

**Prof. Ram Meghe Institute of Technology & Research Badnera**  
**Department Of Computer Application**  
**MCA 2<sup>nd</sup> Year Sem-III**

**Subject :** Data Analytics Practical

**Instructed By :** Prof. Rupali Sherekar Mam

**Submitted By :** Miss Surbhi P. Rahangdale

**Google Colab Link** of performed Practicals :

<https://colab.research.google.com/drive/1g3NmnZZ7Oaw1NxmQms4lcKEsGnrG91C?usp=sharing>

---

### **Practical 1**

**Name:** Tasks Based on basic Python Constructs

**Objective:** Understanding of Python as a language

**Tasks:**

- 1.1 Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2700 (both included)
- 1.2 Write a Python program to count the number of even and odd numbers from a series of numbers.
- 1.3 Write a Python program that accepts a word from the user and reverse it.
- 1.4 Write a Python program that displays a list in sorted order after eliminating the duplicates in the list
- 1.5 Write a Python program that counts the even and odd numbers in a list

## Practical 1.1

### Code:

```
for i in range(1500,2701):
```

```
    if i%7==0 or i%5==0:
```

```
        print(i,end=",")
```

### Output :



```
[3] #Practical 1.1
    for i in range(1500,2701):
        if i%7==0 or i%5==0:
            print(i,end=",")

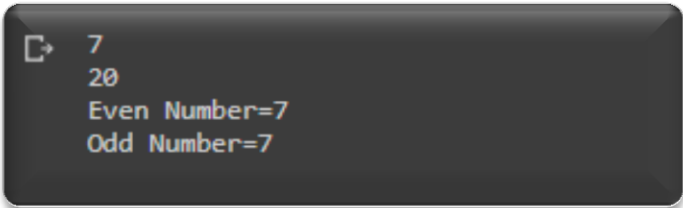
1500,1505,1510,1512,1515,1519,1520,1525,1526,1530,1533,1535,1540,1545,1547,1550,1554,1555,1560,1561,1565,1568,1570,1575,1580,1582,1585,1589,1590,1595,1598,1600,1605,1610,1612,1615,1619,1620,1625,1626,1630,1633,1635,1640,1645,1647,1650,1654,1655,1660,1661,1665,1668,1670,1675,1680,1682,1685,1689,1690,1695,1698,1700,1705,1710,1712,1715,1719,1720,1725,1726,1730,1733,1735,1740,1745,1747,1750,1754,1755,1760,1761,1765,1768,1770,1775,1780,1782,1785,1789,1790,1795,1798,1800,1805,1810,1812,1815,1819,1820,1825,1826,1830,1833,1835,1840,1845,1847,1850,1854,1855,1860,1861,1865,1868,1870,1875,1880,1882,1885,1889,1890,1895,1898,1900,1905,1910,1912,1915,1919,1920,1925,1926,1930,1933,1935,1940,1945,1947,1950,1954,1955,1960,1961,1965,1968,1970,1975,1980,1982,1985,1989,1990,1995,1998,2000,2005,2010,2012,2015,2019,2020,2025,2026,2030,2033,2035,2040,2045,2047,2050,2054,2055,2060,2061,2065,2068,2070,2075,2080,2082,2085,2089,2090,2095,2098,2100,2105,2110,2112,2115,2119,2120,2125,2126,2130,2133,2135,2140,2145,2147,2150,2154,2155,2160,2161,2165,2168,2170,2175,2180,2182,2185,2189,2190,2195,2198,2200,2205,2210,2212,2215,2219,2220,2225,2226,2230,2233,2235,2240,2245,2247,2250,2254,2255,2260,2261,2265,2268,2270,2275,2280,2282,2285,2289,2290,2295,2298,2300,2305,2310,2312,2315,2319,2320,2325,2326,2330,2333,2335,2340,2345,2347,2350,2354,2355,2360,2361,2365,2368,2370,2375,2380,2382,2385,2389,2390,2395,2398,2400,2405,2410,2412,2415,2419,2420,2425,2426,2430,2433,2435,2440,2445,2447,2450,2454,2455,2460,2461,2465,2468,2470,2475,2480,2482,2485,2489,2490,2495,2498,2500,2505,2510,2512,2515,2519,2520,2525,2526,2530,2533,2535,2540,2545,2547,2550,2554,2555,2560,2561,2565,2568,2570,2575,2580,2582,2585,2589,2590,2595,2598,2600,2605,2610,2612,2615,2619,2620,2625,2626,2630,2633,2635,2640,2645,2647,2650,2654,2655,2660,2661,2665,2668,2670,2675,2680,2682,2685,2689,2690,2695,2698,2700,2701
```

### Practical 1.2

#### Code:

```
strt=int(input())
end=int(input())
Ecount=0
Ocount=0
for i in range(strt,end+1):
    if i%2==0:
        Ecount+=1
    else:
        Ocount+=1;
print("Even Number="+str(Ecount))
print("Odd Number="+str(Ocount))
```

#### Output :

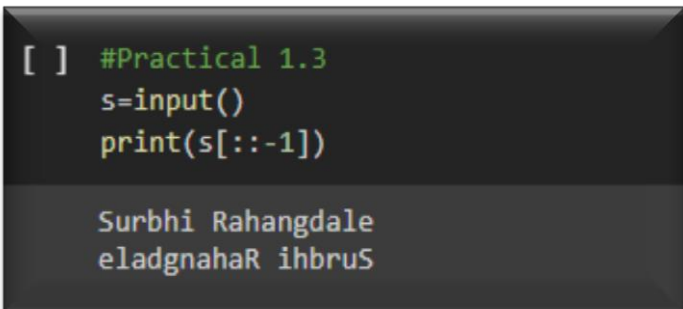


```
7
20
Even Number=7
Odd Number=7
```

### Practical 1.3

```
s=input()
print(s[::-1])
```

#### Output :



```
[ ] #Practical 1.3
    s=input()
    print(s[::-1])

Surbhi Rahangdale
eladgnahaR ihbruS
```

### Practical 1.4

#### Code:

```
lst=[5,6,4,1,2,3,8,1,2,6,4,9,1,3,5]
sort_lst=sorted(set(lst))
print(sort_lst)
```

#### Output:

```
[1, 2, 3, 4, 5, 6, 8, 9]
```

### Practical 1.5

#### Code:

```
Ec=0
Oc=0
lst=[1,25,4,66,54,8,2,44,6,2,5,7,9,3,11,2,5,4,9,5,5]
for i in lst:
    if i%2==0:
        Ec+=1
    else:
        Oc+=1;
print("Even Number="+str(Ec))
print("Odd Number="+str(Oc))
```

#### Output :

```
Even Number=10
Odd Number=11
```

## **Practical No. 2**

### **Tasks:**

1. Write a Python program which iterates over the integers from 1 to 50. For multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

Sample Output :

fizzbuzz

1

2

fizz

4

buzz

2. Write a Python program which takes two digits m (row) and n (column) as input and generates a two-dimensional array. The element value in the i-th row and j-th column of the array should be  $i*j$ . Note :

$i = 0, 1 \dots, m-1$

$j = 0, 1, n-1$ .

Test Data : Rows = 3, Columns = 4

Expected Result :  $[[0, 0, 0, 0], [0, 1, 2, 3], [0, 2, 4, 6]]$

3. Write a Python program which searches for prime numbers between 2 to 9:

### **Practical 2.1**

#### **Code:**

```
for i in range(1,51):
```

```
    if i%3==0 and i%5==0:
```

```
        print("FizzBuzz")
```

```
    elif i%3==0:
```

```
        print("Fizz")
```

```
    elif i%5==0:
```

```
        print("Buzz")
```

```
    else:
```

```
        print(i)
```

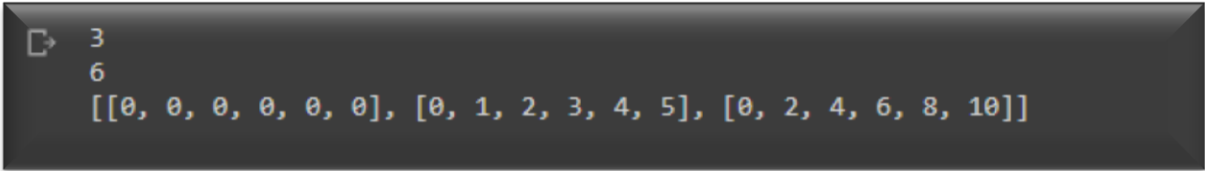
## Output:

```
[ ] 1  
    2  
    Fizz  
    4  
    Buzz  
    Fizz  
    7  
    8  
    Fizz  
    Buzz  
    11  
    Fizz  
    13  
    14  
    FizzBuzz  
    16  
    17  
    Fizz  
    19  
    Buzz  
    Fizz  
    22  
    23  
    Fizz  
    Buzz  
    26  
    Fizz  
    28  
    29  
    FizzBuzz  
    31
```

```
▶ FizzBuzz  
31  
32  
Fizz  
34  
Buzz  
Fizz  
37  
38  
Fizz  
Buzz  
41  
Fizz  
43  
44  
FizzBuzz  
46  
47  
Fizz  
49  
Buzz
```

**Practical 2.2****Code:**

```
m=int(input())
n=int(input())
twoD=[]
for i in range(m):
    temp=[]
    for j in range(n):
        temp.append(i*j)
    twoD.append(temp)
print(twoD)
```


**Output:**

```
3
6
[[0, 0, 0, 0, 0, 0], [0, 1, 2, 3, 4, 5], [0, 2, 4, 6, 8, 10]]
```

**Practical 2.3**

```
for i in range(2,10):
    flag=True
    for j in range(2,i):
        if i%j==0:
            flag=False
            break
    if flag:
        print(i)
```

Output :



```
2
3
5
7
```

### Practical No.3

**Name:** Tasks Based on Numpy package

**Objective:** Understanding the use of Numpy package

**Tasks:**

1.Extract all odd numbers from arr

Input:

```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Desired output:

```
#> array([1, 3, 5, 7, 9])
```

2.Replace all odd numbers in arr with -1 Input:arr =

```
np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Desired Output:

```
#> array([ 0, -1, 2, -1, 4, -1, 6, -1, 8, -1])
```

3.Replace all odd numbers in arr with -1 without changing arr

Input:

```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Desired Output:

```
#> array([ 0, -1, 2, -1, 4, -1, 6, -1, 8, -1])
```

4.Convert a 1D array to a 2D array with 2 rows

Input:

```
np.arange(10)
```

```
#> array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Desired Output:

```
#> array([[0, 1, 2, 3, 4],
```

```
#>      [5, 6, 7, 8, 9]])
```

5.Stack arrays a and b vertically

Input:

```
a = np.arange(10).reshape(2,-1)
```

```
b = np.repeat(1, 10).reshape(2,-1)
```

Desired Output:

```
#> array([[0, 1, 2, 3, 4],
```

```
#>      [5, 6, 7, 8, 9],
```



```
#> [1, 1, 1, 1, 1],
```

```
#> [1, 1, 1, 1, 1]])
```

6. Get the common items between a and b

Input:

```
a = np.array([1,2,3,2,3,4,3,4,5,6])
```

```
b = np.array([7,2,10,2,7,4,9,4,9,8])
```

Desired Output:

```
array([2, 4])
```

### Practical 3.1

**Code:**

```
import numpy as np
```

```
arr=np.array([0,1,2,3,4,5,6,7,8,9])
```

```
odd=[]
```

```
for i in range(len(arr)):
```

```
if arr[i]%2!=0:
```

```
odd.append(arr[i])
```

```
print(odd)
```

**Output:**

```
[1, 3, 5, 7, 9]
```

### Practical 3.2

```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
for i in range(len(arr)):
```

```
    if arr[i]%2!=0:
```

```
        arr[i]=-1
```

```
arr
```

**OUTPUT :**

```
array([ 0, -1,  2, -1,  4, -1,  6, -1,  8, -1])
```

### Practical 3.3

```
arr=np.array([0,1,2,3,4,5,6,7,8,9])
```

```
arr2=[]
```

```
for i in range(len(arr)):
```

```
    if arr[i]%2==0:
```

```
        arr2.append(arr[i])
```

```
    else:
```

```
        arr2.append(-1)
```

```
arr2
```

**Output:**

```
[0, -1, 2, -1, 4, -1, 6, -1, 8, -1]
```

### Practical 3.4

```
arr=np.arange(10)
arr=arr.reshape(2,5)
arr
```

**Output:**

```
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
```

### Practical 3.5

```
a = np.arange(10).reshape(2,-1)
b = np.repeat(1, 10).reshape(2,-1)
Varr=np.vstack([a,b])
Varr
```

**Output:**

```
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]])
```

### Practical 3.6

```
a = np.array([1,2,3,2,3,4,3,4,5,6])
b = np.array([7,2,10,2,7,4,9,4,9,8])
arr=np.intersect1d(a,b)
arr
```

**Output:**

```
array([2, 4])
```

**Practical No.4****Tasks:**

1.From array a remove all items present in array b

Input:

```
a = np.array([1,2,3,4,5])
```

```
b = np.array([5,6,7,8,9])
```

**Desired Output:**

```
array([1,2,3,4])
```

2.Get the positions where elements of a and b match

Input:

```
a = np.array([1,2,3,2,3,4,3,4,5,6])
```

```
b = np.array([7,2,10,2,7,4,9,4,9,8])
```

**Desired Output:**

```
#> (array([1, 3, 5, 7]),)
```

3.Get all items between 5 and 10 from a.

Input:

```
a = np.array([2, 6, 1, 9, 10, 3, 27])
```

**Desired Output:**

```
(array([6, 9, 10]),)
```

4.Swap columns 1 and 2 in the array arr

```
arr = np.arange(9).reshape(3,3)
```

5.Swap rows 1 and 2 in the array arr:

```
arr = np.arange(9).reshape(3,3)
```

6.Create a 2D array of shape 5×3 to contain random decimal numbers. Display numbers with only 3 decimal places

7.Limit the number of items printed in python numpy array a to a maximum of 6 elements.

Input:

```
a = np.arange(15)
```

```
#> array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

**Desired Output:**

```
#> array([ 0,  1,  2, ..., 12, 13, 14])
```

### Practical 4.1

```
import numpy as np
```

```
a = np.array([1,2,3,4,5])
```

```
b = np.array([5,6,7,8,9])
```

```
result=np.setdiff1d(a,b)
```

```
result
```

**Output:**

```
array([1, 2, 3, 4])
```

### Practical 4.2

```
a = np.array([1,2,3,2,3,4,3,4,5,6])
```

```
b = np.array([7,2,10,2,7,4,9,4,9,8])
```

```
post=np.where(a==b)
```

```
post
```

**Output:**

```
(array([1, 3, 5, 7]),)
```

### Practical 4.3

```
a = np.array([2, 6, 1, 9, 10, 3, 27])
```

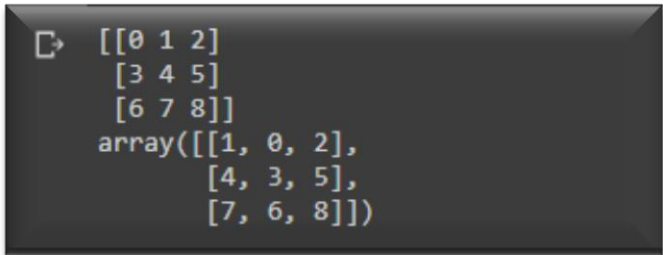
```
a[(a>=5) & (a<=10)]
```

**Output:**

```
array([ 6,  9, 10])
```

**Practical 4.4**

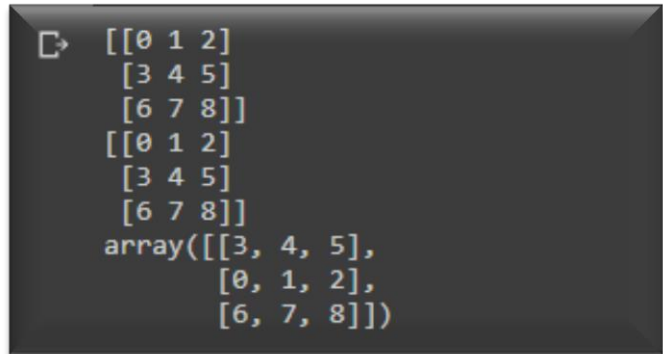
```
arr = np.arange(9).reshape(3,3)
print(arr)
arr[:,[1,0,2]]
```

**Output:**

```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
array([[1, 0, 2],
       [4, 3, 5],
       [7, 6, 8]])
```

**Practical 4.5**

```
arr = np.arange(9).reshape(3,3)
print(arr)
arr[[1,0,2],:]
```

**Output:**

```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
[[0 1 2]
 [3 4 5]
 [6 7 8]]
array([[3, 4, 5],
       [0, 1, 2],
       [6, 7, 8]])
```

**Practical 4.6**

```
rand_arr = np.random.random([5,3])  
np.set_printoptions(precision=3)  
rand_arr[:4]
```

**Output:**

```
[[0 1 2]  
 [3 4 5]  
 [6 7 8]]  
[[0 1 2]  
 [3 4 5]  
 [6 7 8]]  
array([[0.186, 0.357, 0.702],  
       [0.985, 0.023, 0.822],  
       [0.322, 0.205, 0.782],  
       [0.607, 0.719, 0.961]])
```

**Practical 4.7**

```
np.set_printoptions(threshold=6)  
a = np.arange(15)  
a
```

**Output:**

```
array([ 0,  1,  2, ..., 12, 13, 14])
```

## Practical No. 5

**Name:** Perform Data analysis of IMDB movies data

**Objective:** Understand data analysis using pandas

**Prerequisite:** MDB-movies-dataset. It is an open-source dataset and you can download it from this link.

<https://www.kaggle.com/PromptCloudHQ/imdb-data>

**Tasks:**

Read the data from the .csv file and perform the following basic operations on movies data

1. Read data
2. View the data: do a quick preview of the data by using appropriate methods
3. Understand some basic information about the data: Use appropriate functions to understand the shape, number of columns, indexes, and other information and description about the dataframe. Write some observations from the information you see
4. Data Selection – Indexing and Slicing data:
  - a. Extract 'Genre' data from the dataframe; as a series and as a data frame
  - b. Extract multiple columns from the data frame
  - c. Display the title revenue and rating for data in rows 10 to 15
  - d. Display the Genre, Actors, Directors, Rating, Revenue for movie 'Suicide Squad' or of your choice
5. Data Selection – Based on Conditional filtering
  - a. Pick only movies that are released from 2010 to 2016, have a rating of less than 6.0 but topped(quantile) in terms of revenue?

### Practical 5.1

```
import pandas as pd
```

```
data=pd.read_csv('IMDB-Movie-Data (1).csv')
```

```

Rank      Title  ... Revenue (Millions) Metascore
0         1  Guardians of the Galaxy  ...      333.13      76.0
1         2      Prometheus  ...      126.46      65.0
2         3          Split  ...      138.12      62.0
3         4           Sing  ...      270.32      59.0
4         5    Suicide Squad  ...      325.02      40.0
..      ...      ...      ...      ...      ...
995     996  Secret in Their Eyes  ...         NaN      45.0
996     997    Hostel: Part II  ...      17.54      46.0
997     998  Step Up 2: The Streets  ...      58.01      50.0
998     999    Search Party  ...         NaN      22.0
999    1000      Nine Lives  ...      19.64      11.0

[1000 rows x 12 columns]
```

```
[1000 rows x 12 columns]
```

```
0         1
```

```
Rank      Title
```

```
...
```

```
Revenue (Millions)
```

```
Metascore
```



## Practical 5.2

data.head(5)

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Revenue	Revenue
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.46
2	3	Split	Horror,Thriller	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	7.3	157606	138.12
3	4	Sing	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	270.32
4	5	Suicide Squad	Action,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.02

## Practical 5.3

Code:

```
print("shape = "+str(data.shape))

print("indexes = "+str(data.shape[0]))
print("columns = "+str(data.shape[1]))
print(data.info())
print(data.describe())
```

**Output:**

```

shape = (1000, 12)
indexes = 1000
columns = 12
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Rank                   1000 non-null   int64
1   Title                  1000 non-null   object
2   Genre                  1000 non-null   object
3   Description             1000 non-null   object
4   Director               1000 non-null   object
5   Actors                 1000 non-null   object
6   Year                   1000 non-null   int64
7   Runtime (Minutes)     1000 non-null   int64
8   Rating                 1000 non-null   float64
9   Votes                  1000 non-null   int64
10  Revenue (Millions)    872 non-null    float64
11  Metascore              936 non-null    float64
dtypes: float64(3), int64(4), object(5)
memory usage: 93.9+ KB
None

```

	Rank	Year	...	Revenue (Millions)	Metascore
count	1000.000000	1000.000000	...	872.000000	936.000000
mean	500.500000	2012.783000	...	82.956376	58.985043
std	288.819436	3.205962	...	103.253540	17.194757
min	1.000000	2006.000000	...	0.000000	11.000000
25%	250.750000	2010.000000	...	13.270000	47.000000
50%	500.500000	2014.000000	...	47.985000	59.500000
75%	750.250000	2016.000000	...	113.715000	72.000000
max	1000.000000	2016.000000	...	936.630000	100.000000

[8 rows x 7 columns]

**Practical 5.4**

```
print(data['Genre'])
print(data[['Title','Actors','Rating']])
print(data.iloc[10:16,[1,8,10]])
print(data.loc[data['Title']=='Sing',['Genre','Title','Actors','Director','Rating','Revenue
(Millions)']])
```

**Output:**

```
[8 rows x 7 columns]
   Title  ... Rating
0  Guardians of the Galaxy  ...   8.1
1      Prometheus  ...   7.0
2      Split  ...   7.3
3      Sing  ...   7.2
4  Suicide Squad  ...   6.2
..  ...  ...  ...
995  Secret in Their Eyes  ...   6.2
996  Hostel: Part II  ...   5.5
997  Step Up 2: The Streets  ...   6.2
998  Search Party  ...   5.6
999  Nine Lives  ...   5.3

[1000 rows x 3 columns]
   Title  Rating  Revenue (Millions)
10  Fantastic Beasts and Where to Find Them  7.5  234.02
11  Hidden Figures  7.8  169.27
12  Rogue One  7.9  532.17
13  Moana  7.7  248.75
14  Colossal  6.4  2.87
15  The Secret Life of Pets  6.6  368.31
   Genre Title  ... Rating Revenue (Millions)
3  Animation,Comedy,Family Sing  ...   7.2  270.32

[1 rows x 6 columns]
```

## Practical 5.5

```
data[(data['Year']>=2010) & (data['Year']<=2016) & (data['Rating']<6) ]
```

**Output:**

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	(Millions)
24	25	Independence Day: Resurgence	Action,Adventure,Sci-Fi	Two decades after the first Independence Day i...	Roland Emmerich	Liam Hemsworth, Jeff Goldblum, Bill Pullman, Ma...	2016	120	5.3	127553	103.14
27	28	Dead Awake	Horror,Thriller	A young woman must save herself and her friend...	Phillip Guzman	Jocelin Donahue, Jesse Bradford, Jesse Borrego...	2016	99	4.7	523	0.01
29	30	Assassin's Creed	Action,Adventure,Drama	When Callum Lynch explores the memories of his...	Justin Kurzel	Michael Fassbender, Marion Cotillard, Jeremy I...	2016	115	5.9	112813	54.65
34	35	Resident Evil: The Final Chapter	Action,Horror,Sci-Fi	Alice returns to where the nightmare began: Th...	Paul W.S. Anderson	Milla Jovovich, Iain Glen, Ali Larter, Shawn R...	2016	107	5.6	46165	26.84
42	43	Don't Fuck in the Woods	Horror	A group of friends are going on a camping trip...	Shawn Burkett	Brittany Blanton, Ayse Howard, Roman Jossart, N...	2016	73	2.7	496	NaN

985	986	Your Highness	Adventure, Comedy, Fantasy	When Prince Fabious's bride is kidnapped, he g...	David Gordon Green	Danny McBride, Natalie Portman, James Franco, ...	2011	102	5.6	87904	21.56
986	987	Final Destination 5	Horror, Thriller	Survivors of a suspension-bridge collapse lear...	Steven Quale	Nicholas D'Agosto, Emma Bell, Arlen Escarpeta, ...	2011	92	5.9	88000	42.58
993	994	Resident Evil: Afterlife	Action, Adventure, Horror	While still out to destroy the evil Umbrella C...	Paul W.S. Anderson	Milla Jovovich, Ali Larter, Wentworth Miller, K...	2010	97	5.9	140900	60.13
998	999	Search Party	Adventure, Comedy	A pair of friends embark on a mission to reuni...	Scot Armstrong	Adam Pally, T.J. Miller, Thomas Middleditch, Sh...	2014	93	5.6	4881	NaN
999	1000	Nine Lives	Comedy, Family, Fantasy	A stuffy businessman finds himself trapped ins...	Barry Sonnenfeld	Kevin Spacey, Jennifer Garner, Robbie Amell, Ch...	2016	87	5.3	12435	19.64

160 rows x 12 columns

**Practical No. 6****Name:** Perform Data analysis of IMDB movies data**Objective:** Understand data analysis *using pandas***Tasks:**

1. Groupby operations: List the average rating for each director
2. Sorting operation: List the average rating for each 'Director' and sort them from highly rated to lowest
3. Dealing with missing values
4. Dropping columns and null values:
  - b. Drop the 'Metascore' column completely from the data. The operation should make changes to original data
  - c. Drop all the rows and columns containing more than 5 null values
  - d. Fill these null values of Revenue (Millions) with mean Revenue (Millions).
5. apply() functions:
  - e. Write a function to classify the movies based on their ratings as good ( $\geq 7.5$ ), average ( $\geq 5$ ) bad ( $< 5$ ),
  - f. Apply this function to the dataframe's Rating column.
  - g. Simultaneously add this new Rating\_Category to the data frame

**Practical 6.1**

```
print(data.groupby('Director')['Rating'].mean())
```

**Output:**

```
Director
Aamir Khan      8.50
Abdellatif Kechiche  7.80
Adam Leon       6.50
Adam McKay      7.00
Adam Shankman   6.30
...
Xavier Dolan    7.55
Yimou Zhang     6.10
Yorgos Lanthimos 7.20
Zack Snyder     7.04
Zackary Adler   5.10
Name: Rating, Length: 644, dtype: float64
```

**Practical 6.2**

```
print(data.groupby('Director')['Rating'].mean().sort_values(ascending=False))
```

```
Director
Nitesh Tiwari      8.80
Christopher Nolan  8.68
Makoto Shinkai     8.60
Olivier Nakache    8.60
Aamir Khan         8.50
...
Femi Oyeniran      3.50
Jonathan Holbrook  3.20
Shawn Burkett      2.70
James Wong         2.70
Jason Friedberg    1.90
Name: Rating, Length: 644, dtype: float64
```

**Practical 6.3**

```
data.dropna(axis=0,how='any')
```

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Vot		
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13	76.0
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.46	65.0
2	3	Split	Horror,Thriller	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	7.3	157606	138.12	62.0
3	4	Sing	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	270.32	59.0
4	5	Suicide Squad	Action,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.02	40.0
...	...	...	...	...	...	...	...	...	...	...	...	...
993	994	Resident Evil: Afterlife	Action,Adventure,Horror	While still out to destroy the evil Umbrella C...	Paul W.S. Anderson	Milla Jovovich, Ali Larter, Wentworth Miller,K...	2010	97	5.9	140900	60.13	37.0
994	995	Project X	Comedy	3 high school seniors throw a birthday party t...	Nima Nourizadeh	Thomas Mann, Oliver Cooper, Jonathan Daniel Br...	2012	88	6.7	164088	54.72	48.0

996	997	Hostel: Part II	Horror	Three American college students studying abroa...	Eli Roth	Lauren German, Heather Matarazzo, Bijou Philli...	2007	94	5.5	73152	17.54	46.0
997	998	Step Up 2: The Streets	Drama,Music,Romance	Romantic sparks occur between two dance studen...	Jon M. Chu	Robert Hoffman, Briana Evigan, Cassie Ventura,...	2008	98	6.2	70699	58.01	50.0
999	1000	Nine Lives	Comedy,Family,Fantasy	A stuffy businessman finds himself trapped ins...	Barry Sonnenfeld	Kevin Spacey, Jennifer Garner, Robbie Amell,Ch...	2016	87	5.3	12435	19.64	11.0

338 rows x 12 columns





**Practical 6.4**

```
data.drop('Metascore',axis=1,inplace=True)
data
data.dropna(axis=0,thresh=5,inplace=True)
data.dropna(axis=1,thresh=5,inplace=True)
data
data['Revenue (Millions)'].fillna(data['Revenue (Millions)'].mean(),inplace=True)
data
```

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.46
2	3	Split	Horror,Thriller	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	7.3	157606	138.12
3	4	Sing	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	270.32
4	5	Suicide Squad	Action,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.02
...	...	...	...	...	...	...	...	...	...	...	...
995	996	Secret in Their Eyes	Crime,Drama,Mystery	A tight-knit team of rising investigators, alo...	Billy Ray	Chiwetel Ejiofor, Nicole Kidman, Julia Roberts...	2015	111	6.2	27585	NaN
996	997	Hostel: Part II	Horror	Three American college students studying abroa...	Eli Roth	Lauren German, Heather Matarazzo, Bijou Philli...	2007	94	5.5	73152	17.54
997	998	Step Up 2: The Streets	Drama,Music,Romance	Romantic sparks occur between two dance studen...	Jon M. Chu	Robert Hoffman, Briana Evigan, Cassie Ventura,...	2008	98	6.2	70699	58.01
998	999	Search Party	Adventure,Comedy	A pair of friends embark on a mission to reuni...	Scot Armstrong	Adam Pally, T.J. Miller, Thomas Middleditch, Sh...	2014	93	5.6	4881	NaN
999	1000	Nine Lives	Comedy,Family,Fantasy	A stuffy businessman finds himself trapped ins...	Barry Sonnenfeld	Kevin Spacey, Jennifer Garner, Robbie Amell, Ch...	2016	87	5.3	12435	19.64
1000 rows × 11 columns											

**Practical 6.5****Code:**

```
def rating_gp(rating):
    if(rating>=7.5):
        return 'Good'
    elif(rating>=5):
        return 'Avarage'
    else:
        return 'Bad'
data['Rating_category']=data['Rating'].apply(rating_gp)
data
```

**Output:**

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Rating_category
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.130000	Good
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.460000	Avarage
2	3	Split	Horror,Thriller	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	7.3	157606	138.120000	Avarage
3	4	Sing	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	270.320000	Avarage
4	5	Suicide Squad	Action,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.020000	Avarage
...	...	...	...	...	...	...	...	...	...	...	...	...
995	996	Secret in Their Eyes	Crime,Drama,Mystery	A tight-knit team of rising investigators, alo...	Billy Ray	Chiwetel Ejiofor, Nicole Kidman, Julia Roberts...	2015	111	6.2	27585	82.956376	Avarage
996	997	Hostel: Part II	Horror	Three American college students studying abroa...	Eli Roth	Lauren German, Heather Matarazzo, Bijou Philli...	2007	94	5.5	73152	17.540000	Avarage
997	998	Step Up 2: The Streets	Drama,Music,Romance	Romantic sparks occur between two dance studen...	Jon M. Chu	Robert Hoffman, Briana Evigan, Cassie Ventura,...	2008	98	6.2	70699	58.010000	Avarage
998	999	Search Party	Adventure,Comedy	A pair of friends embark on a mission to reuni...	Scot Armstrong	Adam Pally, T.J. Miller, Thomas Middleditch,Sh...	2014	93	5.6	4881	82.956376	Avarage
999	1000	Nine Lives	Comedy,Family,Fantasy	A stuffy businessman finds himself trapped ins...	Barry Sonnenfeld	Kevin Spacey, Jennifer Garner, Robbie Amell,Ch...	2016	87	5.3	12435	19.640000	Avarage

**Practical No. 7**

**Name:** Predict the species of a new flower from its dimensions using KNN

algorithmObjectives:Use Scikit Learn Package

**Tasks:**

1. Make use of the iris data set in scikit learn package.
2. Import the iris data set
3. Divide data into train and test values
4. Predict the species of flower with dimensions

`x_new = np.array([[5, 2.9, 1, 0.2]])`

5. Print the name of the predicted species

**Code:**

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn import datasets
from sklearn.decomposition import PCA
iris = datasets.load_iris()
X = iris.data[:, :2]
y = iris.target
x_min, x_max = X[:, 0].min() - 0.5, X[:, 0].max() + 0.5

y_min, y_max = X[:, 1].min() - 0.5, X[:, 1].max() + 0.5

plt.figure(2, figsize=(8, 6))

plt.clf()
```

```
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Set1, edgecolor="k")

plt.xlabel("Sepal length")

plt.ylabel("Sepal width")

plt.xlim(x_min, x_max)

plt.ylim(y_min, y_max)

plt.xticks(())

plt.yticks(())

fig = plt.figure(1, figsize=(8, 6))

ax = Axes3D(fig, elev=-150, azimuth=110)

X_reduced = PCA(n_components=3).fit_transform(iris.data)
ax.scatter(
    X_reduced[:, 0],

    X_reduced[:, 1],

    X_reduced[:, 2],
    c=y,
    cmap=plt.cm.Set1,
    edgecolor="k",
    s=40,
)
ax.set_title("First three PCA directions")

ax.set_xlabel("1st eigenvector")

ax.w_xaxis.set_ticklabels([])

ax.set_ylabel("2nd eigenvector")

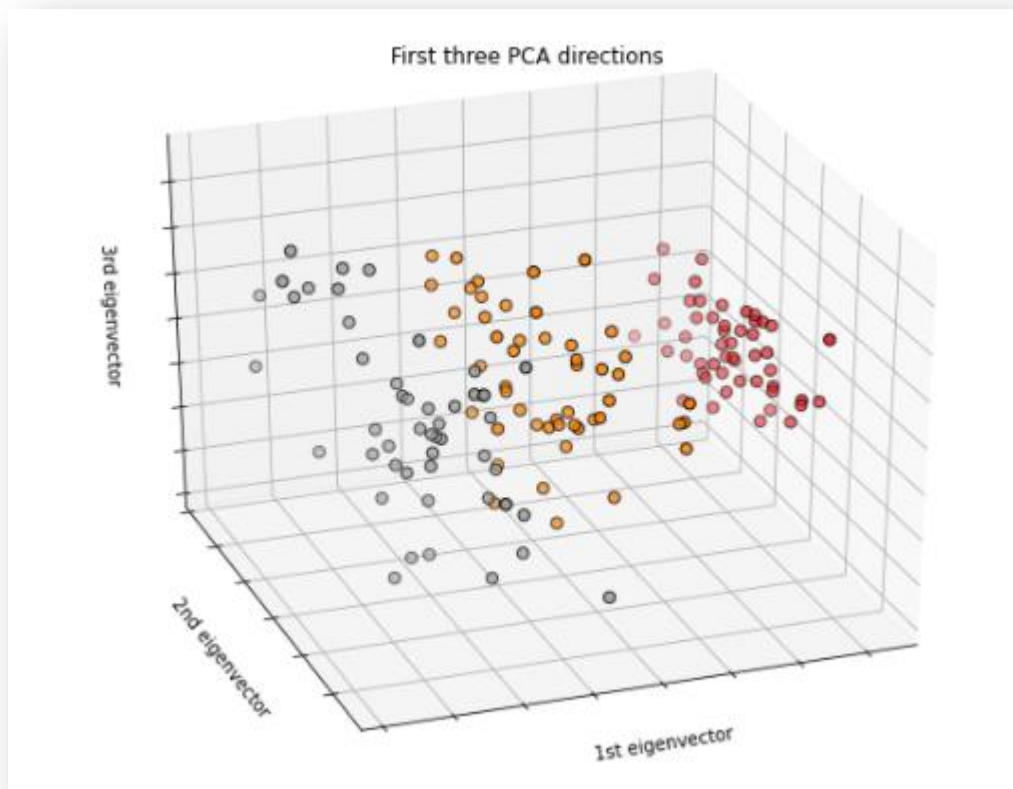
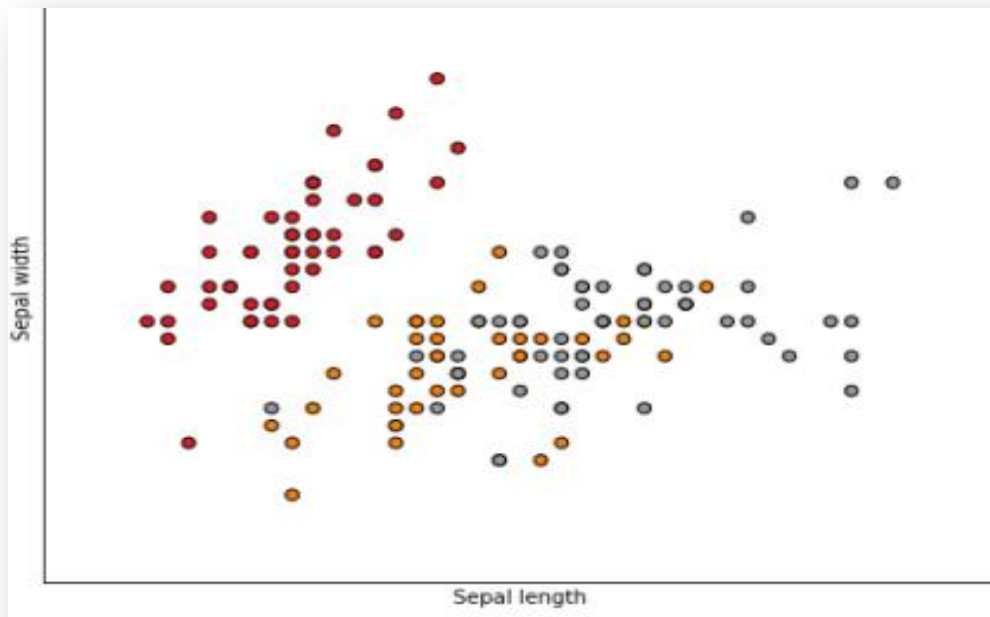
ax.w_yaxis.set_ticklabels([])

ax.set_zlabel("3rd eigenvector")

ax.w_zaxis.set_ticklabels([])

plt.show()
```

**Output:**



## **Practical No. 8**

**Aim:** To plot a histogram using *seaborn*

**visualizationTasks:**

Plot a Distribution Plot in Seaborn with the data at <https://www.kaggle.com/shivamb/netflix-shows>

8.1 Visualize the distribution of the `release_year` feature, to see when Netflix was the most active with new additions

8.2 Visualize the distribution of each of these `release_years` in percentages. Use the `stat` argument

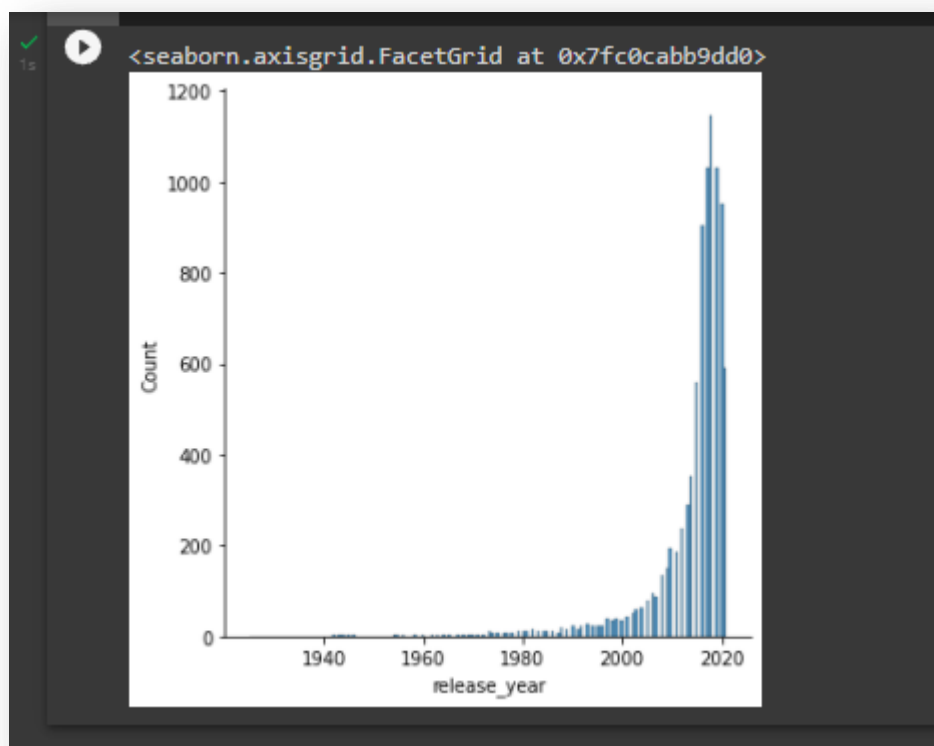
8.3 Change the bin size

## Practical 8.1

### Code:

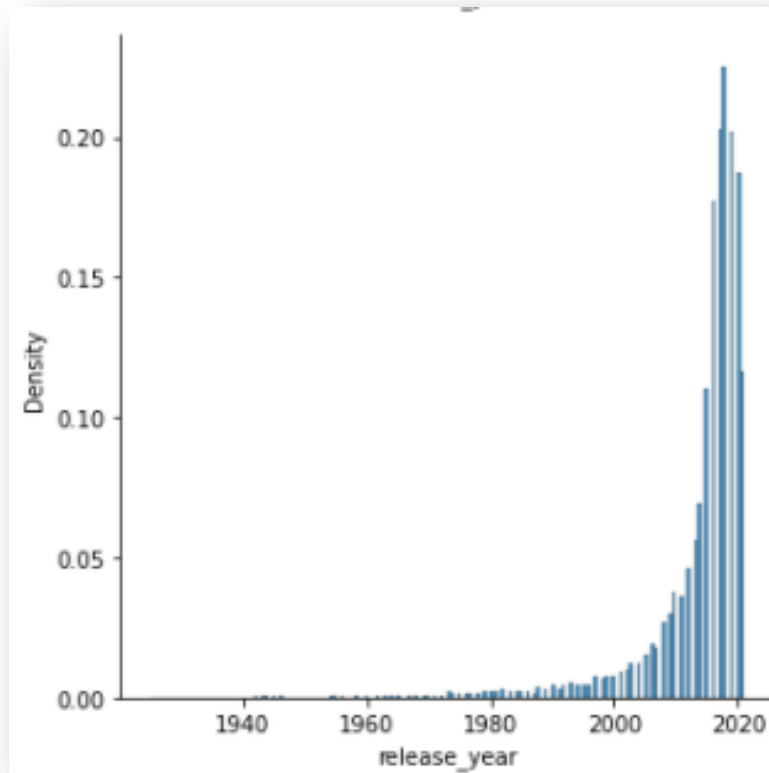
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
df = pd.read_csv('netflix_titles.csv')
data = df['release_year']
sns.displot(data)
```

### Output:



## Practical 8.2

```
data = df['release_year']  
sns.displot(data, stat='density')
```



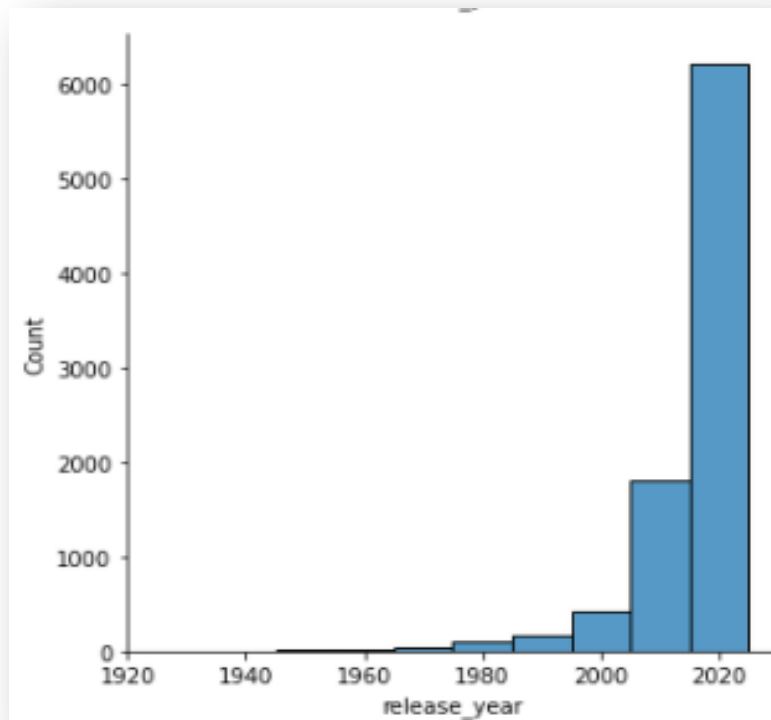


### Practical 8.3

```
data = df['release_year']
```

```
sns.displot(data, binwidth=10)
```

Output:



---

Thank You !