looking f
or trouble, but trouble came looking for him.",El Mariachi,6.6,238,Carlos Galla
rdo Jaime de Hoyos Peter Marquardt Reinol Martinez Ramiro Gomez,"[{\'name\':
\'Robert Rodriguez\', \'gender\': 0, \'department\': \'Directing\', \'job\':
\'Director\', \'credit_id\': \'52fe44eec3a36847f80b280b\', \'id\': 2294}, {\'na
me\': \'Robert Rodriguez\', \'gender\': 0, \'department\': \'Camera\', \'job\':
\'Director of Photography\', \'credit_id\': \'52fe44eec3a36847f80b2847\', \'id
\': 2294}, {\'name\': \'Robert Rodriguez\', \'gender\': 0, \'department\': \'Ed
iting\', \'job\': \'Editor\', \'credit_id\': \'52fe44eec3a36847f80b284d\', \'id
\': 2294}, {\'name\': \'Robert Rodriguez\', \'gender\': 0, \'department\': \'Pr
oduction\', \'job\': \'Producer\', \'credit_id\': \'52fe44eec3a36847f80b281d\',
\'id\': 2294}, {\'name\': \'Robert Rodriguez\', \'gender\': 0, \'department\':
\'Writing\', \'job\': \'Author\', \'credit_id\': \'52fe44eec3a36847f80b2811\',
\'id\': 2294}, {\'name\': \'Carlos Gallardo\', \'gender\': 2, \'department\':
\'Production\', \'job\': \'Producer\', \'credit_id\': \'52fe44eec3a36847f80b281
7\', \'id\': 20498}, {\'name\': \'Eric Guthrie\', \'gender\': 0, \'department
\': \'Sound\', \'job\': \'Music\', \'credit_id\': \'52fe44eec3a36847f80b2829\',
\'id\': 57473}, {\'name\': \'Chris Knudson\', \'gender\': 0, \'department\':
\'Sound\', \'job\': \'Music\', \'credit_id\': \'52fe44eec3a36847f80b282f\', \'i
d\': 57474}, {\'name\': \'\\\\u00c1lvaro Rodr\\\\u00edguez\', \'gender\': 0,
\'department\': \'Sound\', \'job\': \'Music\', \'credit_id\': \'52fe44eec3a3684

7f80b2835\', \'id\': 57475}, {\'name\': \'Cecilio Rodr\\\\u00edguez\', \'gender
\': 0, \'department\': \'Sound\', \'job\': \'Music\', \'credit_id\': \'52fe44ee
c3a36847f80b283b\', \'id\': 57478}, {\'name\': \'Mark Trujillo\', \'gender\':
0, \'department\': \'Sound\', \'job\': \'Music\', \'credit_id\': \'52fe44eec3a3
6847f80b2841\', \'id\': 57479}]",Robert Rodriguez\r\n4799,4799,9000,Comedy Roma
nce,,72766,,en,Newlyweds,A newlywed couple\'s honeymoon is upended by the arriv
als of their respective sisters.,0.642552,[],[],2011-12-26,0,85.0,[],Released,A
newlywed couple\'s honeymoon is upended by the arrivals of their respective sis
ters.,Newlyweds,5.9,5,Edward Burns Kerry Bish\\u00e9 Marsha Dietlein Caitlin Fi
tzgerald Daniella Pineda,"[{\'name\': \'Edward Burns\', \'gender\': 2, \'depart
ment\': \'Directing\', \'job\': \'Director\', \'credit_id\': \'52fe487dc3a36848
4e0fb013\', \'id\': 12833}, {\'name\': \'Edward Burns\', \'gender\': 2, \'depar
tment\': \'Production\', \'job\': \'Producer\', \'credit_id\': \'584a9e89c3a368
141101e3b6\', \'id\': 12833}, {\'name\': \'Edward Burns\', \'gender\': 2, \'dep
artment\': \'Writing\', \'job\': \'Writer\', \'credit_id\': \'52fe487dc3a368484
e0fb019\', \'id\': 12833}, {\'name\': \'William Rexer\', \'gender\': 0, \'depar
tment\': \'Production\', \'job\': \'Producer\', \'credit_id\': \'584a9ead925141
57fa00ccae\', \'id\': 1066767}, {\'name\': \'William Rexer\', \'gender\': 0,
\'department\': \'Crew\', \'job\': \'Cinematography\', \'credit_id\': \'584a9ee
792514119bc01e99c\', \'id\': 1066767}, {\'name\': \'P.T. Walkley\', \'gender\':
0, \'department\': \'Sound\', \'job\': \'Music\', \'credit_id\': \'584a9ed2c3a3
68141a0201c7\', \'id\': 1251352}, {\'name\': \'Janet Gaynor\', \'gender\': 0,
\'department\': \'Editing\', \'job\': \'Editor\', \'credit_id\': \'584aa1ba9251
4119b301ecfa\', \'id\': 1552358}, {\'name\': \'Aaron Lubin\', \'gender\': 0,
\'department\': \'Production\', \'job\': \'Producer\', \'credit_id\': \'584a9e9
e92514119b301eb56\', \'id\': 1566006}]",Edward Burns\r\n4800,4800,0,Comedy Dram
a Romance TV Movie,http://www.hallmarkchannel.com/signedsealeddelivered,231617,
date love at first sight narration investigation team,en,"Signed, Sealed, Deliv
ered","""Signed, Sealed, Delivered"" introduces a dedicated quartet of civil se
rvants in the Dead Letter Office of the U.S. Postal System who transform themse
lves into an elite team of lost-mail detectives. Their determination to deliver
the seemingly undeliverable takes them out of the post office into an unpredict
able world where letters and packages from the past save lives, solve crimes, r
eunite old loves, and change futures by arriving late, but always miraculously
on time.",1.444476,"[{""name"": ""Front Street Pictures"", ""id"": 3958}, {""na
me"": ""Muse Entertainment Enterprises"", ""id"": 6438}]","[{""iso_3166_1"":
""US"", ""name"": ""United States of America""}]",2013-10-13,0,120.0,"[{""iso_6
39_1"": ""en"", ""name"": ""English""}]",Released,,"Signed, Sealed, Delivered",
7.0,6,Eric Mabius Kristin Booth Crystal Lowe Geoff Gustafson Benjamin Hollingsw
orth,"[{\'name\': \'Carla Hetland\', \'gender\': 0, \'department\': \'Costume &
Make-Up\', \'job\': \'Costume Design\', \'credit_id\': \'52fe4df3c3a36847f8275e
cf\', \'id\': 23816}, {\'name\': \'Harvey Kahn\', \'gender\': 2, \'department
\': \'Production\', \'job\': \'Producer\', \'credit_id\': \'583328efc3a368675d0
016e4\', \'id\': 32733}, {\'name\': \'Adam Sliwinski\', \'gender\': 2, \'depart
ment\': \'Camera\', \'job\': \'Director of Photography\', \'credit_id\': \'52fe
4df3c3a36847f8275eb7\', \'id\': 43925}, {\'name\': \'Martha Williamson\', \'gen
der\': 1, \'department\': \'Writing\', \'job\': \'Writer\', \'credit_id\': \'52
fe4df3c3a36847f8275eb1\', \'id\': 166873}, {\'name\': \'Martha Williamson\',
\'gender\': 1, \'department\': \'Production\', \'job\': \'Executive Producer\',
\'credit_id\': \'58332c6ac3a3686753001dcf\', \'id\': 166873}, {\'name\': \'Joel
S. Rice\', \'gender\': 2, \'department\': \'Production\', \'job\': \'Executive
Producer\', \'credit_id\': \'554ea9ddc3a36879530001d9\', \'id\': 174065}, {\'na
me\': \'Jackie Lind\', \'gender\': 1, \'department\': \'Production\', \'job\':
\'Casting\', \'credit_id\': \'5833288e92514103450018f0\', \'id\': 435006}, {\'n
ame\': \'Shana Landsburg\', \'gender\': 0, \'department\': \'Production\', \'jo
b\': \'Casting\', \'credit_id\': \'5833289ac3a36867500019b1\', \'id\': 551521},
{\'name\': \'Renee Read\', \'gender\': 1, \'department\': \'Art\', \'job\': \'P
roduction Design\', \'credit_id\': \'52fe4df3c3a36847f8275ec9\', \'id\': 101289
1}, {\'name\': \'Hal Beckett\', \'gender\': 2, \'department\': \'Sound\', \'job
\': \'Original Music Composer\', \'credit_id\': \'52fe4df3c3a36847f8275ec3\',

```
\'id\': 1099146}, {\'name\': \'Scott Smith\', \'gender\': 0, \'department\':
\'Directing\', \'job\': \'Director\', \'credit_id\': \'52fe4df3c3a36847f8275eab
\', \'id\': 1219158}, {\'name\': \'Scott Smith\', \'gender\': 0, \'department
\': \'Production\', \'job\': \'Executive Producer\', \'credit_id\': \'58332c3f9
251410348001bf2\', \'id\': 1219158}, {\'name\': \'Lisa Binkley\', \'gender\':
0, \'department\': \'Editing\', \'job\': \'Editor\', \'credit_id\': \'52fe4df3c
3a36847f8275ebd\', \'id\': 1271268}]",Scott Smith\r\n4801,4801,0,,http://shangh
aicalling.com/,126186,,en,Shanghai Calling,"When ambitious New York attorney Sa
m is sent to Shanghai on assignment, he immediately stumbles into a legal mess
that could end his career. With the help of a beautiful relocation specialist,
a well-connected old-timer, a clever journalist, and a street-smart legal assis
tant, Sam might just save his job, find romance, and learn to appreciate the be
auty and wonders of Shanghai. Written by Anonymous  (IMDB.com).",0.857008,[],"
[{""iso_3166_1"": ""US"", ""name"": ""United States of America""}, {""iso_3166_
1"": ""CN"", ""name"": ""China""}]",2012-05-03,0,98.0,"[{""iso_639_1"": ""en"",
""name"": ""English""}]",Released,A New Yorker in Shanghai,Shanghai Calling,5.
7,7,Daniel Henney Eliza Coupe Bill Paxton Alan Ruck Zhu Shimao,"[{\'name\': \'D
aniel Hsia\', \'gender\': 2, \'department\': \'Directing\', \'job\': \'Director
\', \'credit_id\': \'52fe4ad9c3a368484e16a36b\', \'id\': 208138}, {\'name\':
\'Daniel Hsia\', \'gender\': 2, \'department\': \'Writing\', \'job\': \'Writer
\', \'credit_id\': \'52fe4ad9c3a368484e16a371\', \'id\': 208138}]",Daniel Hsia
\r\n4802,4802,0,Documentary,,25975,obsession camcorder crush dream girl,en,My D
ate with Drew,"Ever since the second grade when he first saw her in E.T. The Ex
traterrestrial, Brian Herzlinger has had a crush on Drew Barrymore. Now, 20 yea
rs later he\'s decided to try to fulfill his lifelong dream by asking her for a
date. There\'s one small problem: She\'s Drew Barrymore and he\'s, well, Brian
Herzlinger, a broke 27-year-old aspiring filmmaker from New Jersey.",1.929883,"
[{""name"": ""rusty bear entertainment"", ""id"": 87986}, {""name"": ""lucky cr
ow films"", ""id"": 87987}]","[{""iso_3166_1"": ""US"", ""name"": ""United Stat
es of America""}]",2005-08-05,0,90.0,"[{""iso_639_1"": ""en"", ""name"": ""Engl
ish""}]",Released,,My Date with Drew,6.3,16,Drew Barrymore Brian Herzlinger Cor
ey Feldman Eric Roberts Griffin Dunne,"[{\'name\': \'Clark Peterson\', \'gender
\': 2, \'department\': \'Production\', \'job\': \'Executive Producer\', \'credi
t_id\': \'58ce021b9251415a390165d9\', \'id\': 6888}, {\'name\': \'Andrew Reimer
\', \'gender\': 2, \'department\': \'Production\', \'job\': \'Executive Produce
r\', \'credit_id\': \'58ce0232c3a36850e90157da\', \'id\': 61051}, {\'name\':
\'Brian Herzlinger\', \'gender\': 2, \'department\': \'Directing\', \'job\':
\'Director\', \'credit_id\': \'52fe44e8c3a368484e03da8d\', \'id\': 85563}, {\'n
ame\': \'Jon Gunn\', \'gender\': 2, \'department\': \'Directing\', \'job\': \'D
irector\', \'credit_id\': \'52fe44e8c3a368484e03da87\', \'id\': 94471}, {\'name
\': \'Brett Winn\', \'gender\': 0, \'department\': \'Directing\', \'job\': \'Di
rector\', \'credit_id\': \'52fe44e8c3a368484e03da97\', \'id\': 997560}]",Brian
Herzlinger\r\n'
```

In [14]:
```
ash = df1.copy()
print(ash)
```

```
      index     budget                                   genres  \
0         0  237000000  Action Adventure Fantasy Science Fiction
1         1  300000000                  Adventure Fantasy Action
2         2  245000000                    Action Adventure Crime
3         3  250000000               Action Crime Drama Thriller
4         4  260000000          Action Adventure Science Fiction
...     ...        ...                                       ...
4798   4798     220000                     Action Crime Thriller
4799   4799       9000                             Comedy Romance
4800   4800          0             Comedy Drama Romance TV Movie
4801   4801          0                                       NaN
4802   4802          0                               Documentary

                                            homepage      id  \
0                          http://www.avatarmovie.com/   19995
1          http://disney.go.com/disneypictures/pirates/     285
2           http://www.sonypictures.com/movies/spectre/  206647
3                    http://www.thedarkknightrises.com/   49026
4                  http://movies.disney.com/john-carter   49529
...                                                 ...     ...
4798                                               NaN    9367
4799                                               NaN   72766
4800  http://www.hallmarkchannel.com/signedsealeddel...  231617
4801                       http://shanghaicalling.com/  126186
4802                                               NaN   25975

                                            keywords original_language  \
0     culture clash future space war space colony so...               en
1     ocean drug abuse exotic island east india trad...               en
2            spy based on novel secret agent sequel mi6               en
3     dc comics crime fighter terrorist secret ident...               en
4     based on novel mars medallion space travel pri...               en
...                                                 ...              ...
4798  united states\u2013mexico barrier legs arms pa...               es
4799                                               NaN               en
4800  date love at first sight narration investigati...               en
4801                                               NaN               en
4802              obsession camcorder crush dream girl               en

                              original_title  \
0                                     Avatar
1     Pirates of the Caribbean: At World's End
2                                    Spectre
3                       The Dark Knight Rises
4                                John Carter
...                                      ...
4798                             El Mariachi
4799                               Newlyweds
4800                 Signed, Sealed, Delivered
4801                          Shanghai Calling
4802                        My Date with Drew

                                            overview  popularity  ...  \
0     In the 22nd century, a paraplegic Marine is di...  150.437577  ...
1     Captain Barbossa, long believed to be dead, ha...  139.082615  ...
2     A cryptic message from Bond’s past sends him o...  107.376788  ...
3     Following the death of District Attorney Harve...  112.312950  ...
4     John Carter is a war-weary, former military ca...   43.926995  ...
...                                                 ...         ...  ...
4798  El Mariachi just wants to play his guitar and ...   14.269792  ...
```

```
4799  A newlywed couple's honeymoon is upended by th...    0.642552  ...
4800  "Signed, Sealed, Delivered" introduces a dedic...    1.444476  ...
4801  When ambitious New York attorney Sam is sent t...    0.857008  ...
4802  Ever since the second grade when he first saw ...    1.929883  ...

     runtime                                   spoken_languages    status  \
0      162.0  [{"iso_639_1": "en", "name": "English"}, {"iso...  Released
1      169.0           [{"iso_639_1": "en", "name": "English"}]  Released
2      148.0  [{"iso_639_1": "fr", "name": "Fran\u00e7ais"},...  Released
3      165.0           [{"iso_639_1": "en", "name": "English"}]  Released
4      132.0           [{"iso_639_1": "en", "name": "English"}]  Released
...      ...                                                ...       ...
4798    81.0      [{"iso_639_1": "es", "name": "Espa\u00f1ol"}]  Released
4799    85.0                                                []  Released
4800   120.0           [{"iso_639_1": "en", "name": "English"}]  Released
4801    98.0           [{"iso_639_1": "en", "name": "English"}]  Released
4802    90.0           [{"iso_639_1": "en", "name": "English"}]  Released

                                            tagline  \
0                          Enter the World of Pandora.
1       At the end of the world, the adventure begins.
2                                A Plan No One Escapes
3                                      The Legend Ends
4                 Lost in our world, found in another.
...                                                ...
4798  He didn't come looking for trouble, but troubl...
4799  A newlywed couple's honeymoon is upended by th...
4800                                              NaN
4801                          A New Yorker in Shanghai
4802                                              NaN

                                     title vote_average vote_count  \
0                                   Avatar          7.2      11800
1     Pirates of the Caribbean: At World's End          6.9       4500
2                                  Spectre          6.3       4466
3                      The Dark Knight Rises          7.6       9106
4                                 John Carter          6.1       2124
...                                       ...          ...        ...
4798                             El Mariachi          6.6        238
4799                               Newlyweds          5.9          5
4800                 Signed, Sealed, Delivered          7.0          6
4801                          Shanghai Calling          5.7          7
4802                         My Date with Drew          6.3         16

                                                cast  \
0     Sam Worthington Zoe Saldana Sigourney Weaver S...
1     Johnny Depp Orlando Bloom Keira Knightley Stel...
2     Daniel Craig Christoph Waltz L\u00e9a Seydoux ...
3     Christian Bale Michael Caine Gary Oldman Anne ...
4     Taylor Kitsch Lynn Collins Samantha Morton Wil...
...                                                ...
4798  Carlos Gallardo Jaime de Hoyos Peter Marquardt...
4799  Edward Burns Kerry Bish\u00e9 Marsha Dietlein ...
4800  Eric Mabius Kristin Booth Crystal Lowe Geoff G...
4801  Daniel Henney Eliza Coupe Bill Paxton Alan Ruc...
4802  Drew Barrymore Brian Herzlinger Corey Feldman ...

                                                crew           director
0     [{'name': 'Stephen E. Rivkin', 'gender': 0, 'd...      James Cameron
1     [{'name': 'Dariusz Wolski', 'gender': 2, 'depa...     Gore Verbinski
```

```
2     [{'name': 'Thomas Newman', 'gender': 2, 'depar...         Sam Mendes
3     [{'name': 'Hans Zimmer', 'gender': 2, 'departm...  Christopher Nolan
4     [{'name': 'Andrew Stanton', 'gender': 2, 'depa...     Andrew Stanton
...                                                 ...                ...
4798  [{'name': 'Robert Rodriguez', 'gender': 0, 'de...   Robert Rodriguez
4799  [{'name': 'Edward Burns', 'gender': 2, 'depart...       Edward Burns
4800  [{'name': 'Carla Hetland', 'gender': 0, 'depar...        Scott Smith
4801  [{'name': 'Daniel Hsia', 'gender': 2, 'departm...        Daniel Hsia
4802  [{'name': 'Clark Peterson', 'gender': 2, 'depa...   Brian Herzlinger

[4803 rows x 24 columns]
```

In [16]:
```
df1.count()
```

Out[16]:
```
index                  4803
budget                 4803
genres                 4775
homepage               1712
id                     4803
keywords               4391
original_language      4803
original_title         4803
overview               4800
popularity             4803
production_companies   4803
production_countries   4803
release_date           4802
revenue                4803
runtime                4801
spoken_languages       4803
status                 4803
tagline                3959
title                  4803
vote_average           4803
vote_count             4803
cast                   4760
crew                   4803
director               4773
dtype: int64
```

In [19]:
```
df1.dropna()
```

Out[19]:

	index	budget	genres	homepage	id
0	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	19995
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	285
2	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	206647
3	3	250000000	Action Crime Drama Thriller	http://www.thedarkknightrises.com/	49026
4	4	260000000	Action Adventure Science Fiction	http://movies.disney.com/john-carter	49529
...	...	...	...	...	...
4772	4772	31192	Drama Action Comedy	http://downterrace.blogspot.com/	42151
4773	4773	27000	Comedy	http://www.miramax.com/movie/clerks/	2292
4781	4781	22000	Comedy Romance	https://www.facebook.com/DrySpellMovie	255266
4791	4791	13	Horror	http://tincanmanthemovie.com/	157185

	index	budget	genres	homepage	id
4796	4796	7000	Science Fiction Drama Thriller	http://www.primermovie.com	14337

1432 rows × 24 columns

In [20]:
```
df1.any()
```

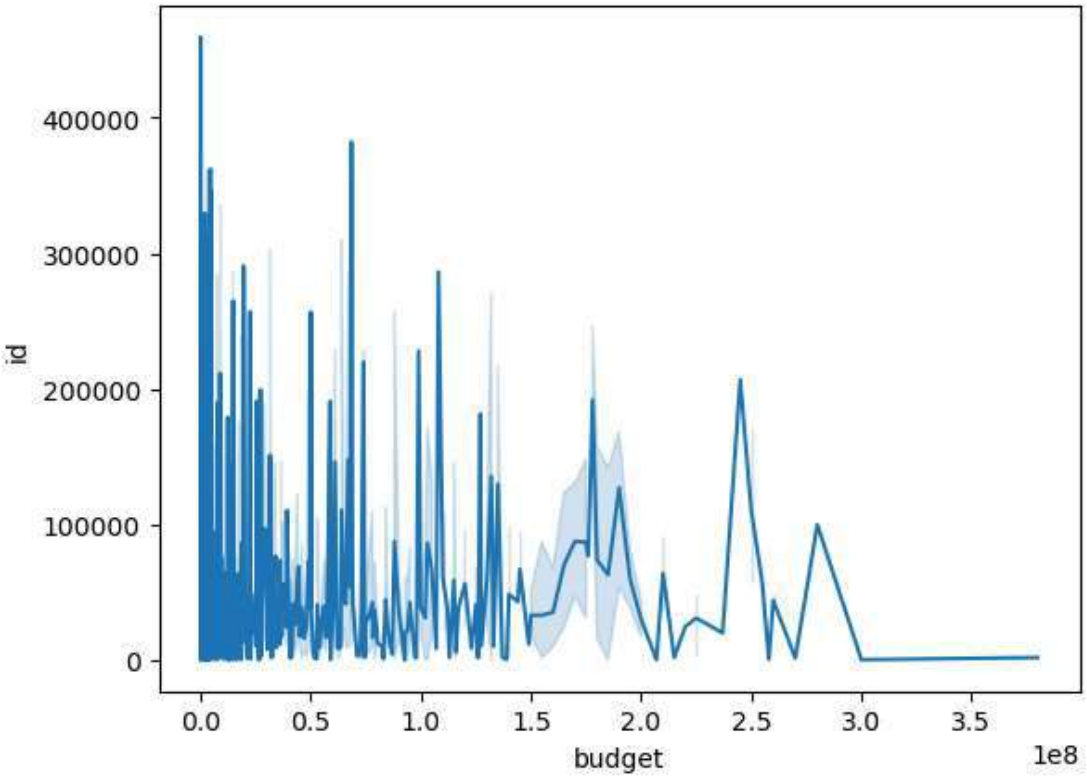
Out[20]:
```
index                   True
budget                  True
genres                  True
homepage                True
id                      True
keywords                True
original_language       True
original_title          True
overview                True
popularity              True
production_companies    True
production_countries    True
release_date            True
revenue                 True
runtime                 True
spoken_languages        True
status                  True
tagline                 True
title                   True
vote_average            True
vote_count              True
cast                    True
crew                    True
director                True
dtype: bool
```

In [21]:
```
mr = df1.get(40)
print(mr)
```

```
None
```

In [27]:
```
import seaborn as sea
sea.lineplot(x="budget", y="id", data=df1)
```

Out[27]:
```
<Axes: xlabel='budget', ylabel='id'>
```


400000
300000
200000
100000
0
id
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
budget
1e8

In [30]:
```
df1.max
```

Out[30]:
```
<bound method NDFrame._add_numeric_operations.<locals>.max of       index     b
udget                                     genres  \
0         0  237000000  Action Adventure Fantasy Science Fiction
1         1  300000000                 Adventure Fantasy Action
2         2  245000000                   Action Adventure Crime
3         3  250000000               Action Crime Drama Thriller
4         4  260000000          Action Adventure Science Fiction
...     ...        ...                                      ...
4798   4798     220000                     Action Crime Thriller
4799   4799       9000                             Comedy Romance
4800   4800          0             Comedy Drama Romance TV Movie
4801   4801          0                                       NaN
4802   4802          0                               Documentary

                                            homepage      id  \
0                          http://www.avatarmovie.com/   19995
1          http://disney.go.com/disneypictures/pirates/     285
2           http://www.sonypictures.com/movies/spectre/  206647
3                   http://www.thedarkknightrises.com/   49026
4                 http://movies.disney.com/john-carter   49529
...                                                ...     ...
4798                                               NaN    9367
4799                                               NaN   72766
4800  http://www.hallmarkchannel.com/signedsealeddel...  231617
4801                       http://shanghaicalling.com/  126186
4802                                               NaN   25975

                                            keywords original_language  \
0     culture clash future space war space colony so...               en
1     ocean drug abuse exotic island east india trad...               en
2            spy based on novel secret agent sequel mi6               en
3     dc comics crime fighter terrorist secret ident...               en
4     based on novel mars medallion space travel pri...               en
...                                                ...              ...
4798  united states\u2013mexico barrier legs arms pa...               es
4799                                               NaN               en
4800  date love at first sight narration investigati...               en
4801                                               NaN               en
4802              obsession camcorder crush dream girl               en

                               original_title  \
0                                      Avatar
1     Pirates of the Caribbean: At World's End
2                                     Spectre
3                        The Dark Knight Rises
4                                  John Carter
...                                        ...
4798                               El Mariachi
4799                                 Newlyweds
4800                 Signed, Sealed, Delivered
4801                          Shanghai Calling
4802                        My Date with Drew

                                            overview  popularity  ...  \
0     In the 22nd century, a paraplegic Marine is di...  150.437577  ...
1     Captain Barbossa, long believed to be dead, ha...  139.082615  ...
2     A cryptic message from Bond’s past sends him o...  107.376788  ...
3     Following the death of District Attorney Harve...  112.312950  ...
4     John Carter is a war-weary, former military ca...   43.926995  ...
...                                                ...         ...  ...
```

```
4798  El Mariachi just wants to play his guitar and ...   14.269792  ...
4799  A newlywed couple's honeymoon is upended by th...    0.642552  ...
4800  "Signed, Sealed, Delivered" introduces a dedic...    1.444476  ...
4801  When ambitious New York attorney Sam is sent t...    0.857008  ...
4802  Ever since the second grade when he first saw ...    1.929883  ...

     runtime                                   spoken_languages    status  \
0      162.0  [{"iso_639_1": "en", "name": "English"}, {"iso...  Released
1      169.0           [{"iso_639_1": "en", "name": "English"}]  Released
2      148.0  [{"iso_639_1": "fr", "name": "Fran\u00e7ais"},...  Released
3      165.0           [{"iso_639_1": "en", "name": "English"}]  Released
4      132.0           [{"iso_639_1": "en", "name": "English"}]  Released
...      ...                                                ...       ...
4798    81.0      [{"iso_639_1": "es", "name": "Espa\u00f1ol"}]  Released
4799    85.0                                                []  Released
4800   120.0           [{"iso_639_1": "en", "name": "English"}]  Released
4801    98.0           [{"iso_639_1": "en", "name": "English"}]  Released
4802    90.0           [{"iso_639_1": "en", "name": "English"}]  Released

                                              tagline  \
0                          Enter the World of Pandora.
1        At the end of the world, the adventure begins.
2                                A Plan No One Escapes
3                                      The Legend Ends
4                  Lost in our world, found in another.
...                                                ...
4798  He didn't come looking for trouble, but troubl...
4799  A newlywed couple's honeymoon is upended by th...
4800                                              NaN
4801                           A New Yorker in Shanghai
4802                                              NaN

                                       title vote_average vote_count  \
0                                     Avatar          7.2      11800
1     Pirates of the Caribbean: At World's End          6.9       4500
2                                    Spectre          6.3       4466
3                       The Dark Knight Rises          7.6       9106
4                                  John Carter          6.1       2124
...                                        ...          ...        ...
4798                               El Mariachi          6.6        238
4799                                 Newlyweds          5.9          5
4800                  Signed, Sealed, Delivered          7.0          6
4801                           Shanghai Calling          5.7          7
4802                          My Date with Drew          6.3         16

                                                 cast  \
0     Sam Worthington Zoe Saldana Sigourney Weaver S...
1     Johnny Depp Orlando Bloom Keira Knightley Stel...
2     Daniel Craig Christoph Waltz L\u00e9a Seydoux ...
3     Christian Bale Michael Caine Gary Oldman Anne ...
4     Taylor Kitsch Lynn Collins Samantha Morton Wil...
...                                                ...
4798  Carlos Gallardo Jaime de Hoyos Peter Marquardt...
4799  Edward Burns Kerry Bish\u00e9 Marsha Dietlein ...
4800  Eric Mabius Kristin Booth Crystal Lowe Geoff G...
4801  Daniel Henney Eliza Coupe Bill Paxton Alan Ruc...
4802  Drew Barrymore Brian Herzlinger Corey Feldman ...

                                                 crew           director
0     [{'name': 'Stephen E. Rivkin', 'gender': 0, 'd...      James Cameron
```

```
1     [{'name': 'Dariusz Wolski', 'gender': 2, 'depa...     Gore Verbinski
2     [{'name': 'Thomas Newman', 'gender': 2, 'depar...         Sam Mendes
3     [{'name': 'Hans Zimmer', 'gender': 2, 'departm...  Christopher Nolan
4     [{'name': 'Andrew Stanton', 'gender': 2, 'depa...     Andrew Stanton
...                                                 ...                ...
4798  [{'name': 'Robert Rodriguez', 'gender': 0, 'de...   Robert Rodriguez
4799  [{'name': 'Edward Burns', 'gender': 2, 'depart...       Edward Burns
4800  [{'name': 'Carla Hetland', 'gender': 0, 'depar...        Scott Smith
4801  [{'name': 'Daniel Hsia', 'gender': 2, 'departm...        Daniel Hsia
4802  [{'name': 'Clark Peterson', 'gender': 2, 'depa...   Brian Herzlinger

[4803 rows x 24 columns]>
```

In [31]:
```
df1.min
```

Out[31]:
```
<bound method NDFrame._add_numeric_operations.<locals>.min of       index     b
udget                                     genres  \
0         0  237000000  Action Adventure Fantasy Science Fiction
1         1  300000000                 Adventure Fantasy Action
2         2  245000000                   Action Adventure Crime
3         3  250000000               Action Crime Drama Thriller
4         4  260000000          Action Adventure Science Fiction
...     ...        ...                                      ...
4798   4798     220000                     Action Crime Thriller
4799   4799       9000                             Comedy Romance
4800   4800          0             Comedy Drama Romance TV Movie
4801   4801          0                                       NaN
4802   4802          0                               Documentary

                                            homepage      id  \
0                          http://www.avatarmovie.com/   19995
1          http://disney.go.com/disneypictures/pirates/     285
2           http://www.sonypictures.com/movies/spectre/  206647
3                   http://www.thedarkknightrises.com/   49026
4                 http://movies.disney.com/john-carter   49529
...                                                ...     ...
4798                                               NaN    9367
4799                                               NaN   72766
4800  http://www.hallmarkchannel.com/signedsealeddel...  231617
4801                       http://shanghaicalling.com/  126186
4802                                               NaN   25975

                                            keywords original_language  \
0     culture clash future space war space colony so...               en
1     ocean drug abuse exotic island east india trad...               en
2            spy based on novel secret agent sequel mi6               en
3     dc comics crime fighter terrorist secret ident...               en
4     based on novel mars medallion space travel pri...               en
...                                                ...              ...
4798  united states\u2013mexico barrier legs arms pa...               es
4799                                               NaN               en
4800  date love at first sight narration investigati...               en
4801                                               NaN               en
4802              obsession camcorder crush dream girl               en

                                original_title  \
0                                       Avatar
1     Pirates of the Caribbean: At World's End
2                                      Spectre
3                        The Dark Knight Rises
4                                   John Carter
...                                         ...
4798                                El Mariachi
4799                                  Newlyweds
4800                  Signed, Sealed, Delivered
4801                           Shanghai Calling
4802                         My Date with Drew

                                            overview  popularity  ...  \
0     In the 22nd century, a paraplegic Marine is di...  150.437577  ...
1     Captain Barbossa, long believed to be dead, ha...  139.082615  ...
2     A cryptic message from Bond’s past sends him o...  107.376788  ...
3     Following the death of District Attorney Harve...  112.312950  ...
4     John Carter is a war-weary, former military ca...   43.926995  ...
...                                                ...         ...  ...
```

```
4798  El Mariachi just wants to play his guitar and ...   14.269792  ...
4799  A newlywed couple's honeymoon is upended by th...    0.642552  ...
4800  "Signed, Sealed, Delivered" introduces a dedic...    1.444476  ...
4801  When ambitious New York attorney Sam is sent t...    0.857008  ...
4802  Ever since the second grade when he first saw ...    1.929883  ...

     runtime                                   spoken_languages    status  \
0      162.0  [{"iso_639_1": "en", "name": "English"}, {"iso...  Released
1      169.0           [{"iso_639_1": "en", "name": "English"}]  Released
2      148.0  [{"iso_639_1": "fr", "name": "Fran\u00e7ais"},...  Released
3      165.0           [{"iso_639_1": "en", "name": "English"}]  Released
4      132.0           [{"iso_639_1": "en", "name": "English"}]  Released
...      ...                                                ...       ...
4798    81.0      [{"iso_639_1": "es", "name": "Espa\u00f1ol"}]  Released
4799    85.0                                                []  Released
4800   120.0           [{"iso_639_1": "en", "name": "English"}]  Released
4801    98.0           [{"iso_639_1": "en", "name": "English"}]  Released
4802    90.0           [{"iso_639_1": "en", "name": "English"}]  Released

                                              tagline  \
0                          Enter the World of Pandora.
1        At the end of the world, the adventure begins.
2                                A Plan No One Escapes
3                                      The Legend Ends
4                  Lost in our world, found in another.
...                                                ...
4798  He didn't come looking for trouble, but troubl...
4799  A newlywed couple's honeymoon is upended by th...
4800                                              NaN
4801                           A New Yorker in Shanghai
4802                                              NaN

                                       title vote_average vote_count  \
0                                     Avatar          7.2      11800
1     Pirates of the Caribbean: At World's End          6.9       4500
2                                    Spectre          6.3       4466
3                       The Dark Knight Rises          7.6       9106
4                                  John Carter          6.1       2124
...                                        ...          ...        ...
4798                               El Mariachi          6.6        238
4799                                 Newlyweds          5.9          5
4800                  Signed, Sealed, Delivered          7.0          6
4801                           Shanghai Calling          5.7          7
4802                          My Date with Drew          6.3         16

                                                 cast  \
0     Sam Worthington Zoe Saldana Sigourney Weaver S...
1     Johnny Depp Orlando Bloom Keira Knightley Stel...
2     Daniel Craig Christoph Waltz L\u00e9a Seydoux ...
3     Christian Bale Michael Caine Gary Oldman Anne ...
4     Taylor Kitsch Lynn Collins Samantha Morton Wil...
...                                                ...
4798  Carlos Gallardo Jaime de Hoyos Peter Marquardt...
4799  Edward Burns Kerry Bish\u00e9 Marsha Dietlein ...
4800  Eric Mabius Kristin Booth Crystal Lowe Geoff G...
4801  Daniel Henney Eliza Coupe Bill Paxton Alan Ruc...
4802  Drew Barrymore Brian Herzlinger Corey Feldman ...

                                                 crew           director
0     [{'name': 'Stephen E. Rivkin', 'gender': 0, 'd...      James Cameron
```

```
1     [{'name': 'Dariusz Wolski', 'gender': 2, 'depa...     Gore Verbinski
2     [{'name': 'Thomas Newman', 'gender': 2, 'depar...         Sam Mendes
3     [{'name': 'Hans Zimmer', 'gender': 2, 'departm...  Christopher Nolan
4     [{'name': 'Andrew Stanton', 'gender': 2, 'depa...     Andrew Stanton
...                                                 ...                ...
4798  [{'name': 'Robert Rodriguez', 'gender': 0, 'de...   Robert Rodriguez
4799  [{'name': 'Edward Burns', 'gender': 2, 'depart...       Edward Burns
4800  [{'name': 'Carla Hetland', 'gender': 0, 'depar...        Scott Smith
4801  [{'name': 'Daniel Hsia', 'gender': 2, 'departm...        Daniel Hsia
4802  [{'name': 'Clark Peterson', 'gender': 2, 'depa...   Brian Herzlinger

[4803 rows x 24 columns]>
```

In [32]:
```
df1.mode()
```

Out[32]:

	index	budget	genres	homepage	id	keywords	o
0	0	0.0	Drama	http://www.missionimpossible.com/	5	independent film	
1	1	NaN	NaN	http://www.thehungergames.movie/	11	NaN	
2	2	NaN	NaN	NaN	12	NaN	
3	3	NaN	NaN	NaN	13	NaN	
4	4	NaN	NaN	NaN	14	NaN	
...	...	...	...	...	...	...	
4798	4798	NaN	NaN	NaN	426067	NaN	
4799	4799	NaN	NaN	NaN	426469	NaN	
4800	4800	NaN	NaN	NaN	433715	NaN	
4801	4801	NaN	NaN	NaN	447027	NaN	
4802	4802	NaN	NaN	NaN	459488	NaN	

4803 rows × 24 columns

In [33]:
```
df1.sort_index(axis=1, ascending=False)
```

Out[33]:

	vote_count	vote_average	title	tagline	status	spoken_languages	r
0	11800	7.2	Avatar	Enter the World of Pandora.	Released	[{"iso_639_1": "en", "name": "English"}, {"iso...	
1	4500	6.9	Pirates of the Caribbean: At World's End	At the end of the world, the adventure begins.	Released	[{"iso_639_1": "en", "name": "English"}]	
2	4466	6.3	Spectre	A Plan No One Escapes	Released	[{"iso_639_1": "fr", "name": "Fran\u00e7ais"},...	
3	9106	7.6	The Dark Knight Rises	The Legend Ends	Released	[{"iso_639_1": "en", "name": "English"}]	
4	2124	6.1	John Carter	Lost in our world, found in another.	Released	[{"iso_639_1": "en", "name": "English"}]	
...	...	...	...	...	...	...	
4798	238	6.6	El Mariachi	He didn't come looking for trouble, but troubl...	Released	[{"iso_639_1": "es", "name": "Espa\u00f1ol"}]	
4799	5	5.9	Newlyweds	A newlywed couple's honeymoon is upended by th...	Released	[]	
4800	6	7.0	Signed, Sealed, Delivered	NaN	Released	[{"iso_639_1": "en", "name": "English"}]	
4801	7	5.7	Shanghai Calling	A New Yorker in Shanghai	Released	[{"iso_639_1": "en", "name": "English"}]	
4802	16	6.3	My Date with Drew	NaN	Released	[{"iso_639_1": "en", "name": "English"}]	

vote_count	vote_average	title	tagline	status	spoken_languages	r

4803 rows × 24 columns

In [35]:
```
df1.sort_values(by="budget")
```

Out[35]:

	index	budget	genres	homepage	
2401	2401	0	Thriller	NaN	319
3187	3187	0	Comedy Music	NaN	126
3183	3183	0	Romance Drama Family	NaN	146
3179	3179	0	Comedy Romance	NaN	246
3178	3178	0	Drama	http://www.foxsearchlight.com/thesavages	82
...	...	...	...	...	
4	4	260000000	Action Adventure Science Fiction	http://movies.disney.com/john-carter	495
10	10	270000000	Adventure Fantasy Action Science Fiction	http://www.superman.com	14
7	7	280000000	Action Adventure Science Fiction	http://marvel.com/movies/movie/193/avengers_ag...	998

	index	budget	genres	homepage	
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	2
17	17	380000000	Adventure Action Fantasy	http://disney.go.com/pirates/index-on-stranger...	18

4803 rows × 24 columns

In [36]:
```
df1.iloc[5]
```

Out[36]:
```
index                                                                5
budget                                                       258000000
genres                                         Fantasy Action Adventure
homepage                 http://www.sonypictures.com/movies/spider-man3/
id                                                                 559
keywords               dual identity amnesia sandstorm love of one's ...
original_language                                                   en
original_title                                             Spider-Man 3
overview               The seemingly invincible Spider-Man goes up ag...
popularity                                                  115.699814
production_companies   [{"name": "Columbia Pictures", "id": 5}, {"nam...
production_countries   [{"iso_3166_1": "US", "name": "United States o...
release_date                                                2007-05-01
revenue                                                      890871626
runtime                                                          139.0
spoken_languages       [{"iso_639_1": "en", "name": "English"}, {"iso...
status                                                        Released
tagline                                                The battle within.
title                                                      Spider-Man 3
vote_average                                                        5.9
vote_count                                                         3576
cast                   Tobey Maguire Kirsten Dunst James Franco Thoma...
crew                   [{'name': 'Francine Maisler', 'gender': 1, 'de...
director                                                     Sam Raimi
Name: 5, dtype: object
```

In [37]:
```
df1[0:3]
```

Out[37]:

	index	budget	genres	homepage	id	key
0	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	19995	spa
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	285	eas
2	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	206647	spy or

3 rows × 24 columns

In [40]:
```
df1.loc[:, ["budget","id"]]
```

Out[40]:

	budget	id
0	237000000	19995
1	300000000	285
2	245000000	206647
3	250000000	49026
4	260000000	49529
...	...	...
4798	220000	9367
4799	9000	72766
4800	0	231617
4801	0	126186
4802	0	25975

4803 rows × 2 columns

In [41]:
```
df1.iloc[:30, :]
```

Out[41]:

	index	budget	genres	homepage	
0	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	19
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	
2	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	206
3	3	250000000	Action Crime Drama Thriller	http://www.thedarkknightrises.com/	49
4	4	260000000	Action Adventure Science Fiction	http://movies.disney.com/john-carter	49
5	5	258000000	Fantasy Action Adventure	http://www.sonypictures.com/movies/spider-man3/	
6	6	260000000	Animation Family	http://disney.go.com/disneypictures/tangled/	38
7	7	280000000	Action Adventure Science Fiction	http://marvel.com/movies/movie/193/avengers_ag...	99
8	8	250000000	Adventure Fantasy Family	http://harrypotter.warnerbros.com/harrypottera...	

	index	budget	genres	homepage	
9	9	250000000	Action Adventure Fantasy	http://www.batmanvsupermandawnofjustice.com/	209
10	10	270000000	Adventure Fantasy Action Science Fiction	http://www.superman.com	1
11	11	200000000	Adventure Action Thriller Crime	http://www.mgm.com/view/movie/234/Quantum-of-S...	10
12	12	200000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	
13	13	255000000	Action Adventure Western	http://disney.go.com/the-lone-ranger/	57
14	14	225000000	Action Adventure Fantasy Science Fiction	http://www.manofsteel.com/	49
15	15	225000000	Adventure Family Fantasy	NaN	2
16	16	220000000	Science Fiction Action Adventure	http://marvel.com/avengers_movie/	24
17	17	380000000	Adventure Action Fantasy	http://disney.go.com/pirates/index-on-stranger...	1

	index	budget	genres	homepage	
18	18	225000000	Action Comedy Science Fiction	http://www.sonypictures.com/movies/meninblack3/	41
19	19	250000000	Action Adventure Fantasy	http://www.thehobbit.com/	122
20	20	215000000	Action Adventure Fantasy	http://www.theamazingspiderman.com	1
21	21	200000000	Action Adventure	http://www.robinhoodthemovie.com/	20
22	22	250000000	Adventure Fantasy	http://www.thehobbit.com/	57
23	23	180000000	Adventure Fantasy	http://www.goldencompassmovie.com/index_german...	2
24	24	207000000	Adventure Drama Action	NaN	
25	25	200000000	Drama Romance Thriller	http://www.titanicmovie.com	
26	26	250000000	Adventure Action Science Fiction	http://marvel.com/captainamericapremiere	271

	index	budget	genres	homepage	
27	27	209000000	Thriller Action Adventure Science Fiction	NaN	44
28	28	150000000	Action Adventure Science Fiction Thriller	http://www.jurassicworld.com/	135
29	29	200000000	Action Adventure Thriller	http://www.skyfall-movie.com	37

30 rows × 24 columns

In [42]:
```
df1.iloc[:, :17]
```

Out[42]:

	index	budget	genres	homepage	
0	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	
2	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	2
3	3	250000000	Action Crime Drama Thriller	http://www.thedarkknightrises.com/	
4	4	260000000	Action Adventure Science Fiction	http://movies.disney.com/john-carter	
...	...	...	...	...	
4798	4798	220000	Action Crime Thriller	NaN	
4799	4799	9000	Comedy Romance	NaN	
4800	4800	0	Comedy Drama Romance TV Movie	http://www.hallmarkchannel.com/signedsealeddel...	2
4801	4801	0	NaN	http://shanghaicalling.com/	1

	index	budget	genres	homepage
4802	4802	0	Documentary	NaN

4803 rows × 17 columns

In [43]:
```
df1.iloc[:6, :12]
```

Out[43]:

	index	budget	genres	homepage	id	key
0	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	19995	spa
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	285	isla
2	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	206647	spy o
3	3	250000000	Action Crime Drama Thriller	http://www.thedarkknightrises.com/	49026	dc t
4	4	260000000	Action Adventure Science Fiction	http://movies.disney.com/john-carter	49529	ba me trav
5	5	258000000	Fantasy Action Adventure	http://www.sonypictures.com/movies/spider-man3/	559	i a san

In [45]:
```
df1.iloc[3:5, 0:2]
```

Out[45]:

	index	budget
3	3	250000000
4	4	260000000

In [46]:
```
df1.iloc[[1, 2,4], [0, 2]]
```

Out[46]:

	index	genres
1	1	Adventure Fantasy Action
2	2	Action Adventure Crime
4	4	Action Adventure Science Fiction

In [47]:
```
df1.iloc[1:3, :]
```

Out[47]:

	index	budget	genres	homepage	id	key
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	285	eas
2	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	206647	spy or

2 rows × 24 columns

In [48]:
```
df1[df1.columns[2:4]].iloc[5:10]
```

Out[48]:

	genres	homepage
5	Fantasy Action Adventure	http://www.sonypictures.com/movies/spider-man3/
6	Animation Family	http://disney.go.com/disneypictures/tangled/
7	Action Adventure Science Fiction	http://marvel.com/movies/movie/193/avengers_ag...
8	Adventure Fantasy Family	http://harrypotter.warnerbros.com/harrypottera...
9	Action Adventure Fantasy	http://www.batmanvsupermandawnofjustice.com/

In [49]:
```
df1.isnull()
```

Out[49]:

	index	budget	genres	homepage	id	keywords	original_language	original_t
0	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	
4798	False	False	False	True	False	False	False	F
4799	False	False	False	True	False	True	False	F
4800	False	False	False	False	False	False	False	F
4801	False	False	True	False	False	True	False	F
4802	False	False	False	True	False	False	False	F

4803 rows × 24 columns

In [50]:
```
df1.isnull().any()
```

Out[50]:
```
index                  False
budget                 False
genres                  True
homepage                True
id                     False
keywords                True
original_language      False
original_title         False
overview                True
popularity             False
production_companies   False
production_countries   False
release_date            True
revenue                False
runtime                 True
spoken_languages       False
status                 False
tagline                 True
title                  False
vote_average           False
vote_count             False
cast                    True
crew                   False
director                True
dtype: bool
```

In [51]:
```
df1.isnull().sum().sum()
```

Out[51]: 4454

In [52]:
```
df1.isnull().sum()
```

Out[52]:
```
index                     0
budget                    0
genres                   28
homepage               3091
id                        0
keywords                412
original_language         0
original_title            0
overview                  3
popularity                0
production_companies      0
production_countries      0
release_date              1
revenue                   0
runtime                   2
spoken_languages          0
status                    0
tagline                 844
title                     0
vote_average              0
vote_count                0
cast                     43
crew                      0
director                 30
dtype: int64
```

In [53]:
```
df1.isnull().sum(axis=1)
```

Out[53]:
```
0       0
1       0
2       0
3       0
4       0
       ..
4798    1
4799    2
4800    1
4801    2
4802    2
Length: 4803, dtype: int64
```

In [54]:
```
df1.isna().sum()
```

Out[54]:
```
index                     0
budget                    0
genres                   28
homepage               3091
id                        0
keywords                412
original_language         0
original_title            0
overview                  3
popularity                0
production_companies      0
production_countries      0
release_date              1
revenue                   0
runtime                   2
spoken_languages          0
status                    0
tagline                 844
title                     0
vote_average              0
vote_count                0
cast                     43
crew                      0
director                 30
dtype: int64
```

In [55]:
```
df1.groupby(['budget'])['id'].apply(lambda x:x.isnull().sum())
```

Out[55]:
```
budget
0            0
1            0
2            0
3            0
4            0
            ..
260000000    0
270000000    0
280000000    0
300000000    0
380000000    0
Name: id, Length: 436, dtype: int64
```

In [56]:
```
df1.dtypes
```

Out[56]:
```
index                     int64
budget                    int64
genres                   object
homepage                 object
id                        int64
keywords                 object
original_language        object
original_title           object
overview                 object
popularity              float64
production_companies     object
production_countries     object
release_date             object
revenue                   int64
runtime                 float64
spoken_languages         object
status                   object
tagline                  object
title                    object
vote_average            float64
vote_count                int64
cast                     object
crew                     object
director                 object
dtype: object
```

In [57]:
```
df1['budget']= df1['budget'].astype("int")
df1['budget']
```

Out[57]:
```
0       237000000
1       300000000
2       245000000
3       250000000
4       260000000
          ...
4798       220000
4799         9000
4800            0
4801            0
4802            0
Name: budget, Length: 4803, dtype: int32
```

In [66]:
```
df1['genres'].unique()
```

Out[66]:
```
array(['Action Adventure Fantasy Science Fiction',
       'Adventure Fantasy Action', 'Action Adventure Crime', ...,
       'Thriller Horror Comedy', 'Foreign Thriller',
       'Comedy Drama Romance TV Movie'], dtype=object)
```

In [67]:
```
label_encoder = preprocessing.LabelEncoder()
```

In [70]:
```
df1['genres']= label_encoder.fit_transform(df1['genres'])
```

In [71]:
```
df1['genres'].unique()
```

Out[71]:
```
array([  59,  327,   29, ..., 1122,  878,  477])
```

In [8]:
```
from sklearn import preprocessing
features_df=df1.drop(columns=['genres'])
```

```
enc = preprocessing.OneHotEncoder()
enc_df=pd.DataFrame(enc.fit_transform(df1[['genres']]))
df_encode = features_df.join(enc_df)
df_encode
```

Out[8]:

	index	budget	homepage	id	
0	0	237000000	http://www.avatarmovie.com/	19995	culture spa
1	1	300000000	http://disney.go.com/disneypictures/pirates/	285	ocea exot
2	2	245000000	http://www.sonypictures.com/movies/spectre/	206647	spy ba secret
3	3	250000000	http://www.thedarkknightrises.com/	49026	dc fig
4	4	260000000	http://movies.disney.com/john-carter	49529	ba ma spa
...	...	...	...	...	
4798	4798	220000	NaN	9367	states\ barr
4799	4799	9000	NaN	72766	
4800	4800	0	http://www.hallmarkchannel.com/signedsealeddel...	231617	dat si
4801	4801	0	http://shanghaicalling.com/	126186	

	index	budget	homepage	id	
4802	4802	0	NaN	25975	cam

4803 rows × 24 columns

In [11]:
```
one_hot_df = pd.get_dummies(df1, prefix="genres",columns=['genres'], drop_first=
one_hot_df
```

Out[11]:

	index	budget	homepage	id	
0	0	237000000	http://www.avatarmovie.com/	19995	culture spa
1	1	300000000	http://disney.go.com/disneypictures/pirates/	285	ocea exot
2	2	245000000	http://www.sonypictures.com/movies/spectre/	206647	spy ba secret
3	3	250000000	http://www.thedarkknightrises.com/	49026	dc fig
4	4	260000000	http://movies.disney.com/john-carter	49529	ba ma spa
...	...	...	...	...	
4798	4798	220000	NaN	9367	states\ barr
4799	4799	9000	NaN	72766	
4800	4800	0	http://www.hallmarkchannel.com/signedsealeddel...	231617	dat si
4801	4801	0	http://shanghaicalling.com/	126186	

	index	budget	homepage	id	
4802	4802	0	NaN	25975	cam

4803 rows × 1190 columns

In [29]:
```
new_data = df1.dropna(axis = 0, how ='any')
new_data
```

Out[29]:

	index	budget	m score	budget	genres	
0	0	80000000.0	80000000.0	237000000	Action Adventure Fantasy Science Fiction	http://www.av
1	1	80000000.0	80000000.0	300000000	Adventure Fantasy Action	http://disney.go.com/disney
2	2	80000000.0	80000000.0	245000000	Action Adventure Crime	http://www.sonypictures.com/
3	3	80000000.0	80000000.0	250000000	Action Crime Drama Thriller	http://www.thedark
4	4	80000000.0	80000000.0	260000000	Action Adventure Science Fiction	http://movies.disney.
...	...	...	...	...	...	
4772	4772	31192.0	31192.0	31192	Drama Action Comedy	http://downterrac
4773	4773	27000.0	27000.0	27000	Comedy	http://www.miramax.co
4781	4781	22000.0	22000.0	22000	Comedy Romance	https://www.facebook.com
4791	4791	13.0	13.0	13	Horror	http://tincanma

	index	budget	m score	budget	genres	
4796	4796	7000.0	7000.0	7000	Science Fiction Drama Thriller	http://www.p

1432 rows × 26 columns

In [30]:
```
import numpy as np
import matplotlib.pyplot as plt
print(np.where(df1['index']>90))
print(np.where(df1['budget']<25))
print(np.where(df1['id']<30))
```

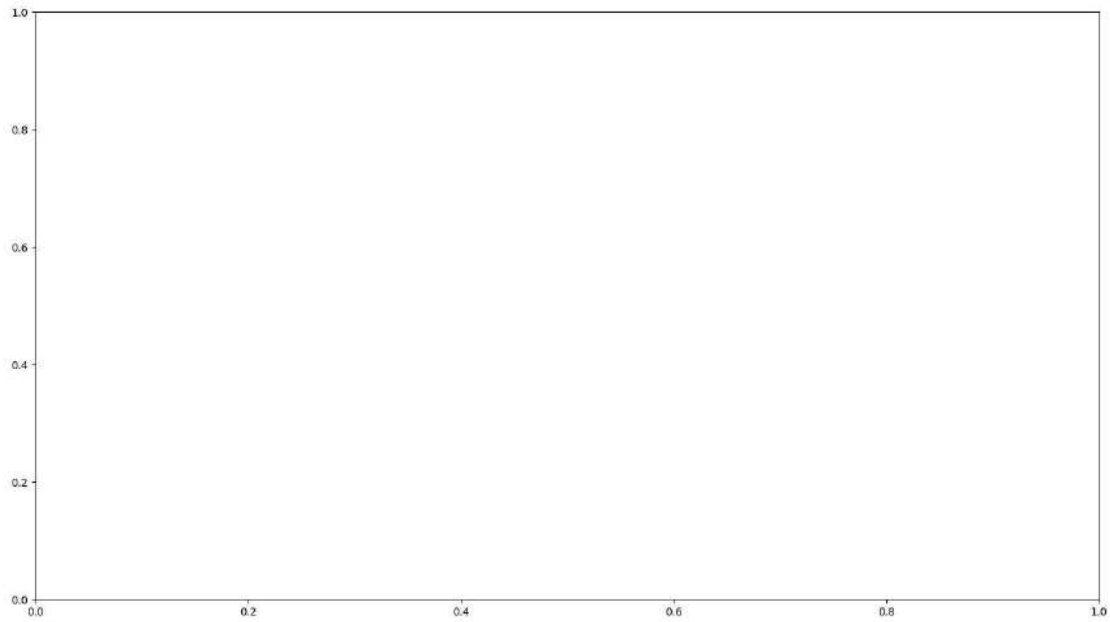
```
(array([  91,   92,   93, ..., 4800, 4801, 4802], dtype=int64),)
(array([ 265,  265,  321, ..., 4801, 4802, 4802], dtype=int64), array([0, 1, 0,
..., 1, 0, 1], dtype=int64))
(array([ 199,  322,  328,  557,  809,  828, 1525, 2516, 2638, 2799, 2912,
       3766, 4023], dtype=int64),)
```

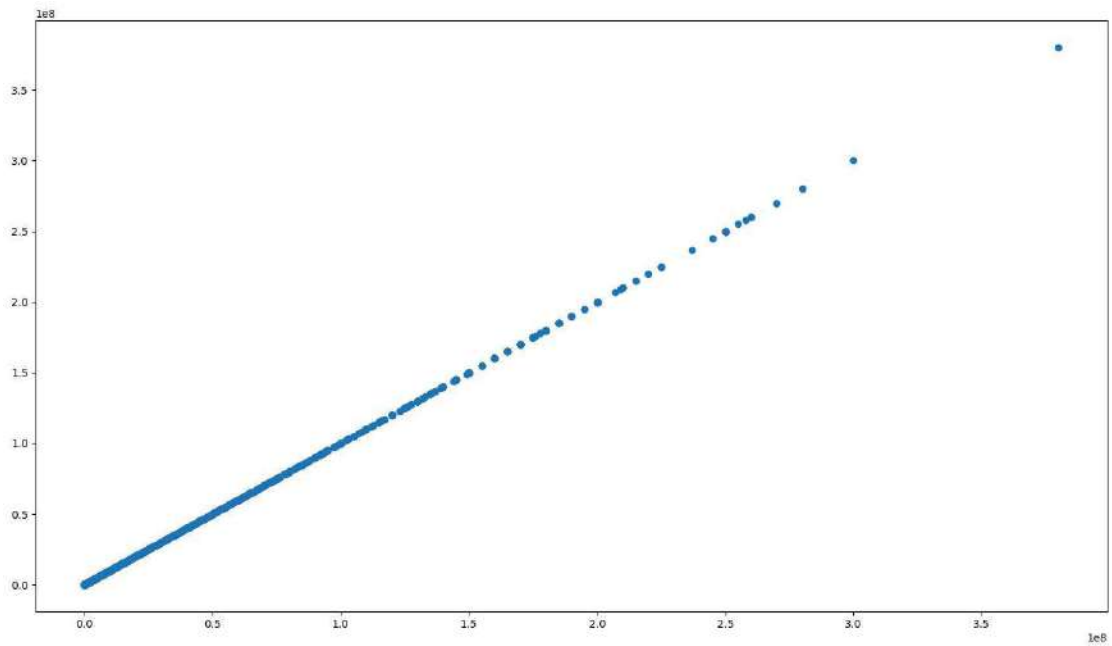
In [33]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

In [36]:
```
z = np.abs(stats.zscore(df1['budget']))
print(z)
```

```
        budget    budget
0     2.073467  5.107181
1     2.073467  6.654402
2     2.073467  5.303653
3     2.073467  5.426449
4     2.073467  5.672039
...        ...       ...
4798  0.897317  0.707916
4799  0.905174  0.713098
4800  0.905509  0.713319
4801  0.905509  0.713319
4802  0.905509  0.713319

[4803 rows x 2 columns]
```

In [41]:
```
ig, ax = plt.subplots(figsize = (18,10))
ax.scatter(df1['budget'], df1['budget'])
plt.show()

ax.set_xlabel('(Proportion non-retail business acres)/(town)')
ax.set_ylabel('(Full-value property-tax rate)/($10,000)')
```





Out[41]: Text(4.444444444444452, 0.5, '(Full-value property-tax rate)/($10,000)')

In [43]:
```
threshold = 0.18
sample_outliers = np.where(z <threshold)
sample_outliers
```

Out[43]:
```
(array([  83,   83,  379, ..., 4036, 4039, 4586], dtype=int64),
 array([0, 1, 0, ..., 1, 1, 1], dtype=int64))
```

In [44]:
```
sorted_rscore= sorted(df1['budget'])
```

In [45]:
```
sorted_rscore
```

Out[45]:
```
['budget', 'budget']
```

In [49]:
```
IQR = q3-q1
lwr_bound = q1-(1.5*IQR)
```

```
upr_bound = q3+(1.5*IQR)
print(lwr_bound, upr_bound)
```

```
-110000000.0 194000000.0
```

In [ ]:

```
r_outliers = []
for i in sorted_rscore:
    if (i<lwr_bound or i>upr_bound):
        r_outliers.append(i)
print(r_outliers)
```

In [47]:

```
new_df=df1
for i in sample_outliers:
    new_df.drop(i,inplace=True)
new_df
```

Out[47]:

	index	budget	m score	budget	genres	
2	2	80000000.0	80000000.0	245000000	Action Adventure Crime	http://www.sonypictur
3	3	80000000.0	80000000.0	250000000	Action Crime Drama Thriller	http://www.
4	4	80000000.0	80000000.0	260000000	Action Adventure Science Fiction	http://movie
5	5	80000000.0	80000000.0	258000000	Fantasy Action Adventure	http://www.sonypictu
6	6	80000000.0	80000000.0	260000000	Animation Family	http://disney.go.com/
...	...	...	...	...	...	
4798	4798	220000.0	220000.0	220000	Action Crime Thriller	
4799	4799	9000.0	9000.0	9000	Comedy Romance	
4800	4800	0.0	0.0	0	Comedy Drama Romance TV Movie	http://www.hallmarkchanne

	index	budget	m score	budget	genres	
4801	4801	0.0	0.0	0	NaN	htt
4802	4802	0.0	0.0	0	Documentary	

3986 rows × 26 columns

In [9]:
```
df_stud=df1
ninetieth_percentile = np.percentile(df_stud['budget'], 90)
b = np.where(df_stud['budget']>ninetieth_percentile,
ninetieth_percentile, df_stud['budget'])
print("New array:",b)
```

```
New array: [80000000. 80000000. 80000000. ...        0.        0.        0.]
```

In [11]:
```
df_stud.insert(1,"budget",b,True)
df_stud
```

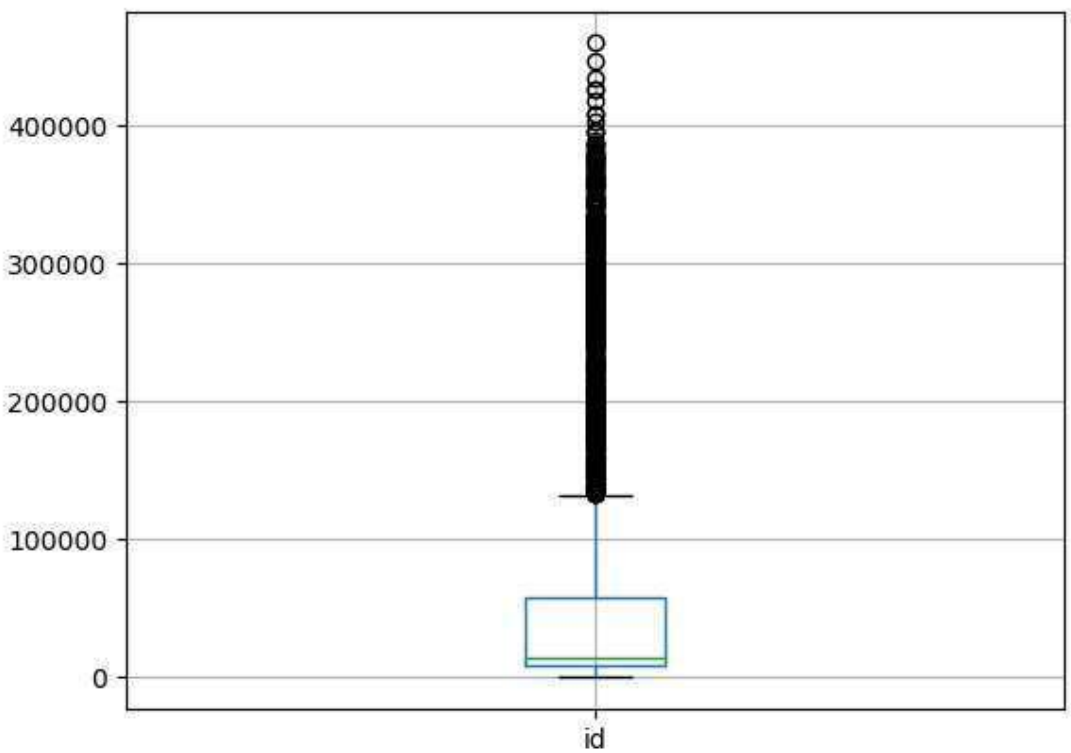
Out[11]:

	index	budget	m score	budget	genres	
0	0	80000000.0	80000000.0	237000000	Action Adventure Fantasy Science Fiction	http:/
1	1	80000000.0	80000000.0	300000000	Adventure Fantasy Action	http://disney.go.com
2	2	80000000.0	80000000.0	245000000	Action Adventure Crime	http://www.sonypictu
3	3	80000000.0	80000000.0	250000000	Action Crime Drama Thriller	http://www.
4	4	80000000.0	80000000.0	260000000	Action Adventure Science Fiction	http://movie
...	...	...	...	...	...	
4798	4798	220000.0	220000.0	220000	Action Crime Thriller	
4799	4799	9000.0	9000.0	9000	Comedy Romance	
4800	4800	0.0	0.0	0	Comedy Drama Romance TV Movie	http://www.hallmarkchanne
4801	4801	0.0	0.0	0	NaN	htt
4802	4802	0.0	0.0	0	Documentary	

index	budget	m score	budget	genres

4803 rows × 26 columns

In [12]:
```
col = ['id']
df1.boxplot(col)
median=np.median(sorted_rscore)
median
refined_df1=df1
```



In [13]:
```
refined_df1['id'] = np.where(refined_df1['id'] <lwr_bound, median,refined_df1['i
refined_df1
```

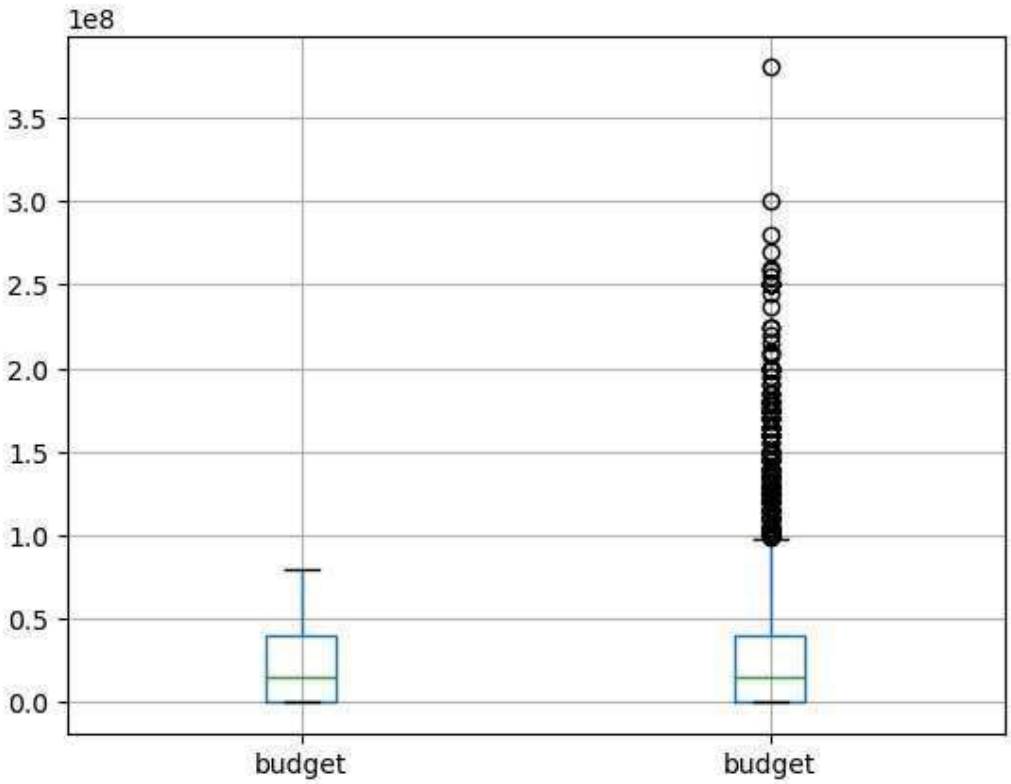
Out[13]:

	index	budget	m score	budget	genres	
0	0	80000000.0	80000000.0	237000000	Action Adventure Fantasy Science Fiction	http:/
1	1	80000000.0	80000000.0	300000000	Adventure Fantasy Action	http://disney.go.com
2	2	80000000.0	80000000.0	245000000	Action Adventure Crime	http://www.sonypictu
3	3	80000000.0	80000000.0	250000000	Action Crime Drama Thriller	http://www.
4	4	80000000.0	80000000.0	260000000	Action Adventure Science Fiction	http://movie
...	...	...	...	...	...	
4798	4798	220000.0	220000.0	220000	Action Crime Thriller	
4799	4799	9000.0	9000.0	9000	Comedy Romance	
4800	4800	0.0	0.0	0	Comedy Drama Romance TV Movie	http://www.hallmarkchanne
4801	4801	0.0	0.0	0	NaN	htt
4802	4802	0.0	0.0	0	Documentary	

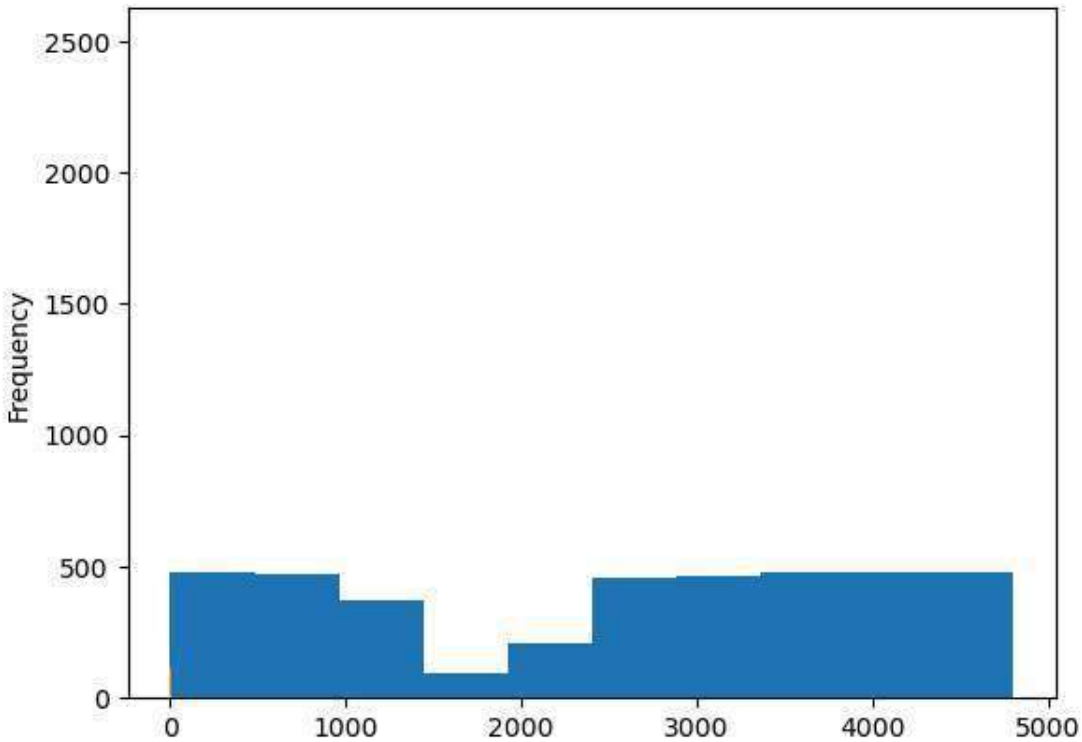
index	budget	m score	budget	genres

4803 rows × 26 columns

In [18]:
```
col = ['budget']
refined_df1.boxplot(col)
plt.show()
```



In [52]:
```
import matplotlib.pyplot as plt
new_df['index'].plot(kind = 'hist')
df1['log_math'] = np.log10(df1['index'])
df1['log_math'].plot(kind = 'hist')
plt.show()
```



In [16]:
```
x=np.array([95,85,80,70,60])
y=np.array([85,95,70,65,70])
```

In [17]:
```
model= np.polyfit(x, y, 1)
model
```

Out[17]:
```
array([ 0.64383562, 26.78082192])
```

In [18]:
```
predict = np.poly1d(model)
predict(65)
```

Out[18]:
```
68.63013698630137
```

In [19]:
```
y_pred= predict(x)
y_pred
```

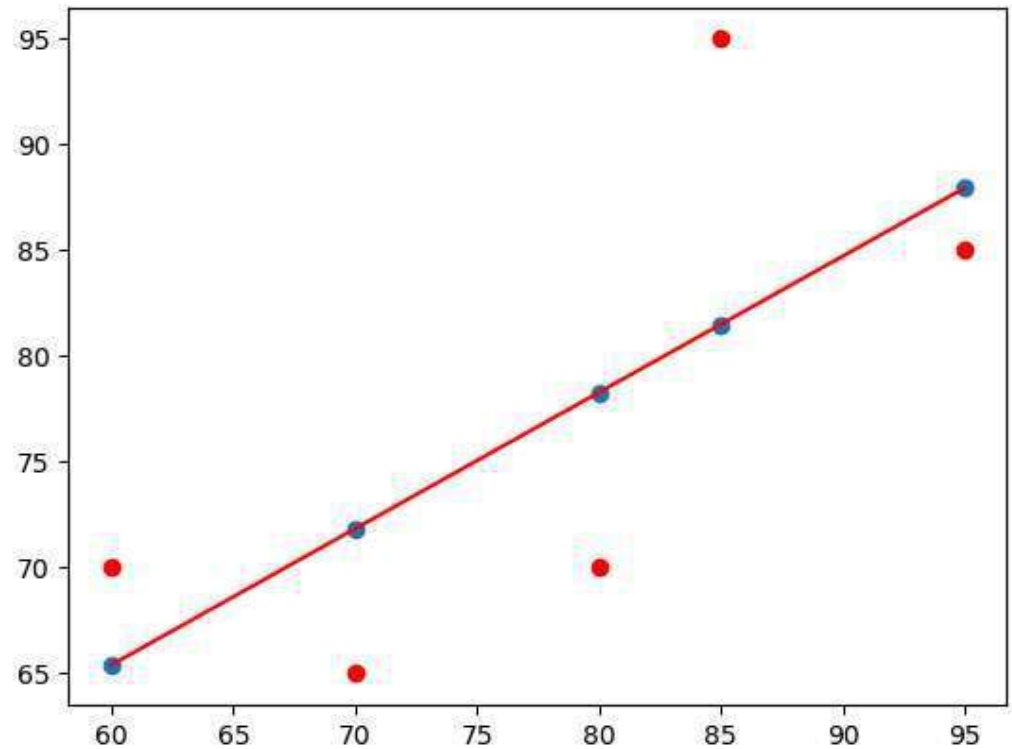
Out[19]:
```
array([87.94520548, 81.50684932, 78.28767123, 71.84931507, 65.4109589 ])
```

In [20]:
```
from sklearn.metrics import r2_score
r2_score(y, y_pred)
```

Out[20]:
```
0.4803218090889326
```

In [21]:
```
y_line = model[1] + model[0]* x
plt.plot(x, y_line, c = 'r')
plt.scatter(x, y_pred)
plt.scatter(x,y,c='r')
```

Out[21]:
```
<matplotlib.collections.PathCollection at 0x15f1de62850>
```



In [25]:
```
x = df1.drop(['budget'], axis = 1)
y = df1['budget']
```

In [34]:
```
print(df1.isnull().sum())
```

```
index                     0
budget                    0
genres                   28
homepage               3091
id                        0
keywords                412
original_language         0
original_title            0
overview                  3
popularity                0
production_companies      0
production_countries      0
release_date              1
revenue                   0
runtime                   2
spoken_languages          0
status                    0
tagline                 844
title                     0
vote_average              0
vote_count                0
cast                     43
crew                      0
director                 30
dtype: int64
```

In [36]:
```
X = df1.iloc[:,0:13]
y = df1.iloc[:,-1]
```

```
X
y
```

Out[36]:
```
0           James Cameron
1          Gore Verbinski
2              Sam Mendes
3       Christopher Nolan
4          Andrew Stanton
                ...
4798     Robert Rodriguez
4799         Edward Burns
4800          Scott Smith
4801          Daniel Hsia
4802     Brian Herzlinger
Name: director, Length: 4803, dtype: object
```

In [37]:
```
df1["tagline"].value_counts(normalize=True)
```

Out[37]:
```
tagline
Based on a true story.                          0.000758
From zero to hero.                              0.000505
The only way out is down.                       0.000505
Be careful what you wish for.                   0.000505
What could go wrong?                            0.000505
                                                   ...
Life is Not Child-Proof.                        0.000253
Every war has a beginning.                      0.000253
First came love... then came Reverend Frank.    0.000253
Get off the bench and get into the game.        0.000253
A New Yorker in Shanghai                        0.000253
Name: proportion, Length: 3944, dtype: float64
```

In [38]:
```
x=df1.drop(["tagline"],axis=1)
y=df1["tagline"]
```

In [39]:
```
x
```

Out[39]:

	index	budget	genres	homepage	
0	0	237000000	Action Adventure Fantasy Science Fiction	http://www.avatarmovie.com/	
1	1	300000000	Adventure Fantasy Action	http://disney.go.com/disneypictures/pirates/	
2	2	245000000	Action Adventure Crime	http://www.sonypictures.com/movies/spectre/	2
3	3	250000000	Action Crime Drama Thriller	http://www.thedarkknightrises.com/	
4	4	260000000	Action Adventure Science Fiction	http://movies.disney.com/john-carter	
...	...	...	...	...	
4798	4798	220000	Action Crime Thriller	NaN	
4799	4799	9000	Comedy Romance	NaN	
4800	4800	0	Comedy Drama Romance TV Movie	http://www.hallmarkchannel.com/signedsealeddel...	2
4801	4801	0	NaN	http://shanghaicalling.com/	1

	index	budget	genres	homepage
4802	4802	0	Documentary	NaN

4803 rows × 23 columns

In [40]:
```
y
```

Out[40]:
```
0                            Enter the World of Pandora.
1         At the end of the world, the adventure begins.
2                                A Plan No One Escapes
3                                      The Legend Ends
4                  Lost in our world, found in another.
                                 ...
4798    He didn't come looking for trouble, but troubl...
4799    A newlywed couple's honeymoon is upended by th...
4800                                                  NaN
4801                            A New Yorker in Shanghai
4802                                                  NaN
Name: tagline, Length: 4803, dtype: object
```

In [41]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,random_
```

In [42]:
```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(3842, 13)
(961, 13)
(3842,)
(961,)
```

In [43]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler
```

Out[43]:
```
▾ MinMaxScaler
MinMaxScaler()
```

In [44]:
```
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler

X, y = make_classification(random_state=42)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
pipe = make_pipeline(StandardScaler(), LogisticRegression())
pipe.fit(X_train, y_train)
```

Out[44]:
```
▸ Pipeline
  ▸ StandardScaler
  ▸ LogisticRegression
```

In [45]:
```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X_train,y_train)
```

Out[45]:
```
▾ LogisticRegression
LogisticRegression()
```

In [46]:
```
y_pred=logreg.predict(X_test)
```

In [47]:
```
from sklearn.linear_model import LinearRegression
```

In [48]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
model.fit(X_train, y_train)
```

Out[48]:
```
▸ Pipeline
  ▸ StandardScaler
  ▸ LinearRegression
```

In [49]:
```
model.score(X_test,y_test)
```

Out[49]:
```
0.6803193862233878
```

In [50]:
```
X_train
```

Out[50]:
```
array([[-1.06239353, -2.68317954,  0.33848384, ..., -0.35316629,
         0.32579632,  0.1943843 ],
       [-0.79047446, -0.07873421, -1.69246463, ...,  1.09419152,
        -0.12578692,  0.05572491],
       [-0.22096417, -0.54561186, -0.57117899, ...,  0.64084286,
        -0.28110029,  1.79768653],
       ...,
       [ 0.84064355,  0.37531604, -0.96697614, ...,  0.42545756,
         0.76041466,  0.78580016],
       [ 0.49403019,  0.63067073,  1.1487657 , ..., -2.84854262,
        -0.37061433,  0.77169871],
       [-0.42018682, -0.24038388,  0.9843224 , ..., -0.99835404,
         0.23421473,  1.55050049]])
```

In [51]:
```
y_train
```

Out[51]:
```
array([1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0,
       1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 0, 1])
```

In [52]:
```
X_test
```

```
Out[52]: array([[ 8.35692112e-01,  5.88465251e-01,  8.65897875e-02,
                 -7.96895255e-01,  5.77072127e-01, -3.75120699e-01,
                  2.54420843e-01, -1.55677235e-01, -1.12970685e+00,
                  5.29804178e-01, -5.75002146e-01,  1.00629281e+00,
                  1.16778206e+00,  1.44156862e+00, -5.76891870e-01,
                 -1.49518014e-01, -2.47164450e+00, -6.03985187e-01,
                 -2.03045386e-01,  3.71145873e-01],
                [ 7.66080278e-01, -2.65461268e+00,  1.49268857e+00,
                 -8.26880354e-01, -2.26478892e-01,  2.25661188e+00,
                 -7.47211679e-01, -2.71123601e-01,  1.22693322e+00,
                 -1.00154076e-01,  2.36867367e+00, -9.98385439e-01,
                 -2.13672943e-02, -2.03673750e-01,  3.01791900e-01,
                  1.17077461e+00, -8.77982587e-01, -8.03178951e-01,
                  3.67365507e-01,  9.13584626e-01],
                [-6.22649362e-01, -1.44723231e+00, -1.50472037e+00,
                 -3.70367970e-02, -4.29302225e-01,  1.25896077e+00,
                 -1.45755150e+00,  7.60055964e-01,  1.94607455e-01,
                 -7.42470586e-01,  1.27988990e+00,  4.36024464e-02,
                  8.24397529e-02, -1.32002251e+00,  1.69505104e+00,
                  6.63522604e-01, -6.11769091e-01, -8.31055726e-02,
                 -6.92420980e-01, -1.40631746e+00],
                [-1.32633746e-01, -1.29826331e+00,  7.09451817e-01,
                  8.47421677e-01, -5.35328186e-01,  1.25291095e+00,
                 -2.59546678e-01, -4.35486371e-01, -9.74529305e-01,
                  1.10708067e+00,  1.09885263e+00,  6.72573701e-01,
                  5.13105798e-01, -1.20381164e-01,  1.89988193e+00,
                  7.03851925e-01, -2.17266955e+00,  1.90499680e-01,
                 -9.05332823e-02,  3.31980331e-01],
                [ 9.98010110e-01,  2.23245441e+00, -1.90682401e-01,
                 -1.03990593e+00,  6.12773905e-01, -4.09257060e-01,
                  4.95681888e-01,  2.17432873e-01, -2.89625538e+00,
                  2.08837470e+00, -2.58590856e+00,  9.16327675e-01,
                  8.70067731e-01, -1.39589628e-01,  3.46488476e-01,
                  3.24237337e-01,  1.10818282e+00,  1.91403135e+00,
                 -1.05341556e+00, -6.23768961e-01],
                [ 8.83109777e-01, -1.76254901e+00,  9.19228935e-01,
                 -7.51791050e-01, -4.26357596e-01,  1.49919830e+00,
                  1.08004807e+00, -6.68144086e-01, -7.78371970e-02,
                 -1.80479914e-01,  1.57233676e+00, -2.91811269e-01,
                  1.87329778e+00,  3.19310757e+00, -1.56319081e+00,
                  7.78139695e-01,  2.98752908e-01, -1.43827798e+00,
                  1.14844571e+00,  1.13270411e-01],
                [ 9.96047686e-01,  1.06660430e-02, -6.43518230e-01,
                 -2.50833016e-01,  1.32819414e+00,  1.38991764e+00,
                  2.68592279e-01,  9.27840128e-01, -7.56795089e-01,
                 -1.42181067e+00, -5.67724525e-01,  1.18570415e+00,
                  5.70131249e-02,  1.50133365e+00,  7.18953311e-01,
                  1.22541983e+00, -3.22679839e-01,  2.16500234e+00,
                  5.56230009e-01,  4.55887772e-01],
                [-1.80313968e+00, -4.62814324e-01, -2.68530648e-01,
                  8.95038314e-01, -4.83061052e-01,  5.73042476e-01,
                 -7.19153319e-01, -8.91192225e-01, -1.58413594e+00,
                  2.67126651e-01,  3.41293945e-01, -7.08406767e-01,
                 -2.15181540e+00,  5.08725023e-01, -1.51371439e+00,
                  3.62053684e-01, -1.58119070e+00,  8.96839316e-01,
                  1.46793012e-01,  1.61222063e+00],
                [ 2.24684726e-01, -1.01695056e+00,  1.54210995e+00,
                  1.26258410e+00, -5.51858146e-01,  5.54483984e-01,
                  1.20836623e+00,  2.00609289e+00,  9.32590832e-01,
                 -1.41836573e+00,  1.03110238e+00,  1.72696385e+00,
```

```
  2.06150358e+00, -1.76080881e+00, -3.99636186e-01,
  1.75930005e-01, -1.52565631e+00,  1.84551303e-01,
  2.55819929e+00, -5.64247599e-01],
[-2.02514259e+00,  2.91021981e-02, -4.74945311e-01,
 -1.14736441e-01,  5.04987279e-01,  8.89621380e-02,
  4.04981711e-01, -6.53329233e-01,  1.86454315e-01,
 -6.61786465e-01, -7.13352412e-02,  2.14394409e+00,
  1.76545424e+00,  8.52433335e-01,  6.33919022e-01,
  8.71420582e-02, -7.92520738e-01, -3.34501236e-01,
  8.65755194e-01, -1.20029641e+00],
[-4.40044487e-01, -2.18003951e-01,  6.98020850e-02,
  1.02330610e-02, -9.81508651e-01,  4.94298226e-01,
  6.62130675e-01, -3.85313597e-01,  1.30740577e-01,
  1.44127329e+00,  7.12364098e-02,  1.63241130e+00,
  1.13517345e-01, -1.43586215e+00, -1.43014138e+00,
  3.67831850e-01,  1.16316375e+00, -6.00216877e-01,
  4.62103474e-01,  1.99059696e-01],
[ 1.60573981e-01,  2.68438233e-01, -1.17011303e+00,
 -1.48489797e+00, -2.55392113e+00,  1.19664076e+00,
  7.59155160e-01, -1.80198044e+00,  3.04602015e-03,
  4.36938170e-01, -8.08044635e-01, -3.88176819e-01,
  5.41462728e-01,  1.19064627e+00,  1.70416222e-01,
  1.13446163e+00,  9.49554135e-01, -2.24765402e-01,
  9.34319911e-01, -1.36687870e+00],
[ 5.76556963e-01,  1.62748917e+00, -6.46572884e-01,
 -9.55540441e-01, -1.60644632e+00, -1.42695838e+00,
  8.81639757e-01, -1.08154800e+00,  3.11250155e-01,
  3.07888081e+00, -1.43483867e+00,  1.64496771e+00,
  1.68714164e+00,  1.11957491e+00, -2.49036040e-01,
 -7.56006002e-01, -1.27917591e-01, -1.42225371e+00,
  2.03463636e-01, -7.56350745e-01],
[ 3.26927374e-01,  2.40437346e+00,  6.32781866e-01,
  7.70865194e-01, -1.47858625e+00, -2.39955005e+00,
  2.48220586e-01,  2.27069286e+00, -2.19100529e-01,
  8.29405581e-01, -2.00347738e+00,  1.84183792e-02,
  1.81866255e-01, -2.21113531e+00,  1.67643731e+00,
 -1.37314120e+00,  2.35614558e-01, -4.15287914e-01,
  1.14375404e+00,  3.38496407e-01],
[ 5.22835488e-01,  1.77872961e+00, -2.57376537e-01,
  4.36323670e-01,  3.80034782e-02, -4.86290939e-02,
  6.47195940e-01, -1.66858407e+00, -5.73700004e-01,
 -2.43545923e-02, -2.17105282e+00, -3.20347308e-01,
  3.99223123e-01,  2.14227036e+00,  4.24165946e-01,
  5.02300750e-01,  1.72754317e+00, -1.02279257e+00,
  1.20031327e-01,  6.13517973e-01],
[ 1.79894151e-01, -2.22100055e-01,  1.05375506e-01,
  9.40771188e-01, -9.82487394e-01,  1.56010259e+00,
  3.19781934e-01, -1.33402549e+00,  1.39200229e+00,
  9.18316606e-01, -3.48984841e-01, -9.88591107e-01,
 -6.01367644e-01, -1.57050060e+00, -1.10358932e+00,
  1.30373602e+00, -9.89628137e-01, -9.68344455e-01,
 -2.24633150e-01,  5.50052099e-01],
[-1.04449219e-01, -1.46447269e+00, -2.70323229e+00,
  2.38368983e-01,  9.75197630e-01,  1.07954187e+00,
 -1.83063290e+00,  6.77875320e-01, -1.68821723e-01,
  7.00521631e-02,  1.37271070e+00,  9.50307592e-01,
 -6.54075683e-01,  1.16187830e+00,  1.71061337e+00,
  5.00476965e-01, -9.27353134e-01,  1.00104609e+00,
  5.01094170e-01,  1.89581617e-01],
[ 8.30335817e-01,  9.46218141e-01, -4.44293260e-01,
```

```
  3.33662105e-01,  1.03753994e+00, -9.31125899e-01,
 -9.22165324e-01,  3.77300493e-01, -8.56083826e-01,
  7.15662372e-02, -7.93713869e-01, -4.59360900e-01,
  7.56988617e-01, -4.77657447e-01, -8.49844369e-01,
 -5.28785077e-01,  4.78979826e-01, -9.78763716e-01,
 -5.10016399e-01, -2.69874935e-01],
[ 6.44744148e-02, -2.39384676e-01,  1.48667457e-01,
  5.99928730e-01,  1.53075083e+00,  1.56931739e+00,
 -3.02469686e-01, -3.37085971e-01, -1.97546657e+00,
 -9.39335394e-01, -3.31376097e-01, -7.04987781e-02,
 -6.13402664e-01, -1.44087556e-01,  4.86501644e-01,
  1.30654200e+00, -1.20969474e+00,  1.49072614e+00,
  1.21876185e+00, -2.13442871e-01],
[ 9.63879117e-01,  1.39398330e+00,  5.68983079e-01,
  2.57970934e+00, -8.03674568e-01,  2.59956720e-01,
 -2.03580360e-01,  1.62839662e+00,  2.21052300e+00,
 -5.57491785e-01, -1.82037691e+00,  4.38114720e-02,
 -3.79127741e-01, -1.36980298e+00, -1.47002002e-01,
  6.55740858e-01, -8.82820484e-02, -5.53588242e-01,
  1.63911681e+00,  1.67770081e+00],
[-1.71313453e+00,  1.74355119e+00, -1.06762043e+00,
 -5.99375023e-01,  5.24369972e-03, -1.51042899e+00,
  5.14438834e-01, -1.42379485e-01,  1.35387237e+00,
 -1.14539845e-01, -1.54446032e+00,  1.17944012e+00,
  1.20295632e-01,  1.23781631e+00, -4.69175652e-01,
 -7.93836362e-01, -1.59442766e+00,  6.22849932e-01,
  4.69805938e-02, -4.50065471e-01],
[-5.17611299e-01,  5.06885374e-01,  5.83928185e-01,
 -5.30868773e-01, -4.89439443e-01, -1.73597973e+00,
  1.10870358e+00, -3.59292091e-01,  2.23787952e-01,
 -1.64228961e-02,  6.84561582e-02, -3.31269729e-02,
  5.90654831e-01,  1.18839327e+00,  1.79455786e+00,
 -1.37111696e+00,  2.52693243e+00,  1.84670733e+00,
  1.04416088e+00,  6.81891490e-01],
[ 1.06667469e+00, -4.79369952e-01,  2.87448229e-02,
  1.46713686e-01,  1.20650897e+00, -1.08324727e+00,
  4.64365482e-02,  1.27845186e+00,  1.16929559e+00,
  1.38215899e+00,  1.02255619e+00,  8.20482181e-01,
  1.91099068e-01,  6.48709888e-01,  5.07274031e-01,
 -1.09939128e+00, -1.67118080e-01, -3.93338812e-01,
 -8.16935671e-01,  3.68673309e-01],
[-1.37448515e-01, -4.68094222e-01,  2.17094272e+00,
  7.42127491e-01,  7.54336389e-02,  6.24530321e-01,
  5.51485376e-01, -1.75885812e-01,  9.52874547e-01,
  1.61227826e+00,  3.27251881e-01,  5.11202600e-01,
  1.23204807e-01,  1.31491445e+00,  1.37365855e+00,
  4.05708897e-01,  1.63996453e+00, -8.43246595e-01,
 -1.60196581e+00, -2.46062488e-01],
[ 7.58045582e-02,  2.48904785e+00,  2.44966571e-01,
 -3.21385842e-01,  4.12931454e-01, -2.52343407e+00,
  2.32049937e-01, -5.06943175e-01, -6.77161712e-01,
  9.75119733e-01, -2.05832072e+00, -1.22467728e-02,
 -4.71038306e-01, -1.47057382e-01, -8.97254371e-01,
 -1.45612516e+00, -8.25497197e-01,  2.43687211e-01,
 -5.63724553e-01, -8.22220396e-01]])
```

In [53]:
```
y_test
```

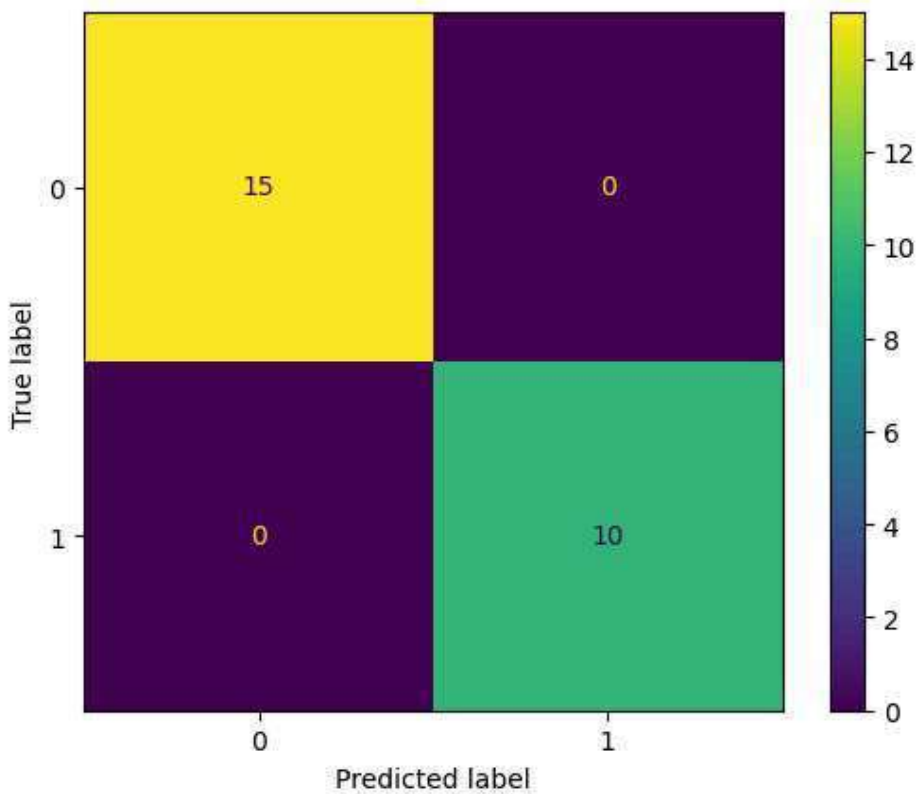
Out[53]:
```
array([0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       1, 1, 0])
```

In [54]:
```
from sklearn.metrics import precision_score,ConfusionMatrixDisplay, confusion_ma
cm= confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix = cm)
print("Confusion matrix :")
print(cm)
```

```
Confusion matrix :
[[15  0]
 [ 0 10]]
```

In [55]:
```
disp.plot()
```

Out[55]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1236b1f11d0
>
```



In [56]:
```
true_negative =cm[0][0]
false_negative =cm[1][0]
false_positive =cm[0][1]
true_positive =cm[1][1]
```

In [57]:
```
Accuracy = (true_positive + true_negative) / (true_positive +false_positive + tr
Accuracy
# Precison
Precision = true_positive/(true_positive+false_positive)
Precision
# Recall
Recall = true_positive/(true_positive+false_negative)
Recall
# F1 Score
F1_Score = 2*(Recall * Precision) / (Recall + Precision)
F1_Score
```

Out[57]: 1.0

In [58]:
```
print("Accuracy:", Accuracy)
print("Confusion Matrix:")
print(cm)
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 1.0
Confusion Matrix:
[[15  0]
 [ 0 10]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        15
           1       1.00      1.00      1.00        10

    accuracy                           1.00        25
   macro avg       1.00      1.00      1.00        25
weighted avg       1.00      1.00      1.00        25
```

In [59]:
```
Accuracy
```

Out[59]: 1.0

In [73]:
```
Precision
```

Out[73]: 1.0

In [74]:
```
Recall
```

Out[74]: 1.0

In [75]:
```
F1_Score
```

Out[75]: 1.0

In [76]:
```
from sklearn.metrics import f1_score, confusion_matrix, roc_auc_score, roc_curve
import matplotlib as plt
```

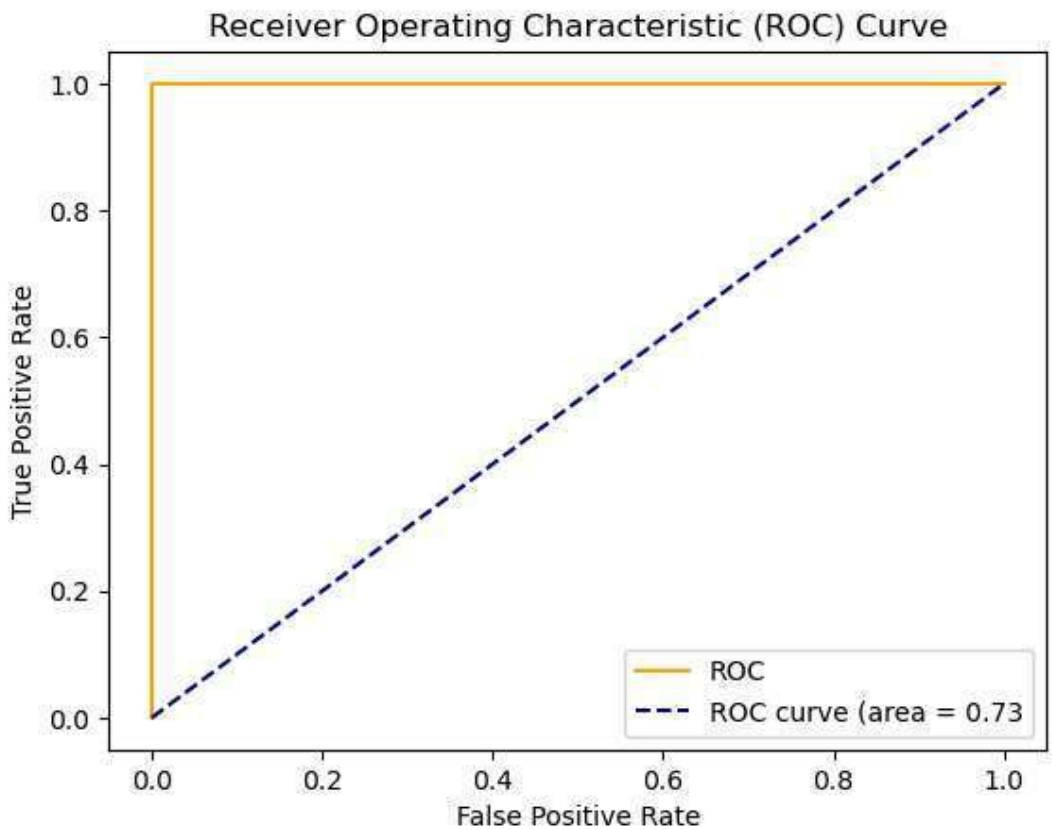
In [77]:
```
auc_score=roc_auc_score(y_test,y_pred)
```

In [78]:
```
fpr,tpr,threasholds=roc_curve(y_test,y_pred)
```

In [79]:
```
threasholds
```

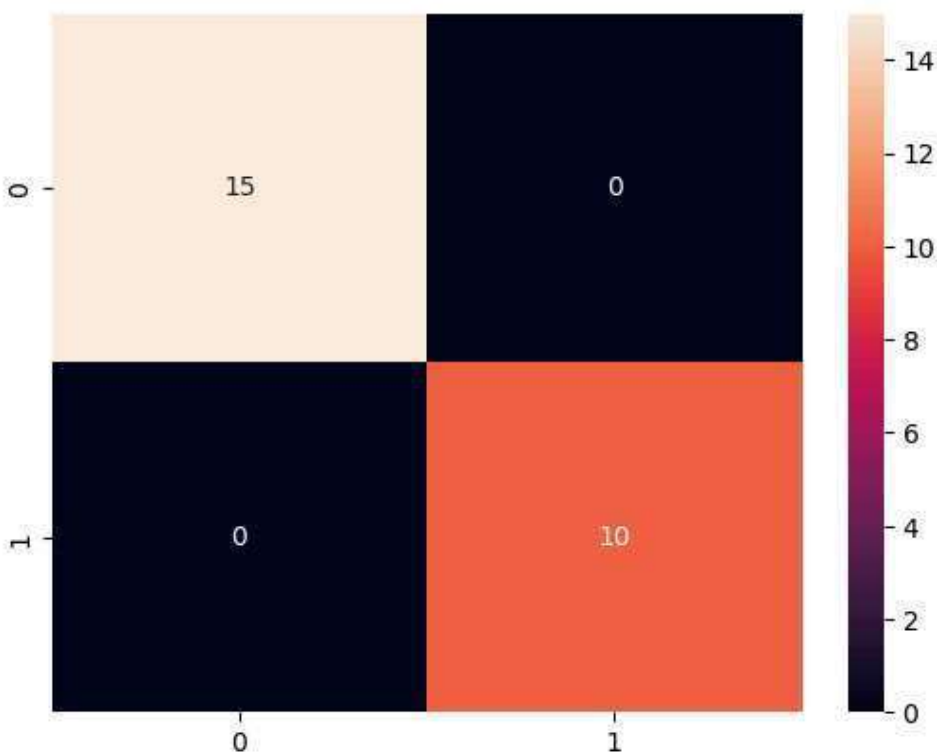
Out[79]:
```
array([inf,  1.,  0.])
```

In [80]:
```
import matplotlib.pyplot as plt
plt.plot(fpr, tpr, color='orange', label='ROC')
plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--',label='ROC curve (area
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend()
plt.show()
```



In [81]:
```
import seaborn as sns
sns.heatmap(cm, annot=True)
```

Out[81]: <Axes: >



In [34]:
```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
```

```
model.fit(X_train, y_train)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[34], line 3
      1 from sklearn.naive_bayes import GaussianNB
      2 model = GaussianNB()
----> 3 model.fit(X_train, y_train)

NameError: name 'X_train' is not defined
```

In [84]:
```
from sklearn.metrics import accuracy_score, confusion_matrix, ConfusionMatrixDis
print(accuracy_score(y_test, y_pred))
```

```
1.0
```

In [86]:
```
def get_confusion_matrix_values(y_true, y_pred):
    cm = confusion_matrix(y_true, y_pred)
    return(cm[0][0], cm[0][1], cm[1][0], cm[1][1])

TP, FP, FN, TN = get_confusion_matrix_values(y_test, y_pred)
print("TP: ", TP)
print("FP: ", FP)
print("FN: ", FN)
print("TN: ", TN)
```

```
TP:  15
FP:  0
FN:  0
TN:  10
```

In [7]:
```
import nltk
```

In [8]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Welcome\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[8]: True

In [9]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Welcome\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[9]: True

In [10]:
```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Welcome\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[10]: True

In [11]:
```
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Welcome\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Out[11]: True

In [12]:
```
text= "Tokenization is the first step in text analytics. The process of breaking
text
```

Out[12]:
```
'Tokenization is the first step in text analytics. The process of breaking down
a text paragraph into smaller chunkssuch as words or sentences is called Tokeni
zation.'
```

In [13]:
```
from nltk.tokenize import sent_tokenize
tokenized_text= sent_tokenize(text)
print(tokenized_text)
```

```
['Tokenization is the first step in text analytics.', 'The process of breaking do
wn a text paragraph into smaller chunkssuch as words or sentences is called Token
ization.']
```

In [14]:
```
from nltk.tokenize import word_tokenize
tokenized_word=word_tokenize(text)
print(tokenized_word)
```

```
['Tokenization', 'is', 'the', 'first', 'step', 'in', 'text', 'analytics', '.', 'T
he', 'process', 'of', 'breaking', 'down', 'a', 'text', 'paragraph', 'into', 'smal
ler', 'chunkssuch', 'as', 'words', 'or', 'sentences', 'is', 'called', 'Tokenizati
on', '.']
```

In [15]:
```
import re
from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)
```

```
{"she's", "you've", 'be', 'more', 'to', "aren't", "you're", 'did', 'through', 'co
uldn', "it'd", 'y', "haven't", 'from', 'doesn', 'can', "we're", 'whom', 'while',
'who', 'should', 'wasn', 'yourself', "he'd", 're', 'each', 'it', 'above', 'him',
'my', "i'll", 'am', 'until', 'any', 'with', 'only', 's', 'over', 'at', "isn't",
"he's", 'shan', 'having', 'we', "weren't", 'you', 'most', 'here', "don't", 'is',
'ours', "i've", 'which', 'does', 'doing', 'itself', 'didn', 'just', 'those', 'i
f', 'll', 'themselves', 'its', "won't", 'about', 'in', "shouldn't", "mustn't", "n
eedn't", 'an', 'these', 'that', 'then', 'them', "they've", "he'll", 'do', 'into',
'himself', 'during', 'd', 'me', "couldn't", 'ourselves', 'or', "should've", 'wha
t', 'and', 'same', 'a', 'hasn', "she'll", 'why', 'as', 'by', "we've", 'being', 'o
nce', 'between', 'on', "shan't", 'ain', "hadn't", "hasn't", 'how', 'our', 'would
n', 'all', "mightn't", 'the', 'i', "they're", 'have', 'before', 'against', 'nor',
'mustn', 'out', 'after', 'hadn', 'herself', "she'd", 'some', 'too', 'again', 'bec
ause', 'were', "wouldn't", 'yours', 'aren', 'was', 'up', "wasn't", 'don', 'both',
'haven', 'her', 'own', 'off', "you'd", 'such', 'for', 'm', 'than', 'yourselves',
'but', 'his', 'your', 'hers', "didn't", 'mightn', 'has', 'below', "doesn't", 'o',
'few', 'had', 'are', "we'd", "i'm", 'needn', "they'll", 've', 'will', 'weren', 'h
e', "it'll", 'their', 'ma', 'under', "that'll", "i'd", 'down', "they'd", 'where',
'there', 'further', 'of', 'this', "you'll", 'shouldn', 'no', 'isn', 'other', 'no
w', 'so', "we'll", 'she', "it's", 'theirs', 'been', 'when', 'they', 'myself',
't', 'won', 'not', 'very'}
```

In [16]:
```
text= "How to remove stop words with NLTK library inPython?"
text= re.sub('[^a-zA-Z]', ' ',text)
tokens = word_tokenize(text.lower())
```

```
filtered_text=[]
for w in tokens:
    if w not in stop_words:
        filtered_text.append(w)
print("Tokenized Sentence:",tokens)
print("Filterd Sentence:",filtered_text)
```

```
Tokenized Sentence: ['how', 'to', 'remove', 'stop', 'words', 'with', 'nltk', 'lib
rary', 'inpython']
Filterd Sentence: ['remove', 'stop', 'words', 'nltk', 'library', 'inpython']
```

In [17]:
```
from nltk.stem import PorterStemmer
e_words= ["wait", "waiting", "waited","waits"]
ps =PorterStemmer()
for w in e_words:
    rootWord=ps.stem(w)
print(rootWord)
```

```
wait
```

In [18]:
```
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
text = "studies studying cries cry"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
    print("Lemma for {} is {}".format(w,wordnet_lemmatizer.lemmatize(w)))
```

```
Lemma for studies is study
Lemma for studying is studying
Lemma for cries is cry
Lemma for cry is cry
```

In [19]:
```
import nltk
from nltk.tokenize import word_tokenize
data="The pink sweater fit herperfectly"
words=word_tokenize(data)
print(data)
print(words)
```

```
The pink sweater fit herperfectly
['The', 'pink', 'sweater', 'fit', 'herperfectly']
```

In [20]:
```
for word in words:
    print(nltk.pos_tag([word]))
```

```
[('The', 'DT')]
[('pink', 'NN')]
[('sweater', 'NN')]
[('fit', 'NN')]
[('herperfectly', 'RB')]
```

In [39]:
```
import pandas as pd
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [40]:
```
documentA = 'Jupiter is the largest Planet'
documentB = 'Mars is the fourth planet from the Sun'
print(documentA)
print(documentB)
```

```
Jupiter is the largest Planet
Mars is the fourth planet from the Sun
```

In [41]:
```
bagOfWordsA = documentA.split(' ')
bagOfWordsB = documentB.split(' ')
print(bagOfWordsA)
print(bagOfWordsB)
```

```
['Jupiter', 'is', 'the', 'largest', 'Planet']
['Mars', 'is', 'the', 'fourth', 'planet', 'from', 'the', 'Sun']
```

In [42]:
```
uniqueWords = set(bagOfWordsA).union(set(bagOfWordsB))
```

In [43]:
```
numOfWordsA = dict.fromkeys(uniqueWords, 0)
for word in bagOfWordsA:
    numOfWordsA[word] += 1
    numOfWordsB = dict.fromkeys(uniqueWords,0)
    for word in bagOfWordsB:
        numOfWordsB[word] += 1
```

In [44]:
```
def computeTF(wordDict, bagOfWords):
    tfDict = {}
    bagOfWordsCount =len(bagOfWords)
    for word, count in wordDict.items():
        tfDict[word] = count / float(bagOfWordsCount)
    return tfDict
tfA = computeTF(numOfWordsA, bagOfWordsA)
tfB = computeTF(numOfWordsB, bagOfWordsB)
```

In [45]:
```
def computeIDF(documents):
    import math
    N = len(documents)
    idfDict = dict.fromkeys(documents[0].keys(),0)
    for document in documents:
        for word, val in document.items():
            if val > 0:
                idfDict[word] += 1
    for word, val in idfDict.items():
        idfDict[word] = math.log(N /float(val))
    return idfDict
```

In [46]:
```
idfs = computeIDF([numOfWordsA, numOfWordsB])
idfs
```

Out[46]:
```
{'Mars': 0.6931471805599453,
 'is': 0.0,
 'Jupiter': 0.6931471805599453,
 'Sun': 0.6931471805599453,
 'planet': 0.6931471805599453,
 'from': 0.6931471805599453,
 'fourth': 0.6931471805599453,
 'Planet': 0.6931471805599453,
 'largest': 0.6931471805599453,
 'the': 0.0}
```

In [47]:
```
def computeTFIDF(tfBagOfWords, idfs):
    tfidf = {}
    for word, val in tfBagOfWords.items():
        tfidf[word] = val * idfs[word]
    return tfidf
tfidfA = computeTFIDF(tfA,idfs)
tfidfB = computeTFIDF(tfB,idfs)
```

```
df = pd.DataFrame([tfidfA,tfidfB])
df
```

Out[47]:

	Mars	is	Jupiter	Sun	planet	from	fourth	Planet	largest
0	0.000000	0.0	0.138629	0.000000	0.000000	0.000000	0.000000	0.138629	0.138629
1	0.086643	0.0	0.000000	0.086643	0.086643	0.086643	0.086643	0.000000	0.000000

In [70]:
```
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from nltk.corpus import stopwords
import nltk
import re
```

In [71]:
```
stop_words = set(stopwords.words('english'))
```

In [72]:
```
data = {'Text': ["Agriculture is the backbone of our economy.",
                 "Farmers work hard to grow crops for food production.",
                 "Market trends affect crop prices and demand."]}
```

In [73]:
```
df = pd.DataFrame(data)
```

In [74]:
```
def clean_text(text):
    text = text.lower()  # Lowercase
    text = re.sub(r'\W+', ' ', text)  # Remove special characters
    text = ' '.join(word for word in text.split() if word not in stop_words)  #
    return text
```

In [75]:
```
df['Cleaned_Text'] = df['Text'].apply(clean_text)
```

In [76]:
```
text = " ".join(df['Cleaned_Text'])
```

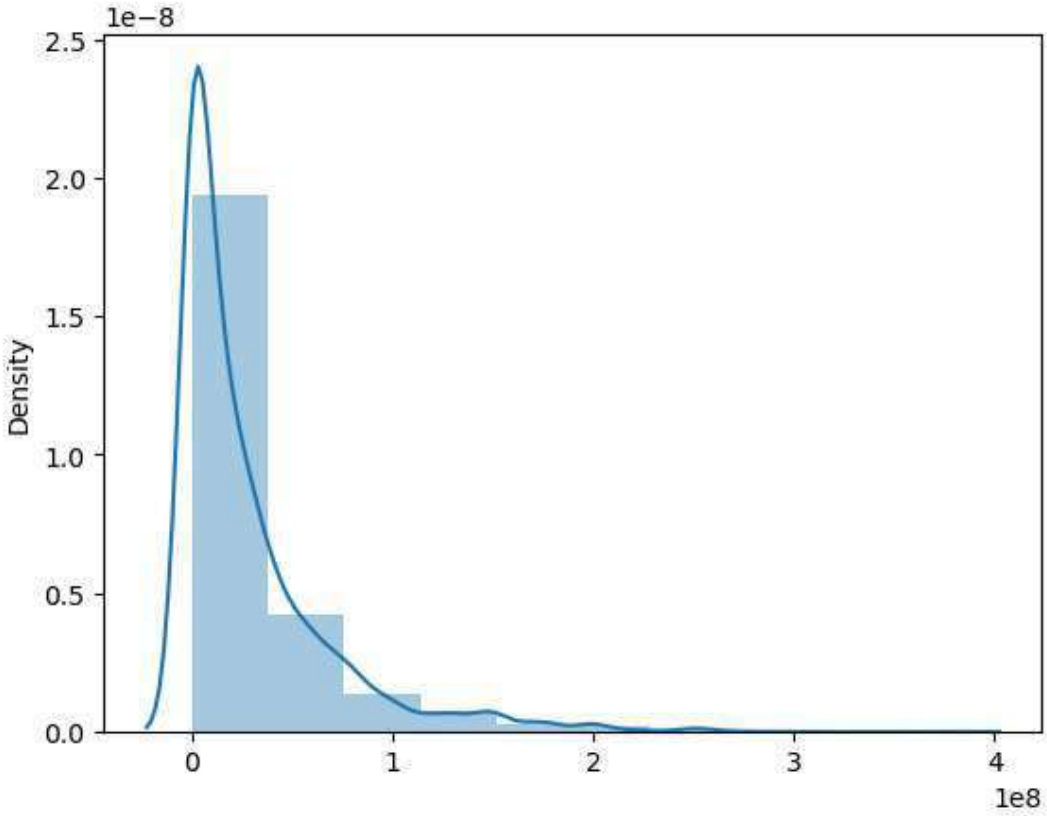
In [77]:
```
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(
```

In [79]:
```
plt.figure(figsize=(7, 4))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

In [48]:
```
sns.distplot(x = df1['budget'], bins = 10)
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_12292\1955079267.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x = df1['budget'], bins = 10)
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

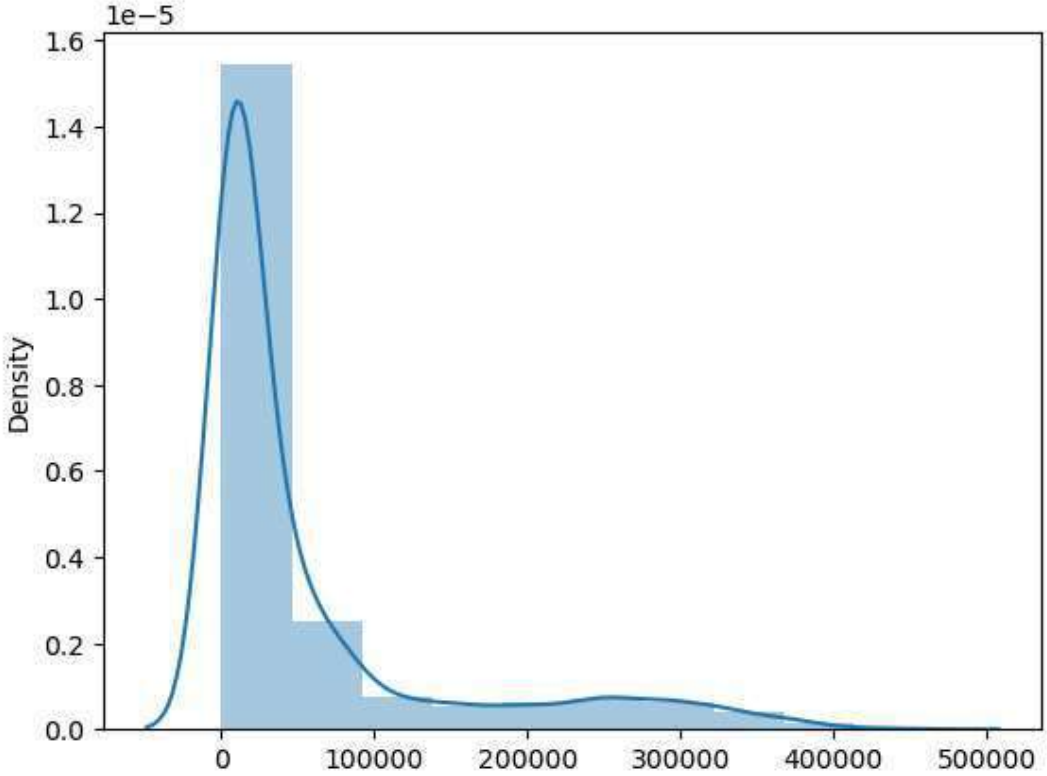
Out[48]:
```
<Axes: ylabel='Density'>
```



In [49]:
```
sns.distplot(x = df1['id'], bins = 10)
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_12292\3470013663.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x = df1['id'], bins = 10)
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

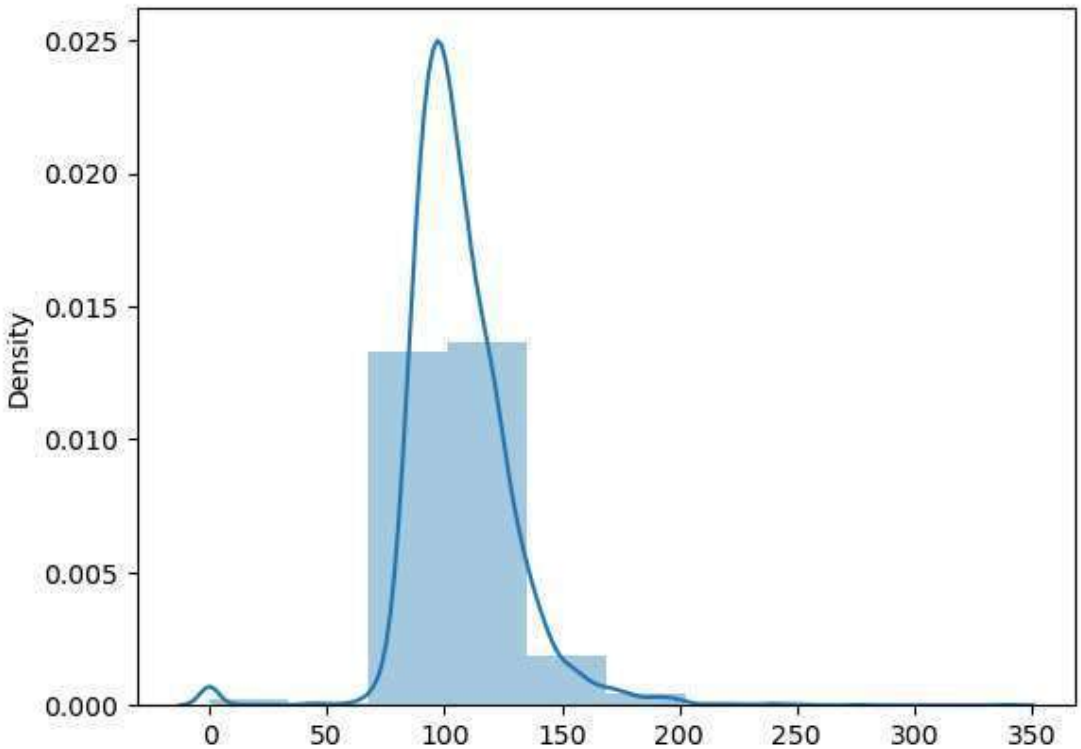
Out[49]:
```
<Axes: ylabel='Density'>
```



In [50]:
```
sns.distplot(x = df1['runtime'], bins = 10)
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_12292\1852036295.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x = df1['runtime'], bins = 10)
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

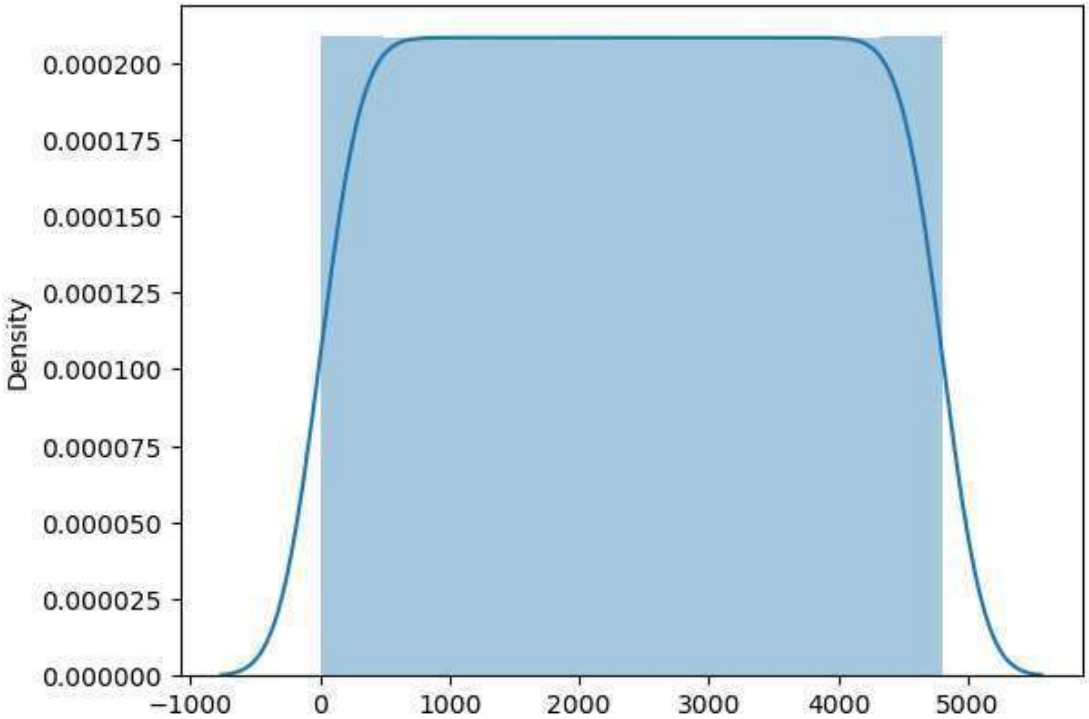
Out[50]:
```
<Axes: ylabel='Density'>
```



In [51]:
```
sns.distplot(x = df1['index'], bins = 10)
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_12292\3913942669.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x = df1['index'], bins = 10)
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

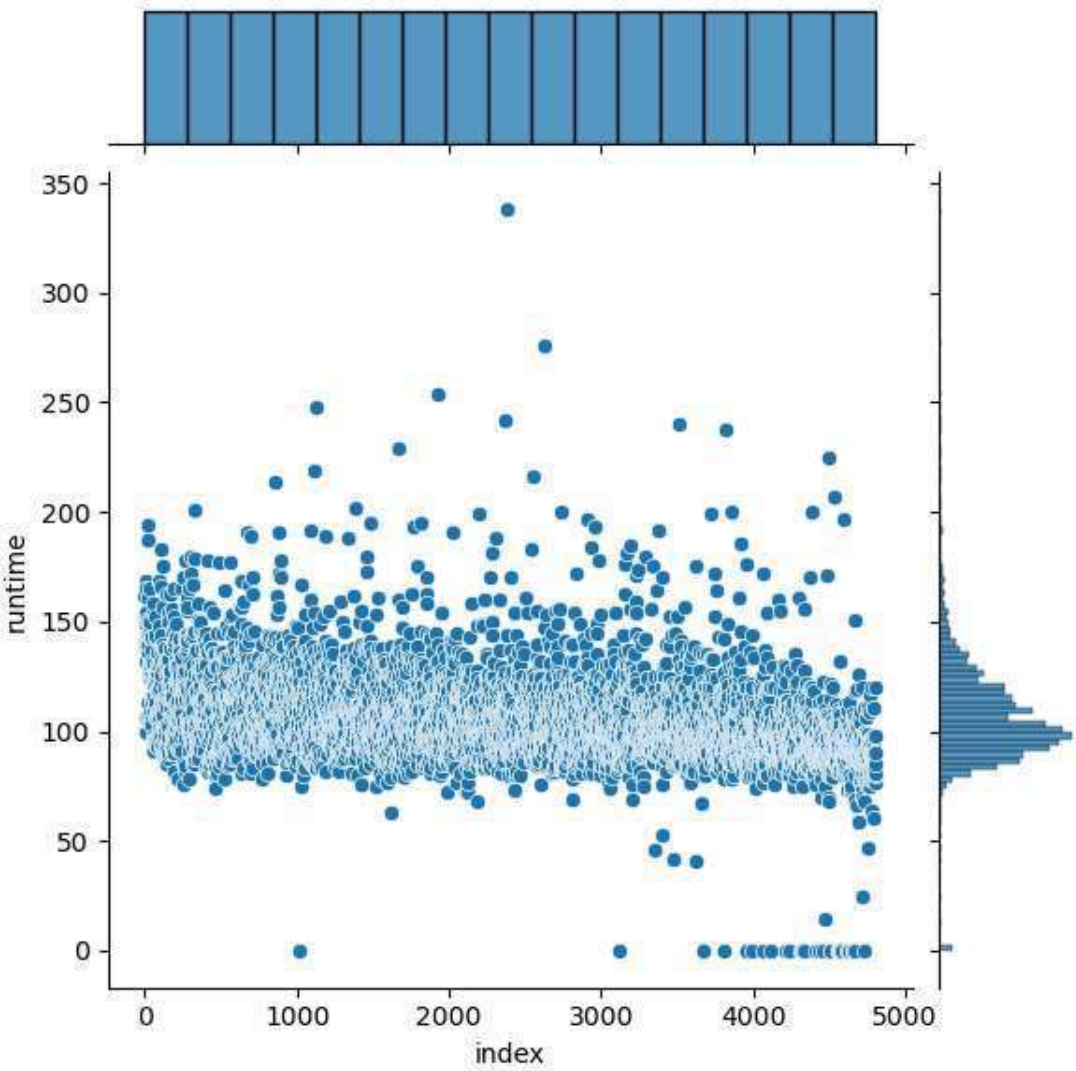
Out[51]:
```
<Axes: ylabel='Density'>
```

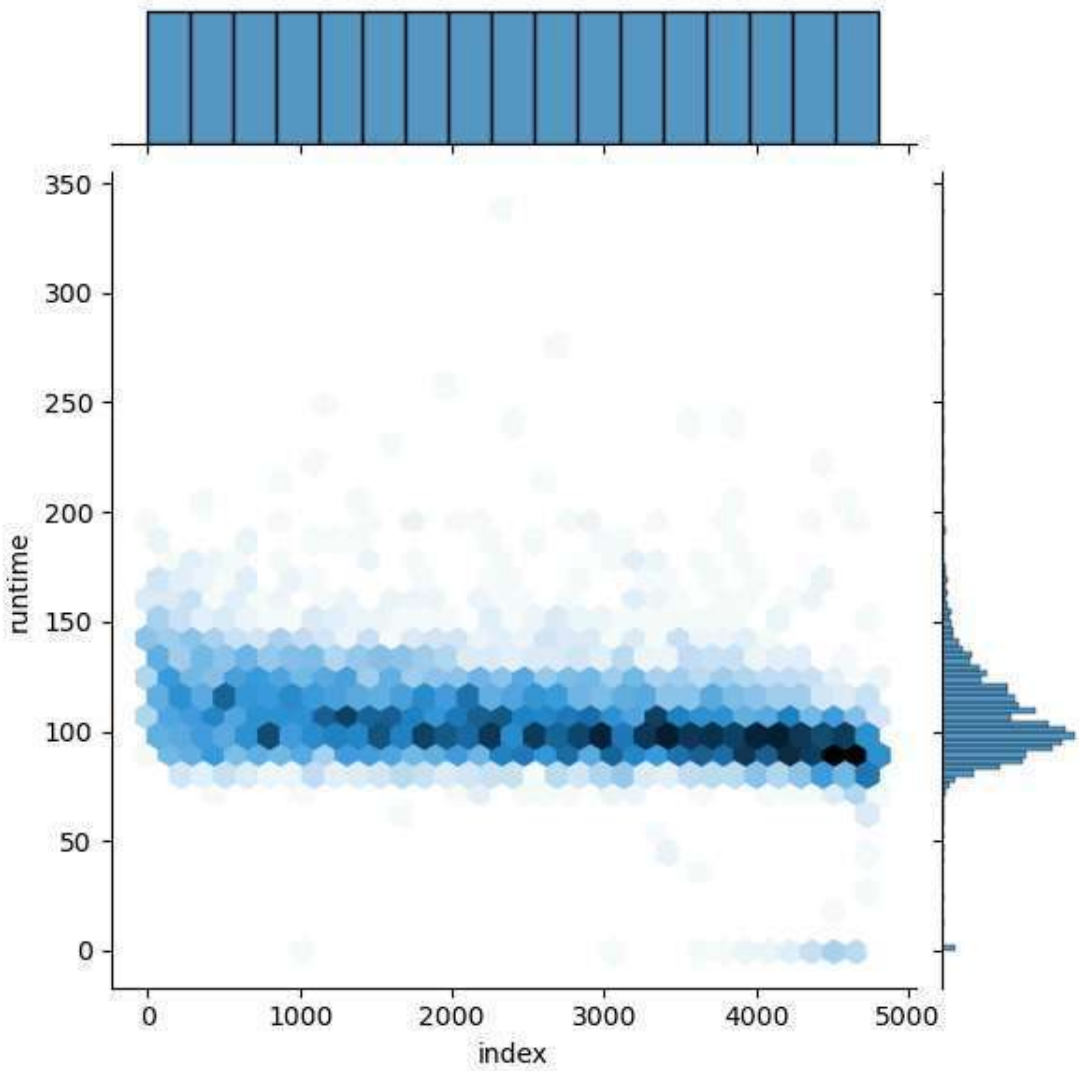


In [52]:
```
sns.jointplot(x = df1['index'], y = df1['runtime'], kind ='scatter')
sns.jointplot(x = df1['index'], y = df1['runtime'], kind = 'hex')
```

```
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[52]:
```
<seaborn.axisgrid.JointGrid at 0x228f93116d0>
```

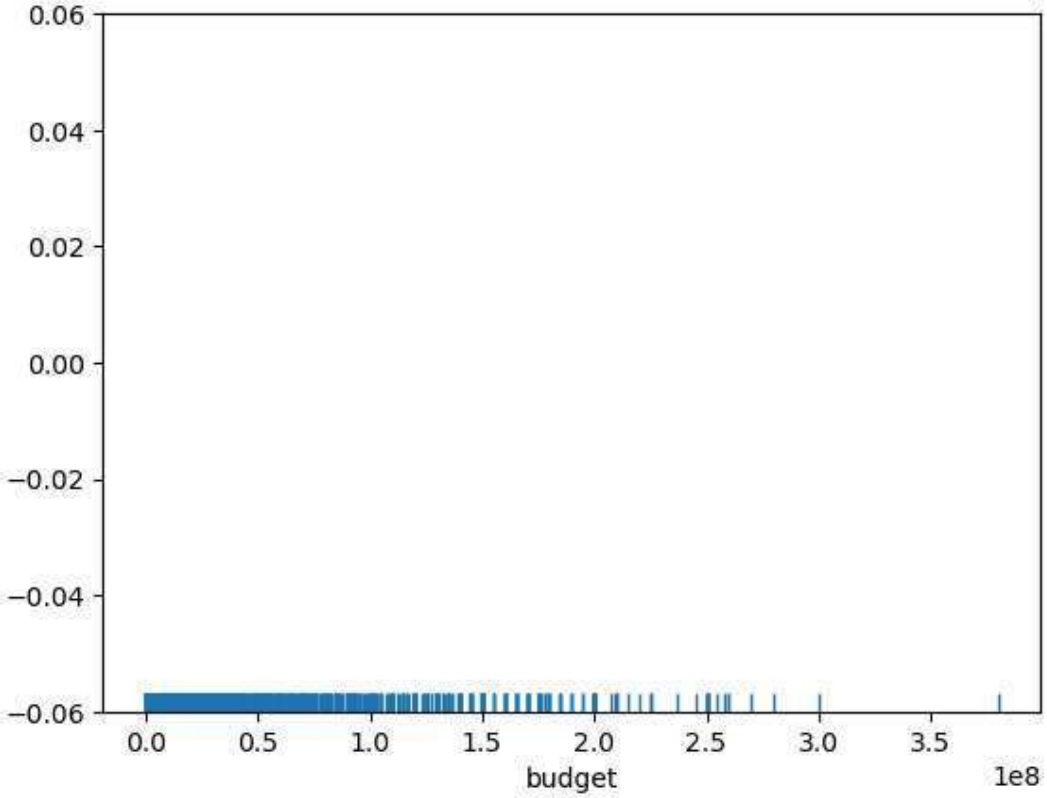

350
300
250
200
150
100
50
0
runtime
0
1000
2000
3000
4000
5000
index



In [53]:
```
sns.rugplot(df1['budget'])
```

```
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

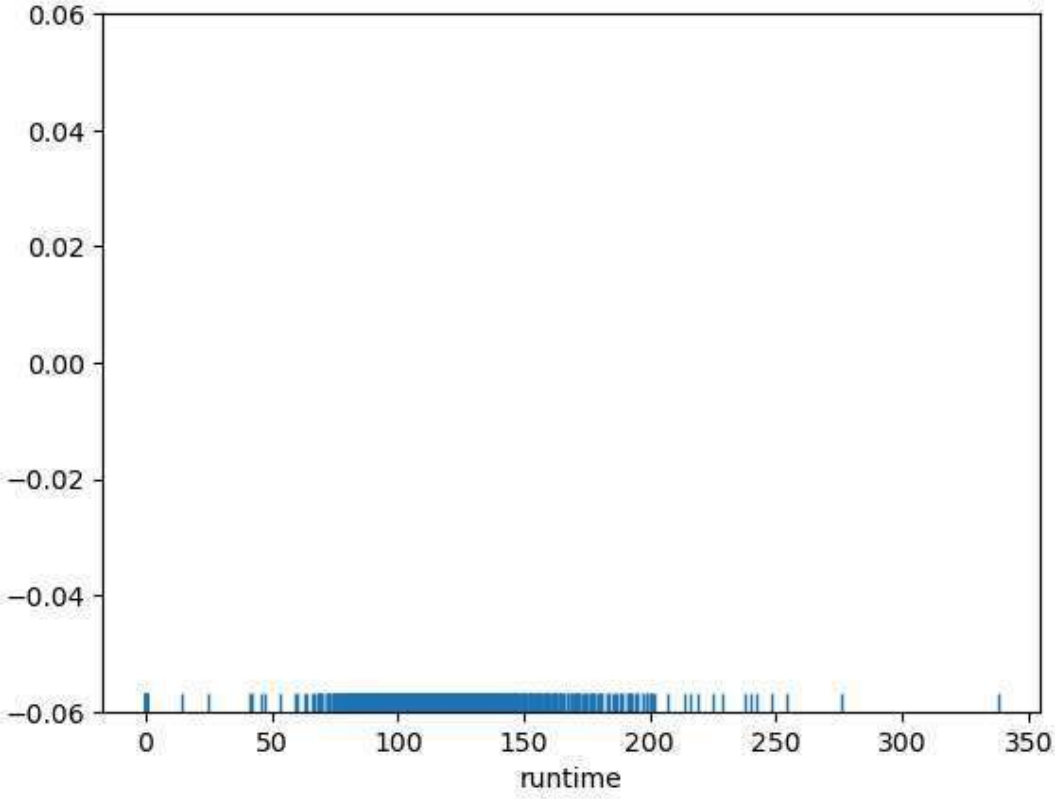
Out[53]:
```
<Axes: xlabel='budget'>
```



In [54]:
```
sns.rugplot(df1['runtime'])
```

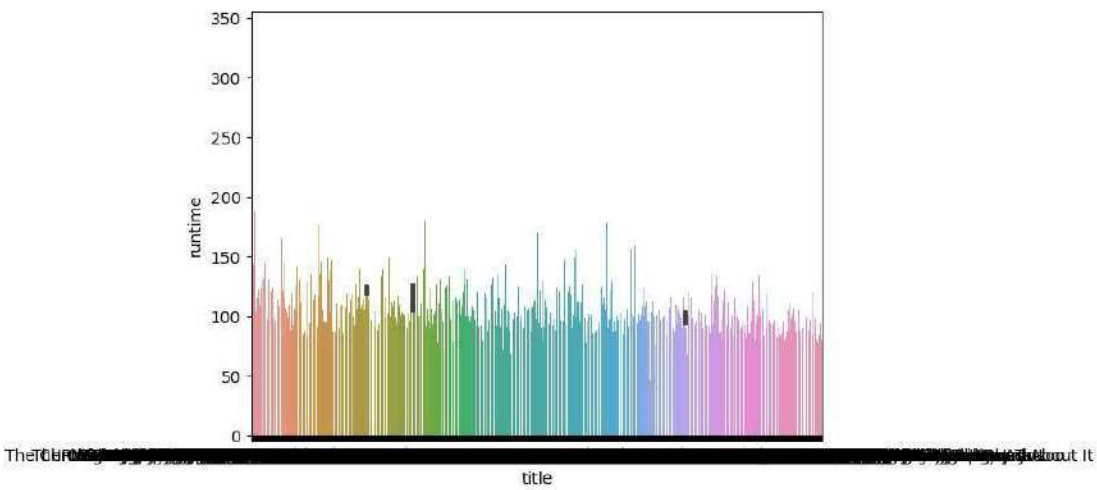
```
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[54]:
```
<Axes: xlabel='runtime'>
```



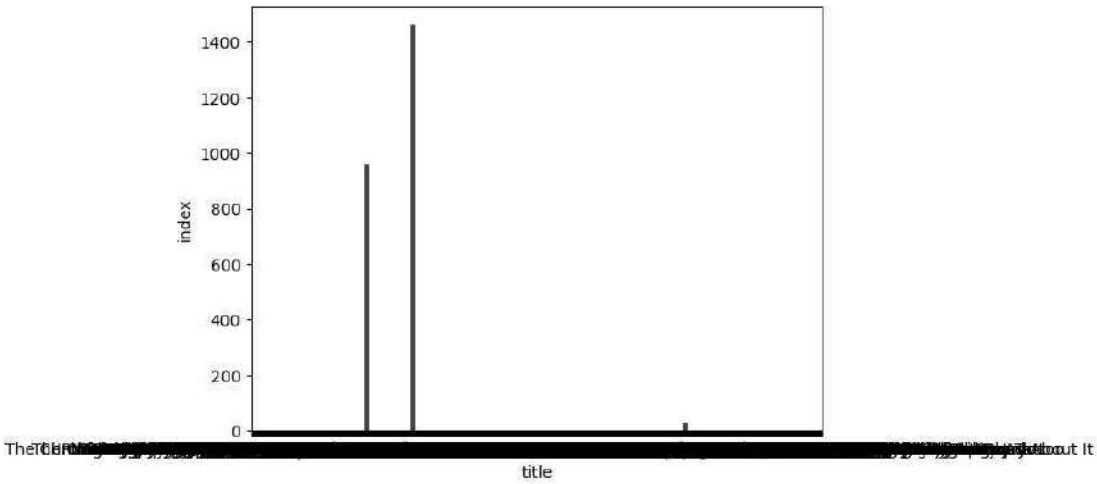
In [14]:
```
sns.barplot(x='title', y='runtime', data=df1)
```

Out[14]: <Axes: xlabel='title', ylabel='runtime'>



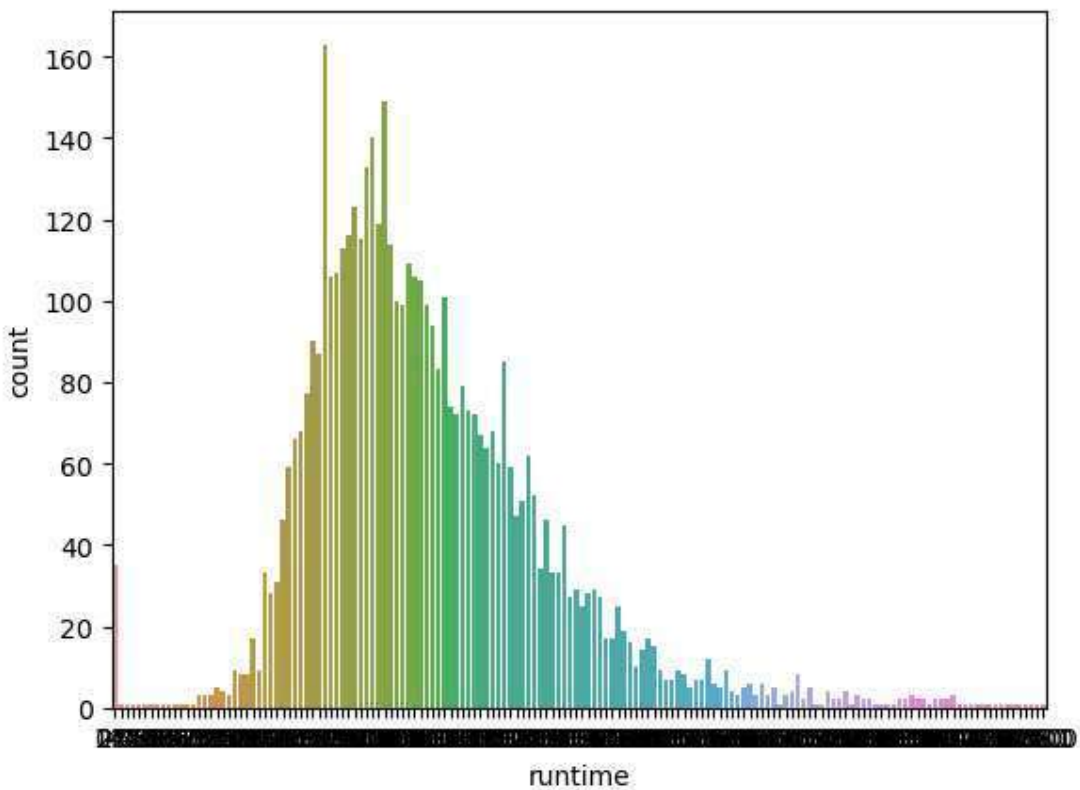
In [8]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.barplot(x='title', y='index', data=df1, estimator=np.std)
```

Out[8]: <Axes: xlabel='title', ylabel='index'>



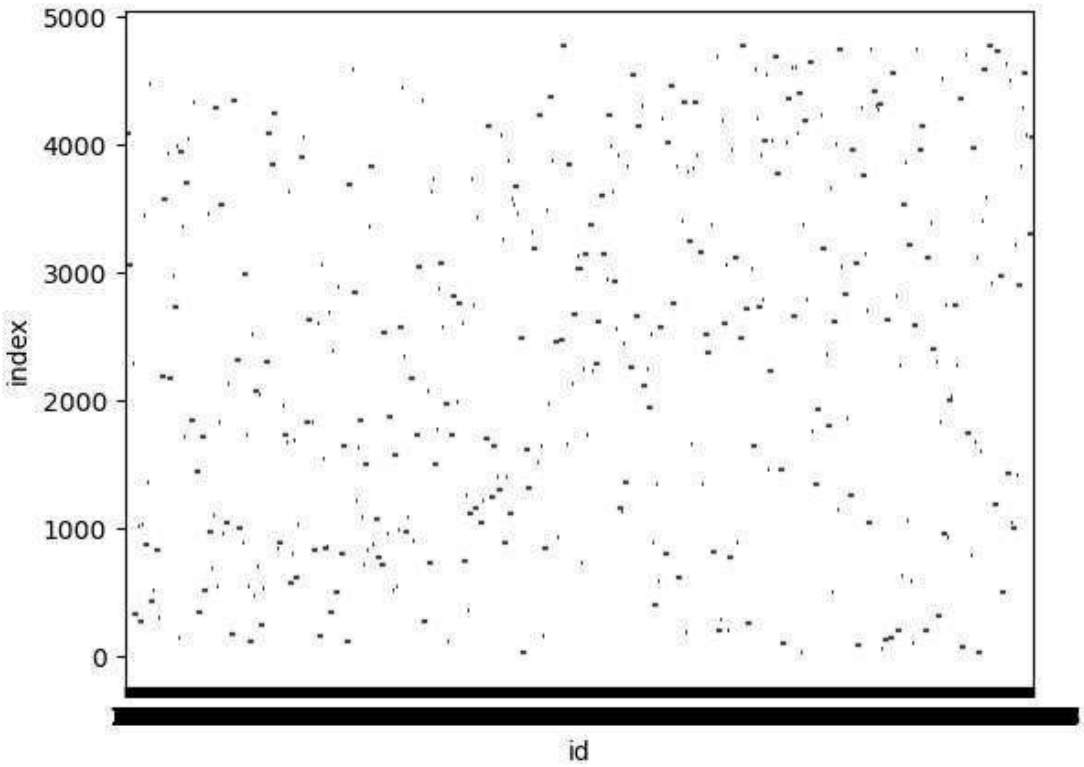
In [12]:
```
sns.countplot(x='runtime', data=df1)
```

Out[12]: <Axes: xlabel='runtime', ylabel='count'>



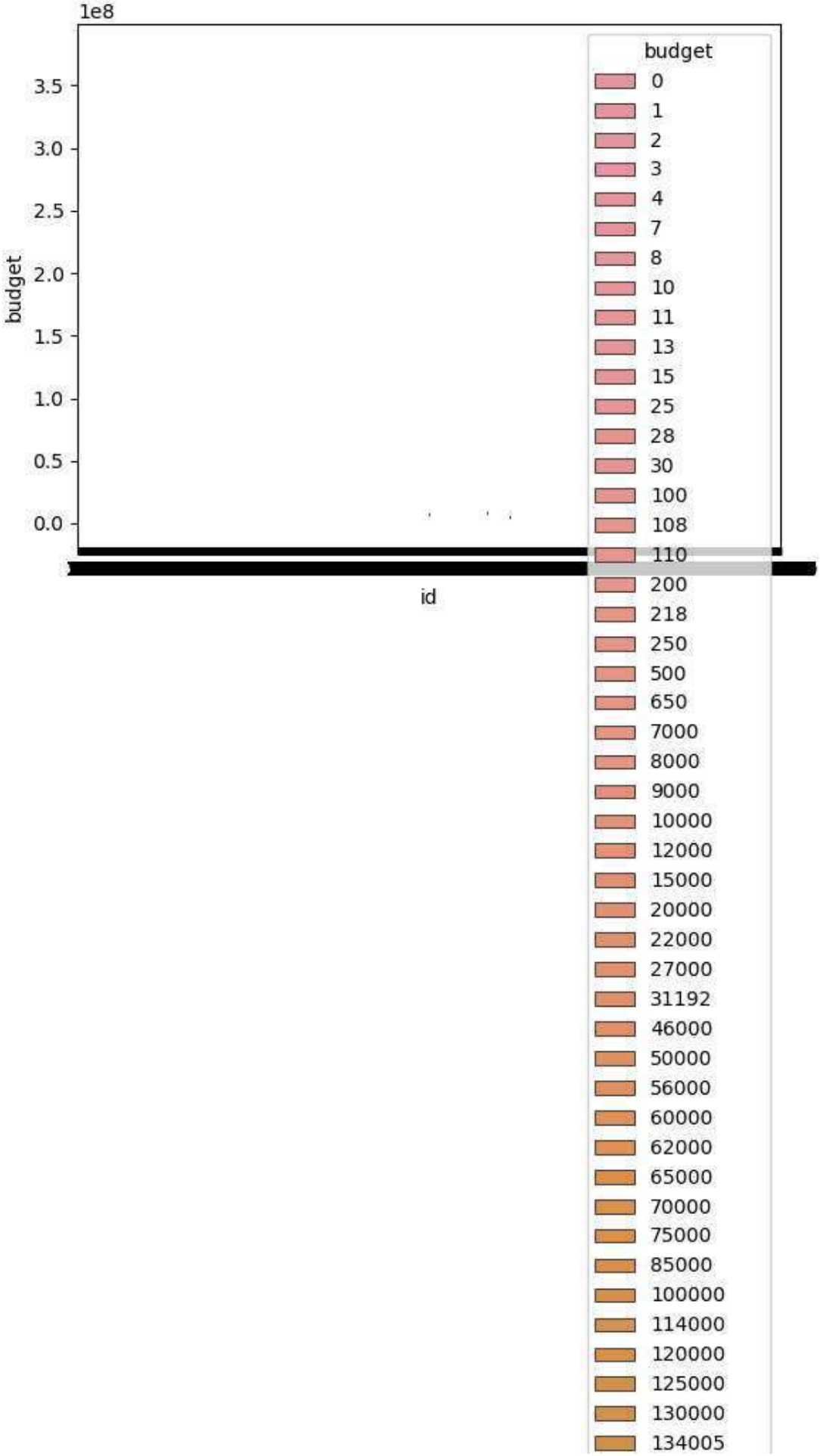
In [13]:
```
sns.boxplot(x='id', y='index', data=df1)
```

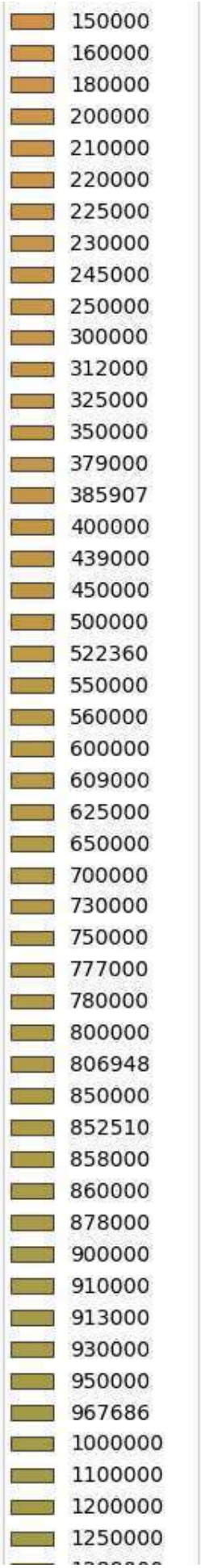
Out[13]: <Axes: xlabel='id', ylabel='index'>

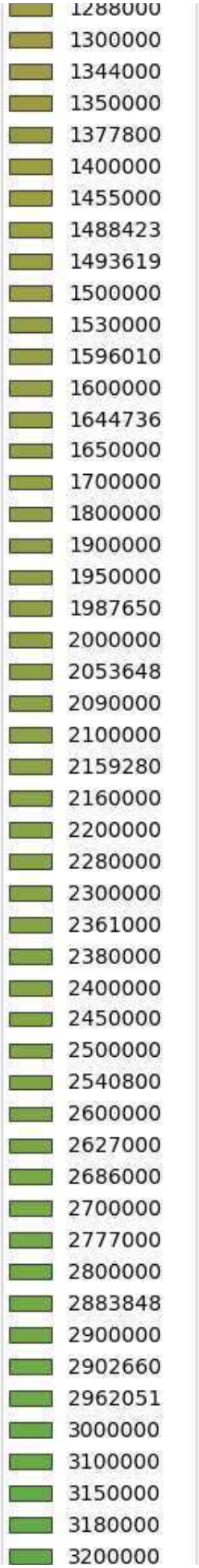


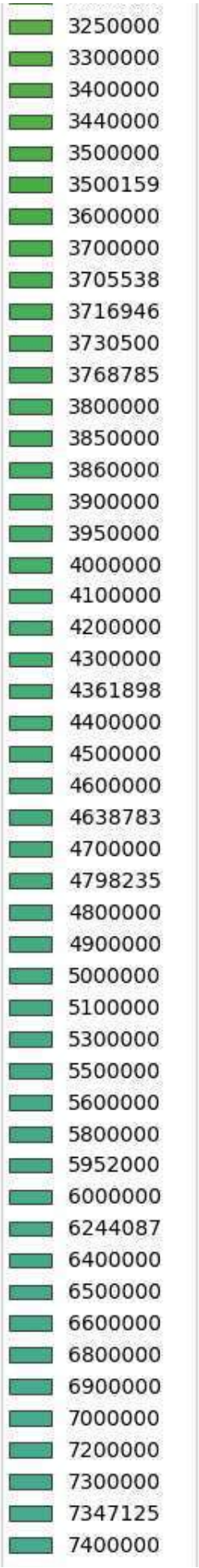
In [18]:
```
sns.boxplot(x='id', y='budget', data=df1, hue="budget")
```

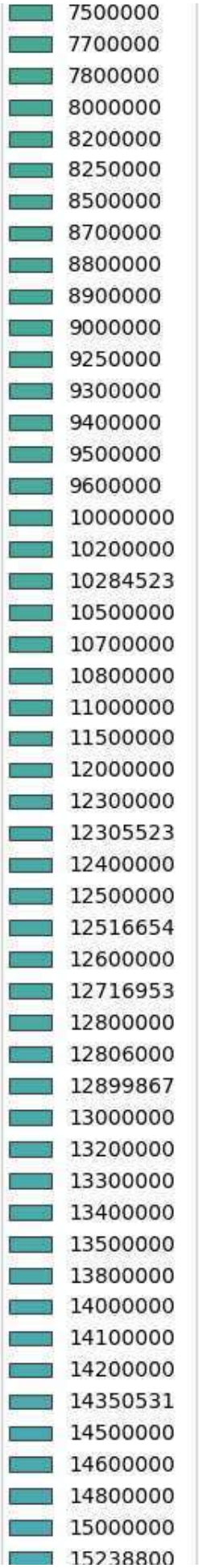
Out[18]:
```
<Axes: xlabel='id', ylabel='budget'>
```

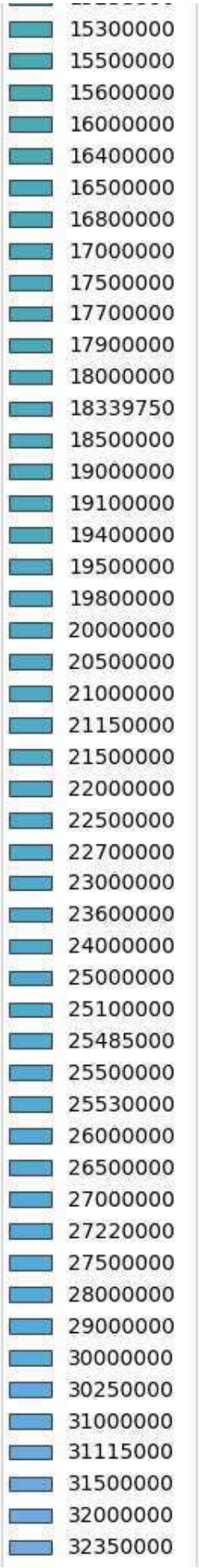

1e8
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
budget
id
budget
0
1
2
3
4
7
8
10
11
13
15
25
28
30
100
108
110
200
218
250
500
650
7000
8000
9000
10000
12000
15000
20000
22000
27000
31192
46000
50000
56000
60000
62000
65000
70000
75000
85000
100000
114000
120000
125000
130000
134005

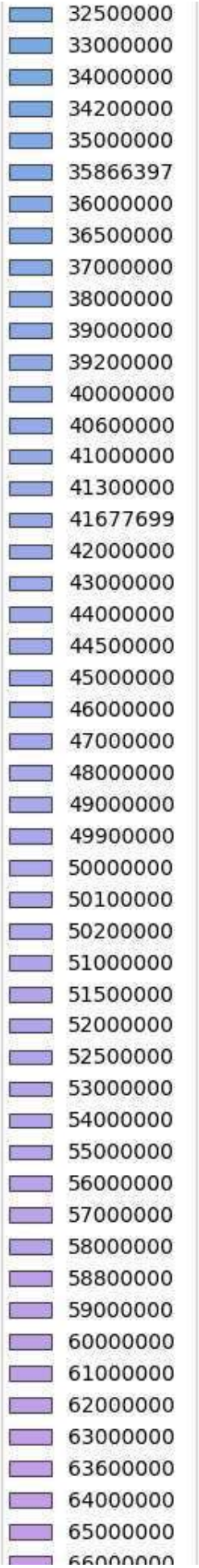

150000
160000
180000
200000
210000
220000
225000
230000
245000
250000
300000
312000
325000
350000
379000
385907
400000
439000
450000
500000
522360
550000
560000
600000
609000
625000
650000
700000
730000
750000
777000
780000
800000
806948
850000
852510
858000
860000
878000
900000
910000
913000
930000
950000
967686
1000000
1100000
1200000
1250000

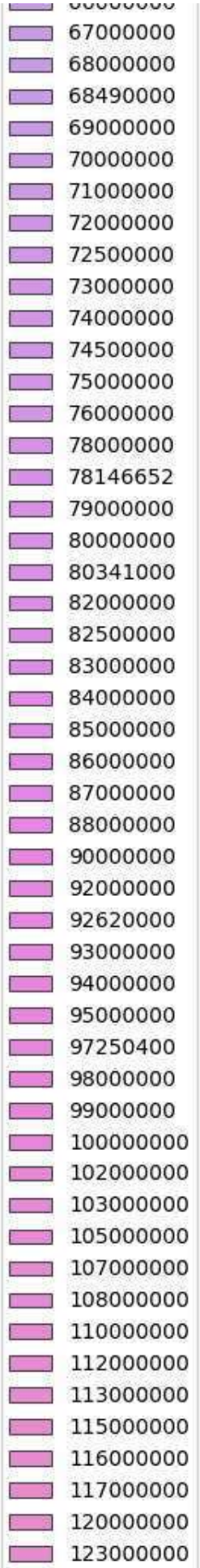

1288000
1300000
1344000
1350000
1377800
1400000
1455000
1488423
1493619
1500000
1530000
1596010
1600000
1644736
1650000
1700000
1800000
1900000
1950000
1987650
2000000
2053648
2090000
2100000
2159280
2160000
2200000
2280000
2300000
2361000
2380000
2400000
2450000
2500000
2540800
2600000
2627000
2686000
2700000
2777000
2800000
2883848
2900000
2902660
2962051
3000000
3100000
3150000
3180000
3200000

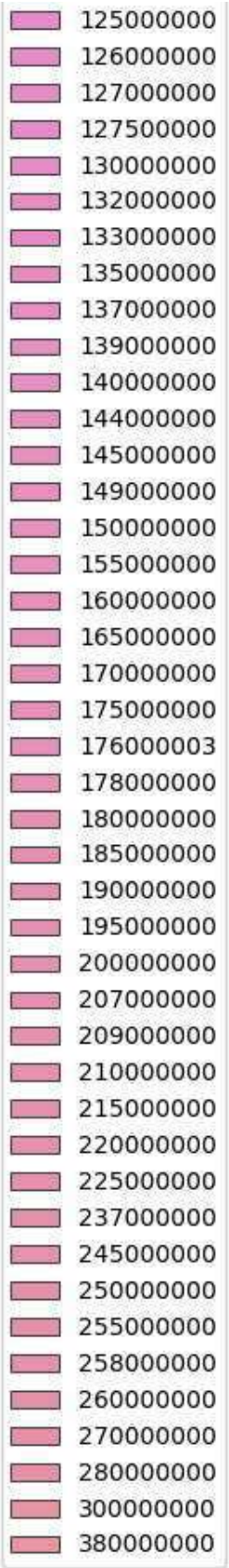

3250000
3300000
3400000
3440000
3500000
3500159
3600000
3700000
3705538
3716946
3730500
3768785
3800000
3850000
3860000
3900000
3950000
4000000
4100000
4200000
4300000
4361898
4400000
4500000
4600000
4638783
4700000
4798235
4800000
4900000
5000000
5100000
5300000
5500000
5600000
5800000
5952000
6000000
6244087
6400000
6500000
6600000
6800000
6900000
7000000
7200000
7300000
7347125
7400000


7500000
7700000
7800000
8000000
8200000
8250000
8500000
8700000
8800000
8900000
9000000
9250000
9300000
9400000
9500000
9600000
10000000
10200000
10284523
10500000
10700000
10800000
11000000
11500000
12000000
12300000
12305523
12400000
12500000
12516654
12600000
12716953
12800000
12806000
12899867
13000000
13200000
13300000
13400000
13500000
13800000
14000000
14100000
14200000
14350531
14500000
14600000
14800000
15000000
15238800


15300000
15500000
15600000
16000000
16400000
16500000
16800000
17000000
17500000
17700000
17900000
18000000
18339750
18500000
19000000
19100000
19400000
19500000
19800000
20000000
20500000
21000000
21150000
21500000
22000000
22500000
22700000
23000000
23600000
24000000
25000000
25100000
25485000
25500000
25530000
26000000
26500000
27000000
27220000
27500000
28000000
29000000
30000000
30250000
31000000
31115000
31500000
32000000
32350000

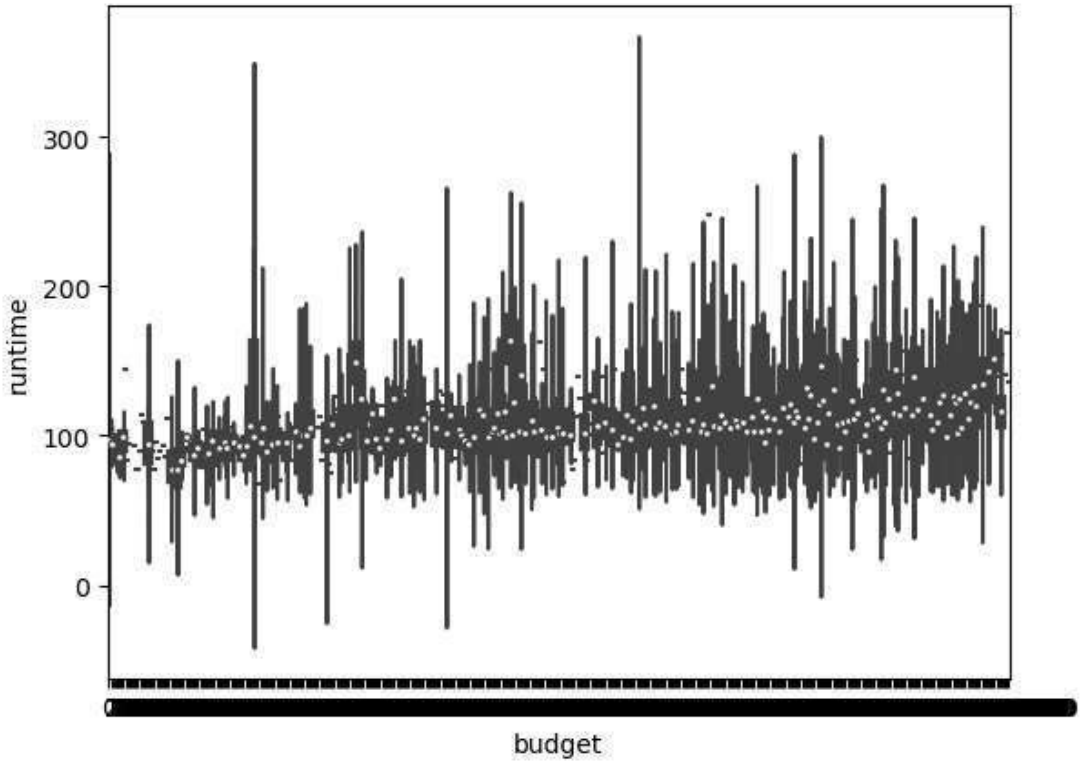

32500000
33000000
34000000
34200000
35000000
35866397
36000000
36500000
37000000
38000000
39000000
39200000
40000000
40600000
41000000
41300000
41677699
42000000
43000000
44000000
44500000
45000000
46000000
47000000
48000000
49000000
49900000
50000000
50100000
50200000
51000000
51500000
52000000
52500000
53000000
54000000
55000000
56000000
57000000
58000000
58800000
59000000
60000000
61000000
62000000
63000000
63600000
64000000
65000000
66000000


67000000
68000000
68490000
69000000
70000000
71000000
72000000
72500000
73000000
74000000
74500000
75000000
76000000
78000000
78146652
79000000
80000000
80341000
82000000
82500000
83000000
84000000
85000000
86000000
87000000
88000000
90000000
92000000
92620000
93000000
94000000
95000000
97250400
98000000
99000000
100000000
102000000
103000000
105000000
107000000
108000000
110000000
112000000
113000000
115000000
116000000
117000000
120000000
123000000



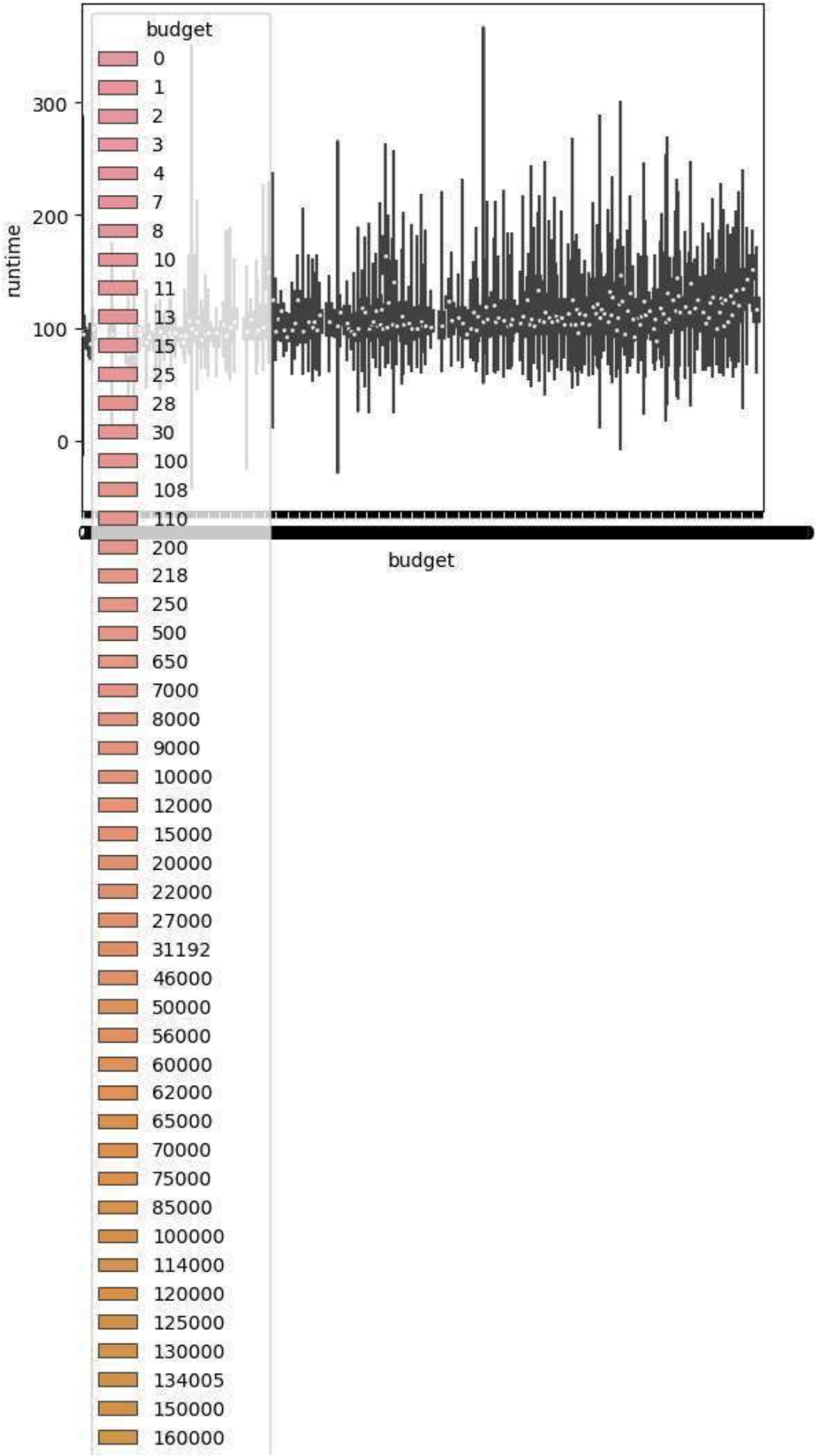
In [17]:
```
sns.violinplot(x='budget', y='runtime', data=df1)
```

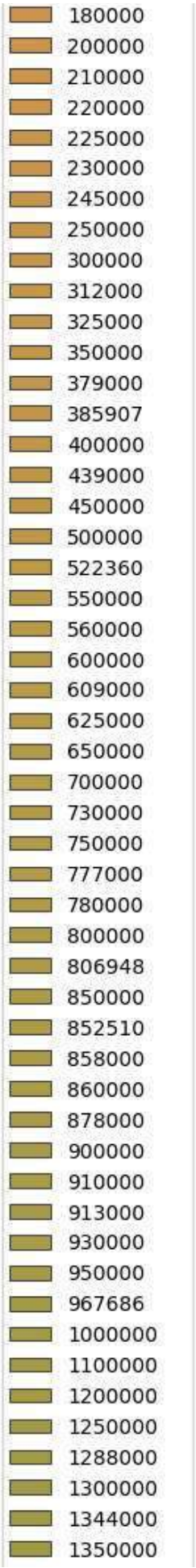
Out[17]:
```
<Axes: xlabel='budget', ylabel='runtime'>
```



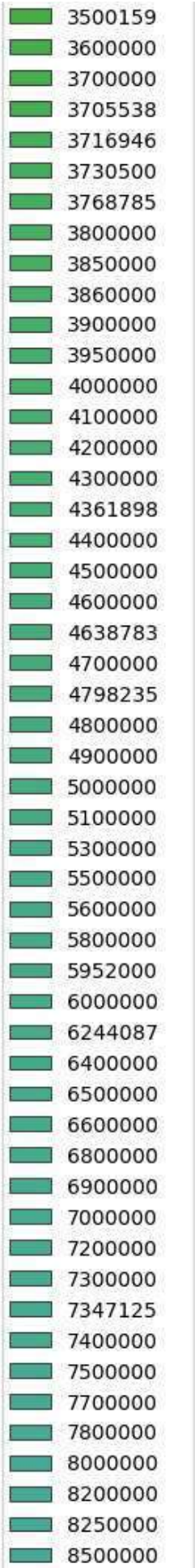
In [19]:
```
sns.violinplot(x='budget', y='runtime', data=df1, hue='budget')
```

Out[19]:
```
<Axes: xlabel='budget', ylabel='runtime'>
```

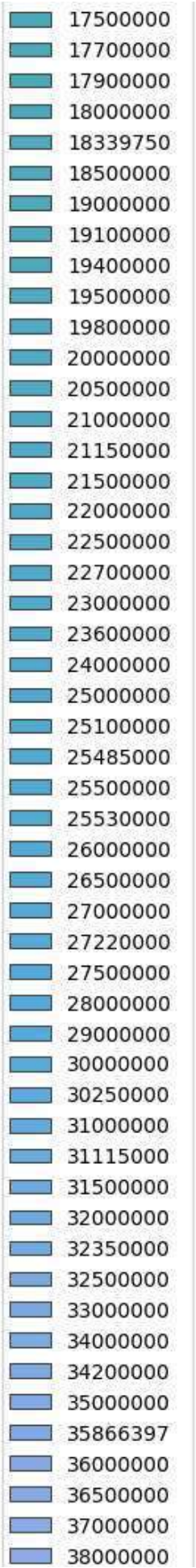

budget
0
1
2
3
4
7
8
10
11
13
15
25
28
30
100
108
110
200
218
250
500
650
7000
8000
9000
10000
12000
15000
20000
22000
27000
31192
46000
50000
56000
60000
62000
65000
70000
75000
85000
100000
114000
120000
125000
130000
134005
150000
160000
runtime
300
200
100
0
budget


180000
200000
210000
220000
225000
230000
245000
250000
300000
312000
325000
350000
379000
385907
400000
439000
450000
500000
522360
550000
560000
600000
609000
625000
650000
700000
730000
750000
777000
780000
800000
806948
850000
852510
858000
860000
878000
900000
910000
913000
930000
950000
967686
1000000
1100000
1200000
1250000
1288000
1300000
1344000
1350000

1377800
1400000
1455000
1488423
1493619
1500000
1530000
1596010
1600000
1644736
1650000
1700000
1800000
1900000
1950000
1987650
2000000
2053648
2090000
2100000
2159280
2160000
2200000
2280000
2300000
2361000
2380000
2400000
2450000
2500000
2540800
2600000
2627000
2686000
2700000
2777000
2800000
2883848
2900000
2902660
2962051
3000000
3100000
3150000
3180000
3200000
3250000
3300000
3400000
3440000
3500000

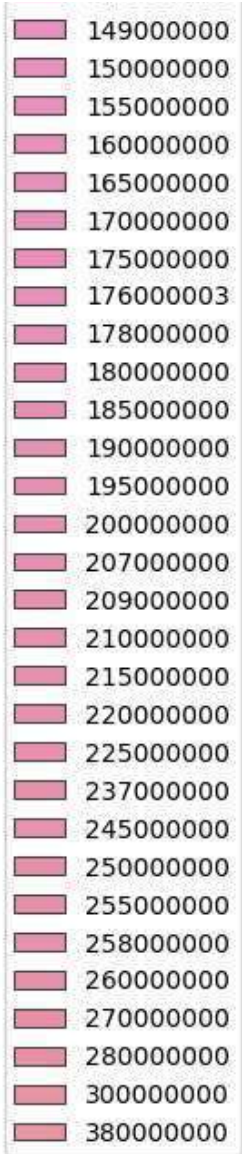

3500159
3600000
3700000
3705538
3716946
3730500
3768785
3800000
3850000
3860000
3900000
3950000
4000000
4100000
4200000
4300000
4361898
4400000
4500000
4600000
4638783
4700000
4798235
4800000
4900000
5000000
5100000
5300000
5500000
5600000
5800000
5952000
6000000
6244087
6400000
6500000
6600000
6800000
6900000
7000000
7200000
7300000
7347125
7400000
7500000
7700000
7800000
8000000
8200000
8250000
8500000

8700000
8800000
8900000
9000000
9250000
9300000
9400000
9500000
9600000
10000000
10200000
10284523
10500000
10700000
10800000
11000000
11500000
12000000
12300000
12305523
12400000
12500000
12516654
12600000
12716953
12800000
12806000
12899867
13000000
13200000
13300000
13400000
13500000
13800000
14000000
14100000
14200000
14350531
14500000
14600000
14800000
15000000
15238800
15300000
15500000
15600000
16000000
16400000
16500000
16800000
17000000


17500000
17700000
17900000
18000000
18339750
18500000
19000000
19100000
19400000
19500000
19800000
20000000
20500000
21000000
21150000
21500000
22000000
22500000
22700000
23000000
23600000
24000000
25000000
25100000
25485000
25500000
25530000
26000000
26500000
27000000
27220000
27500000
28000000
29000000
30000000
30250000
31000000
31115000
31500000
32000000
32350000
32500000
33000000
34000000
34200000
35000000
35866397
36000000
36500000
37000000
38000000

39000000
39200000
40000000
40600000
41000000
41300000
41677699
42000000
43000000
44000000
44500000
45000000
46000000
47000000
48000000
49000000
49900000
50000000
50100000
50200000
51000000
51500000
52000000
52500000
53000000
54000000
55000000
56000000
57000000
58000000
58800000
59000000
60000000
61000000
62000000
63000000
63600000
64000000
65000000
66000000
67000000
68000000
68490000
69000000
70000000
71000000
72000000
72500000
73000000
74000000
74500000

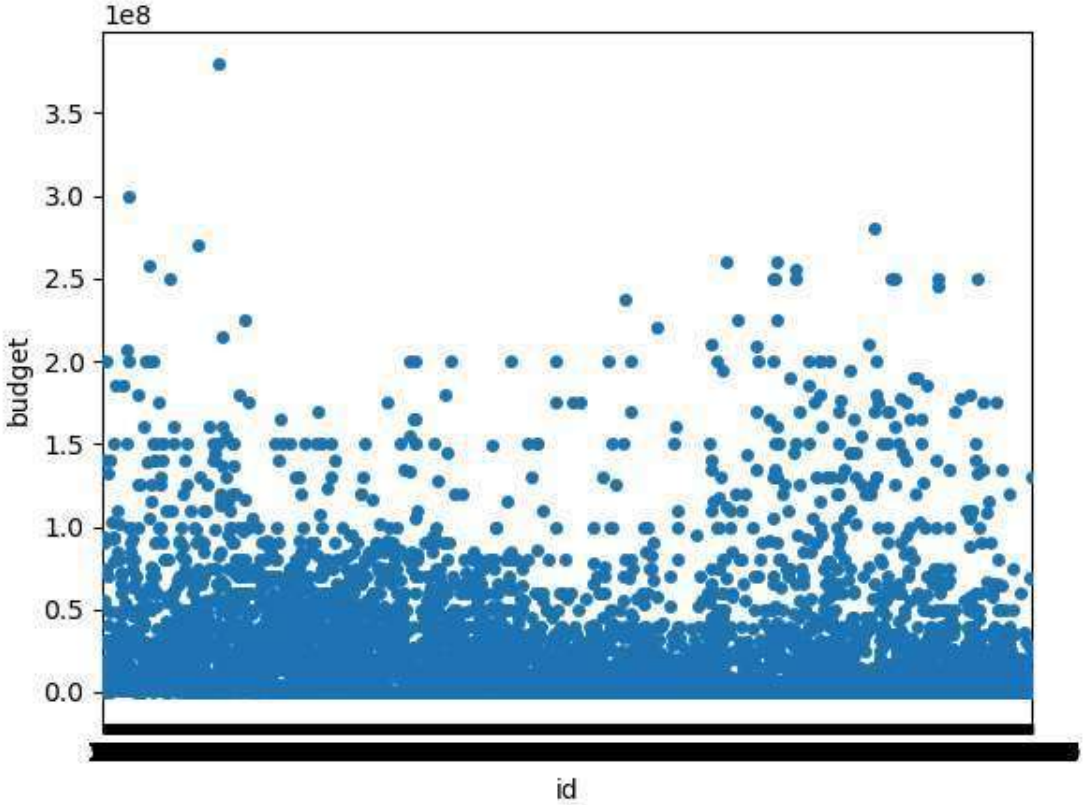
75000000
76000000
78000000
78146652
79000000
80000000
80341000
82000000
82500000
83000000
84000000
85000000
86000000
87000000
88000000
90000000
92000000
92620000
93000000
94000000
95000000
97250400
98000000
99000000
100000000
102000000
103000000
105000000
107000000
108000000
110000000
112000000
113000000
115000000
116000000
117000000
120000000
123000000
125000000
126000000
127000000
127500000
130000000
132000000
133000000
135000000
137000000
139000000
140000000
144000000
145000000



In [20]:
```
sns.stripplot(x='id', y='budget', data=df1, jitter=False)
```

```
C:\Users\System21\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWar
ning: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\System21\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWar
ning: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

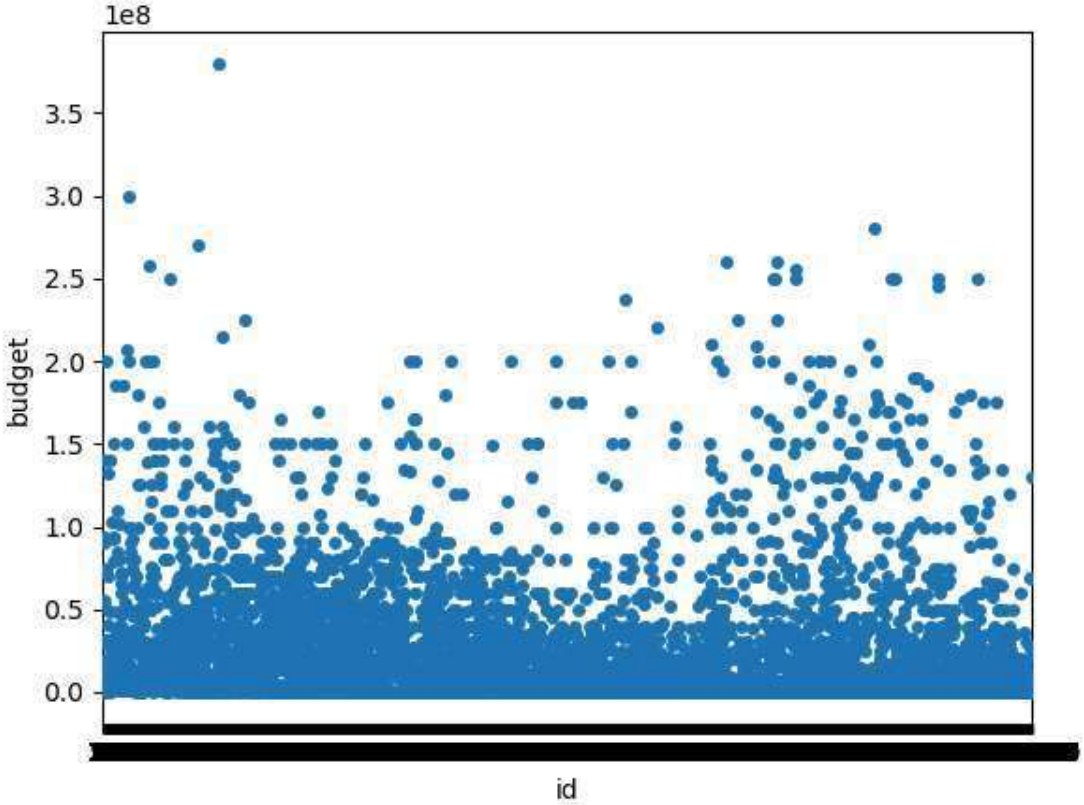
Out[20]:
```
<Axes: xlabel='id', ylabel='budget'>
```



In [21]:
```
sns.stripplot(x='id', y='budget', data=df1, jitter=True)
```

```
C:\Users\System21\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWar
ning: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\System21\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWar
ning: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

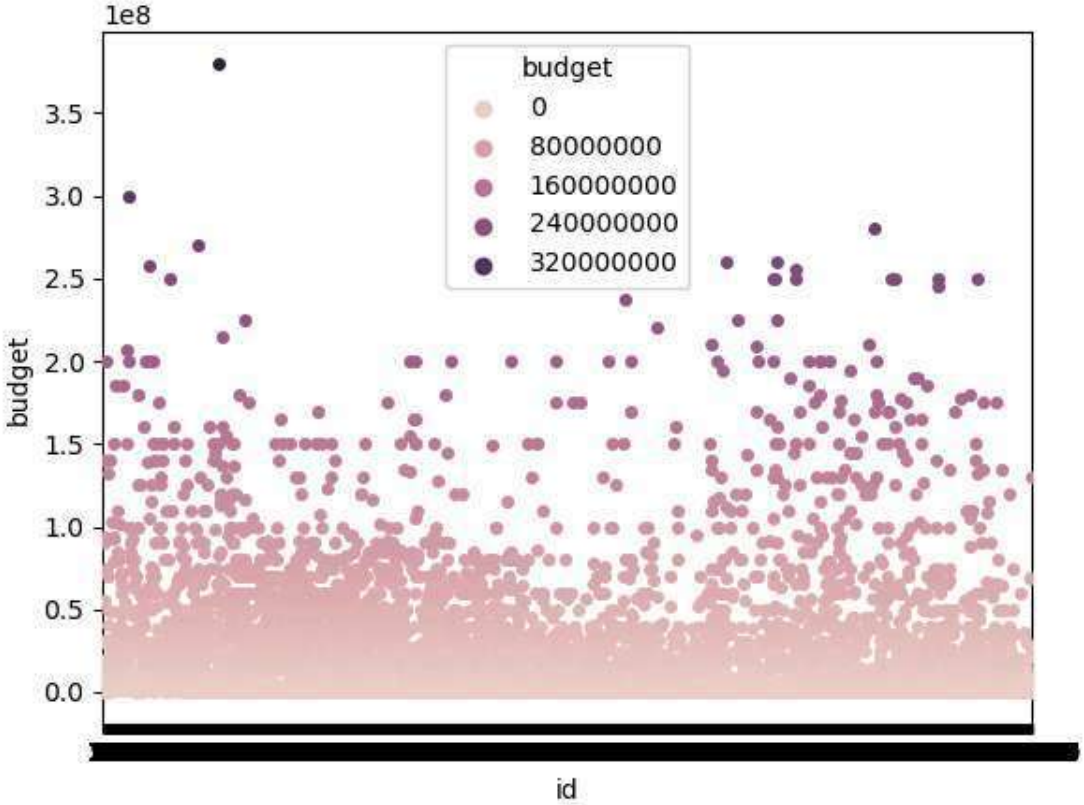
Out[21]:
```
<Axes: xlabel='id', ylabel='budget'>
```



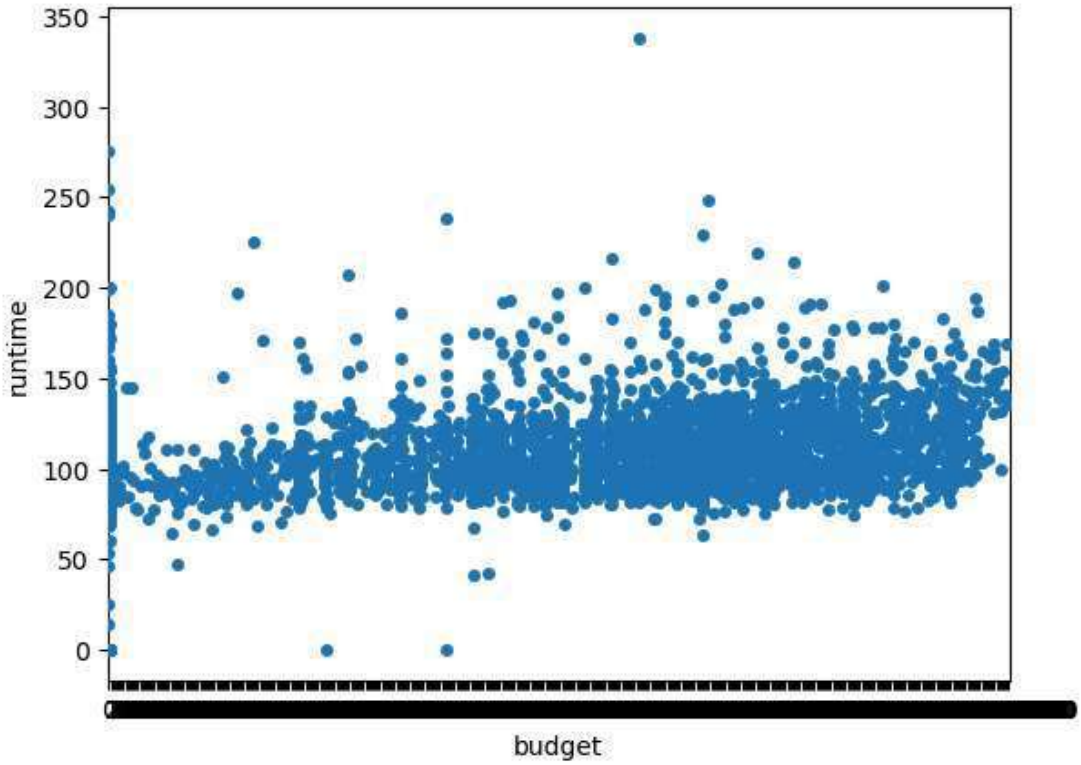
In [22]:
```
sns.stripplot(x='id', y='budget', data=df1, jitter=False, hue='budget')
```

```
C:\Users\System21\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWar
ning: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\System21\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWar
ning: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[22]:
```
<Axes: xlabel='id', ylabel='budget'>
```



In [23]:
```
sns.swarmplot(x='budget', y='runtime', data=df1)
```



In [25]:
```
numerical_df = df1.select_dtypes(include=['number'])
```

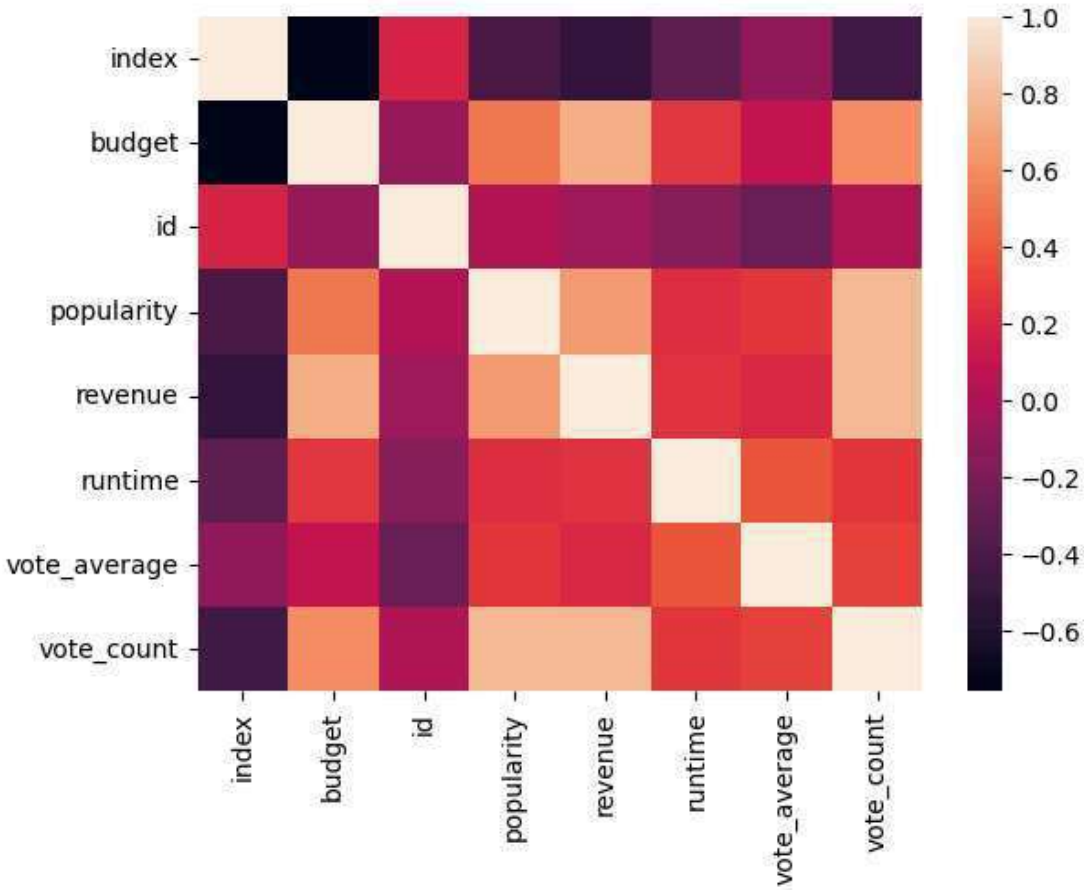
In [26]:
```
corr = numerical_df.corr()
corr
```

Out[26]:

	index	budget	id	popularity	revenue	runtime	vote_ave
index	1.000000	-0.761579	0.190771	-0.414342	-0.522110	-0.319370	-0.12
budget	-0.761579	1.000000	-0.089377	0.505414	0.730823	0.269851	0.09
id	0.190771	-0.089377	1.000000	0.031202	-0.050425	-0.153536	-0.27
popularity	-0.414342	0.505414	0.031202	1.000000	0.644724	0.225502	0.27
revenue	-0.522110	0.730823	-0.050425	0.644724	1.000000	0.251093	0.19
runtime	-0.319370	0.269851	-0.153536	0.225502	0.251093	1.000000	0.37
vote_average	-0.120157	0.093146	-0.270595	0.273952	0.197150	0.375046	1.00
vote_count	-0.442207	0.593180	-0.004128	0.778130	0.781487	0.271944	0.31

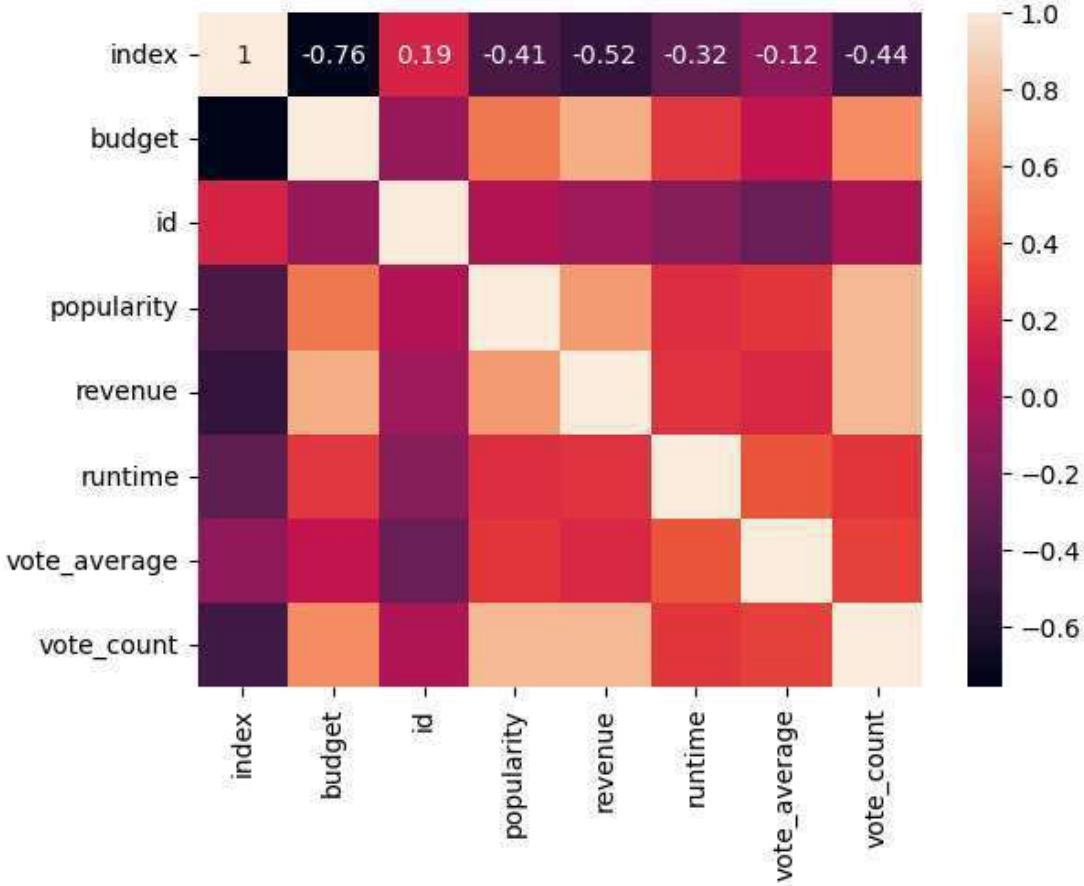
In [27]:
```
sns.heatmap(corr)
```

Out[27]: <Axes: >



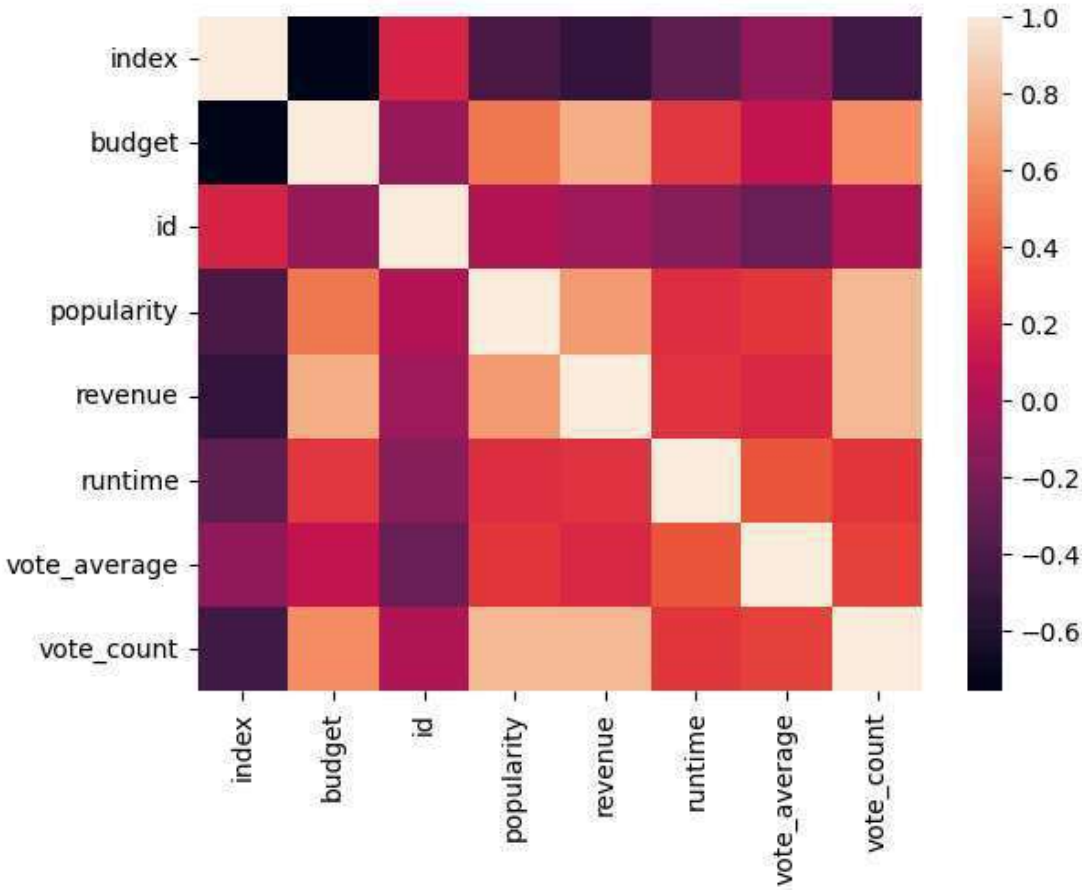
In [28]:
```
sns.heatmap(corr, annot=True)
```

Out[28]: <Axes: >



In [29]:
```
sns.heatmap(corr)
```

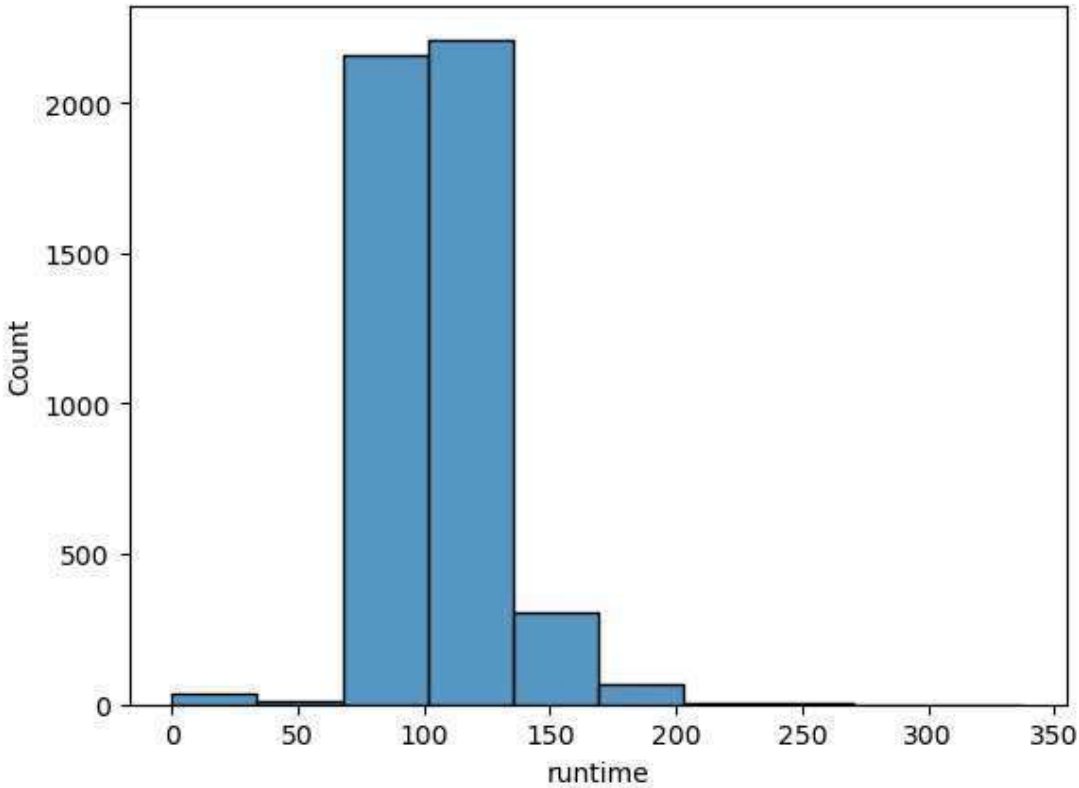
Out[29]: <Axes: >



In [30]:
```
sns.histplot(df1['runtime'], kde=False, bins=10)
```

```
C:\Users\System21\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWar
ning: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[30]:
```
<Axes: xlabel='runtime', ylabel='Count'>
```

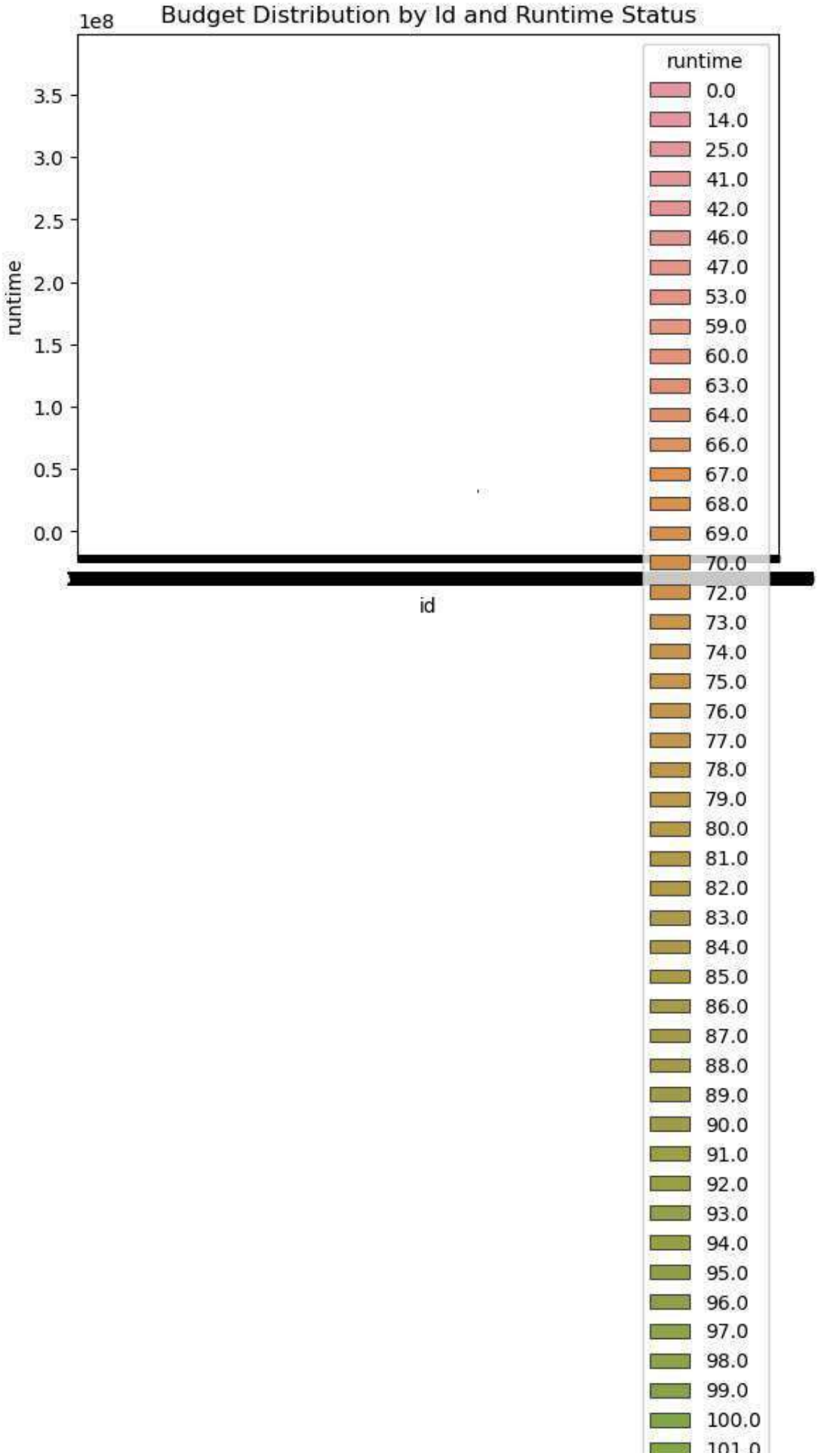


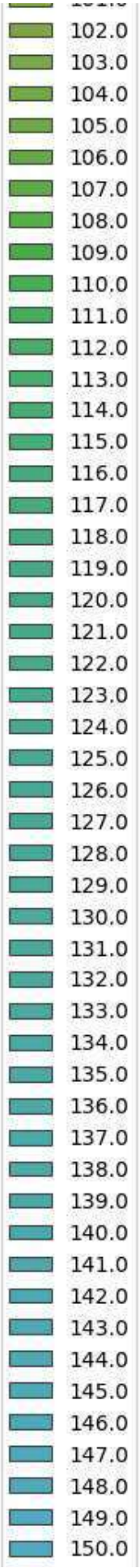
In [31]:
```
plt.figure(figsize=(10, 6))
```

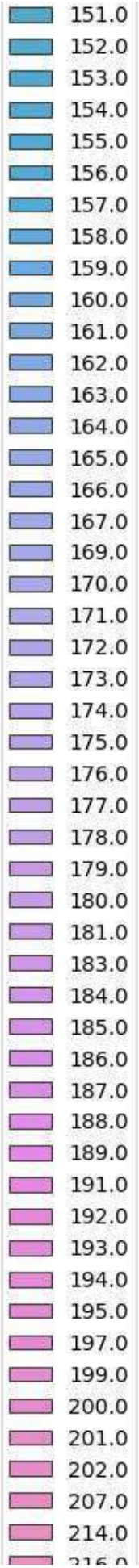
Out[31]:
```
<Figure size 1000x600 with 0 Axes>
```

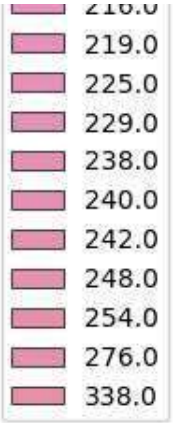

```
<Figure size 1000x600 with 0 Axes>
```


In [32]:
```
sns.boxplot(x='id', y='budget', hue='runtime', data=df1)
plt.title('Budget Distribution by Id and Runtime Status')
plt.xlabel('id')
plt.ylabel('runtime')
plt.show()
```

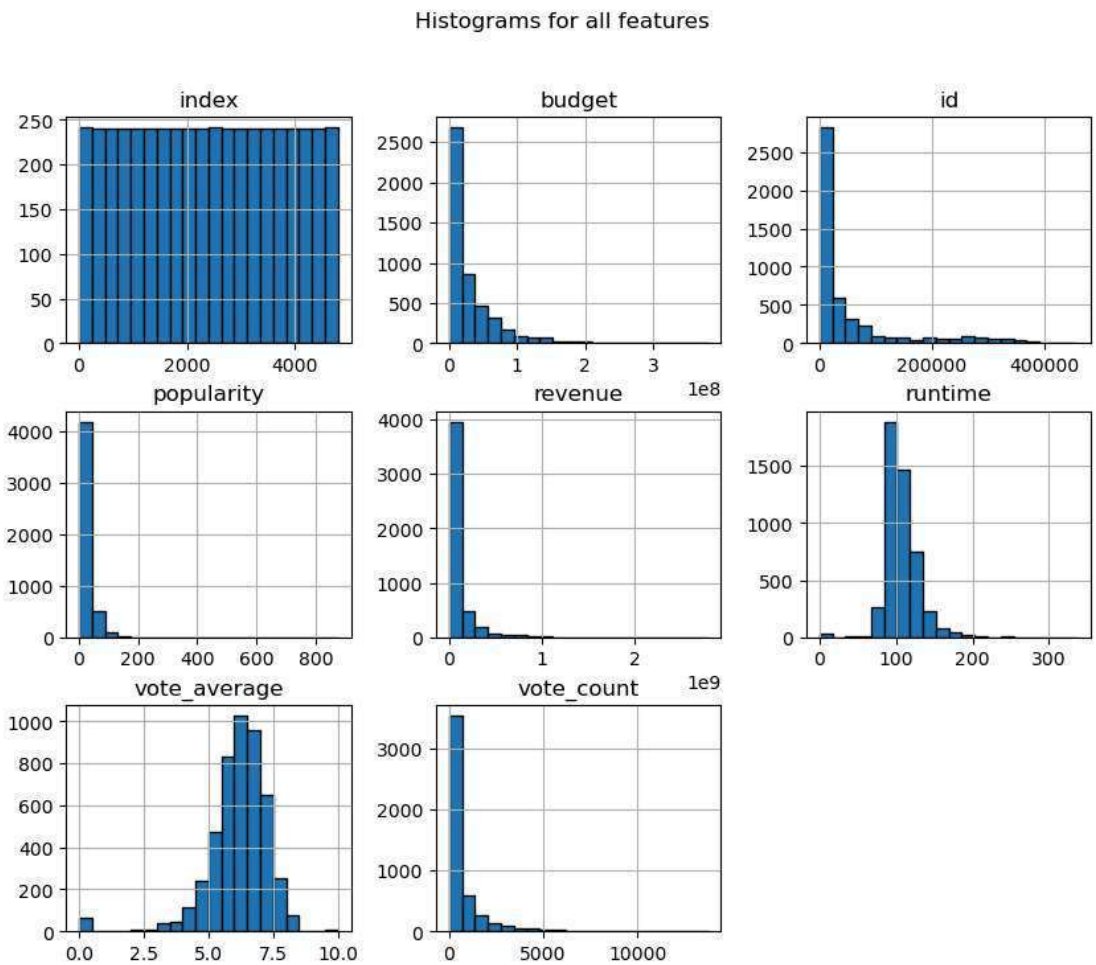

Budget Distribution by Id and Runtime Status
1e8
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
runtime
id
runtime
0.0
14.0
25.0
41.0
42.0
46.0
47.0
53.0
59.0
60.0
63.0
64.0
66.0
67.0
68.0
69.0
70.0
72.0
73.0
74.0
75.0
76.0
77.0
78.0
79.0
80.0
81.0
82.0
83.0
84.0
85.0
86.0
87.0
88.0
89.0
90.0
91.0
92.0
93.0
94.0
95.0
96.0
97.0
98.0
99.0
100.0
101.0


102.0
103.0
104.0
105.0
106.0
107.0
108.0
109.0
110.0
111.0
112.0
113.0
114.0
115.0
116.0
117.0
118.0
119.0
120.0
121.0
122.0
123.0
124.0
125.0
126.0
127.0
128.0
129.0
130.0
131.0
132.0
133.0
134.0
135.0
136.0
137.0
138.0
139.0
140.0
141.0
142.0
143.0
144.0
145.0
146.0
147.0
148.0
149.0
150.0


151.0
152.0
153.0
154.0
155.0
156.0
157.0
158.0
159.0
160.0
161.0
162.0
163.0
164.0
165.0
166.0
167.0
169.0
170.0
171.0
172.0
173.0
174.0
175.0
176.0
177.0
178.0
179.0
180.0
181.0
183.0
184.0
185.0
186.0
187.0
188.0
189.0
191.0
192.0
193.0
194.0
195.0
197.0
199.0
200.0
201.0
202.0
207.0
214.0



In [61]:
```
df1.drop('genres', axis=1).hist(figsize=(10, 8), bins=20, edgecolor='black')
plt.suptitle('Histograms for all features')
plt.show()
```



Name : Mohan Kadambande

Roll No : TECO-B1 (13212)