MAJOR PROJECT 2

RINEX PROJECT REPORT

NAME : Rahul Bhattacharya

COLLEGE : GITA AUTONOMOUS COLLEGE,BHUBANESWAR

YEAR : 2nd Year (2021-25)

Roll No: 2102127

MAIL: rahulbhattacharya661@gmail.com

GITHUB: https://github.com/RahulBh007

GDRIVE:

https://drive.google.com/open?id=1e_fJV1AfPn1wFsImeHJJLBXaL_5gkokP&authuser=0&usp=drive_link

MAJOR PROJECT 1 (COMPULSORY)

1. Choose any dataset of your choice and apply a suitable CLASSIFIER/REGRESSOR .

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

MAJOR PROJECT 1 (COMPULSORY)

```
#MACHINE LEARNING - SUPERVISED LEARNING - REGRESSION - LINEAR REGRESSION
#Univariate/Single - 1 column as input , 1  column as output
#MUltivariate/Multiple - multiple column as input , 1 column as output
#Dataset - https://www.kaggle.com/datasets/shubham47/students-score-dataset-linear-regression
#Study time in hours, Scores out of 100
```

```
#1.Take the data and create dataframe
import pandas as pd
df = pd.read_csv('/content/student_scores.csv')
df
#IMAGINARY STORY - IMAGINE a REAL ESTATE/PROPERTY BROKER COMES TO YOU AND GIVES YOU THE BELOW DATASET and says
#CREATE A MODEL FOR ME ,WHICH COULD PREDICT THE PROPERTY PRICES ,BASED ON THE DATA I GIVE
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

```
df.head()
```

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

MAJOR PROJECT 1 (COMPULSORY)

```
df.describe()
```

	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

```
df.corr()
```

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

```
df.columns
```

```
Index(['Hours', 'Scores'], dtype='object')
```

MAJOR PROJECT 1 (COMPULSORY)

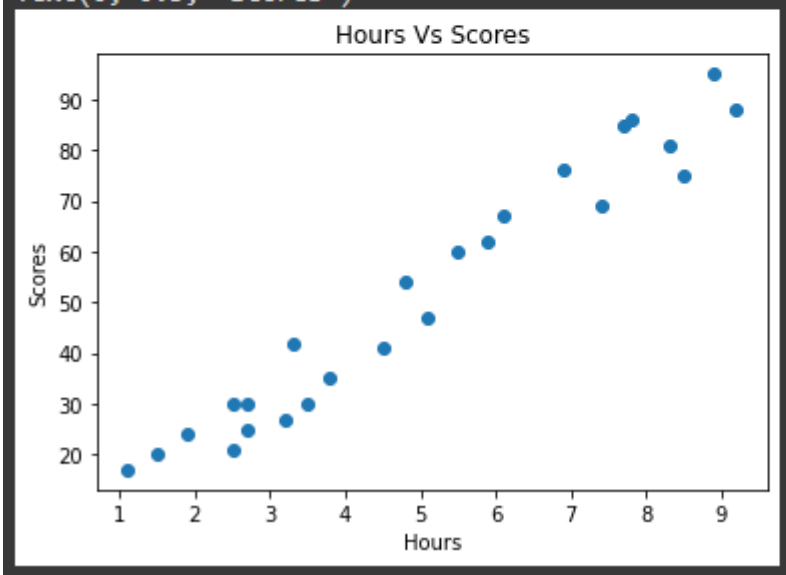
```
df.count
```

```
<bound method DataFrame.count of     Hours  Scores
0     2.5      21
1     5.1      47
2     3.2      27
3     8.5      75
4     3.5      30
5     1.5      20
6     9.2      88
7     5.5      60
8     8.3      81
9     2.7      25
10    7.7      85
11    5.9      62
12    4.5      41
13    3.3      42
14    1.1      17
15    8.9      95
16    2.5      30
17    1.9      24
18    6.1      67
19    7.4      69
20    2.7      30
21    4.8      54
22    3.8      35
23    6.9      76
24    7.8      86>
```

MAJOR PROJECT 1 (COMPULSORY)

```
import matplotlib.pyplot as plt
plt.scatter(df['Hours'],df['Scores'])
plt.title('Hours Vs Scores')
plt.xlabel('Hours')
plt.ylabel('Scores')
```

```
Text(0, 0.5, 'Scores')
```



```
#Slicing
#df.iloc[row slicing,column slicing]
x = df.iloc[0:6,0:1].values
x
#.values converts the dataframe into an array
```

```
array([[2.5],
       [5.1],
       [3.2],
       [8.5],
       [3.5],
       [1.5]])
```

MAJOR PROJECT 1 (COMPULSORY)

```
y = df.iloc[0:6,1].values
y
```

```
array([21, 47, 27, 75, 30, 20])
```

```
#Run classifier,REGRESSOR or clusterer(APPLYING a suitable ALGORITHM)
#sklearn.linear_model - package(collection of libraries),LinearRegression - Library
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

```
#FIT the MODEL(Mapping/Plotting the inputs with the outputs in the library)
#LinearRegression.fit(x,y)
model.fit(x,y) #We are mapping the values of x and y in the LinearRegression library
```

```
LinearRegression()
```

```
y_pred = model.predict(x) #Using the input values,we predict the output
y_pred
```

```
array([23.5920761 , 45.52364737, 29.49672991, 74.20339441, 32.02729582,
       15.15685639])
```

```
y
```

```
array([21, 47, 27, 75, 30, 20])
```

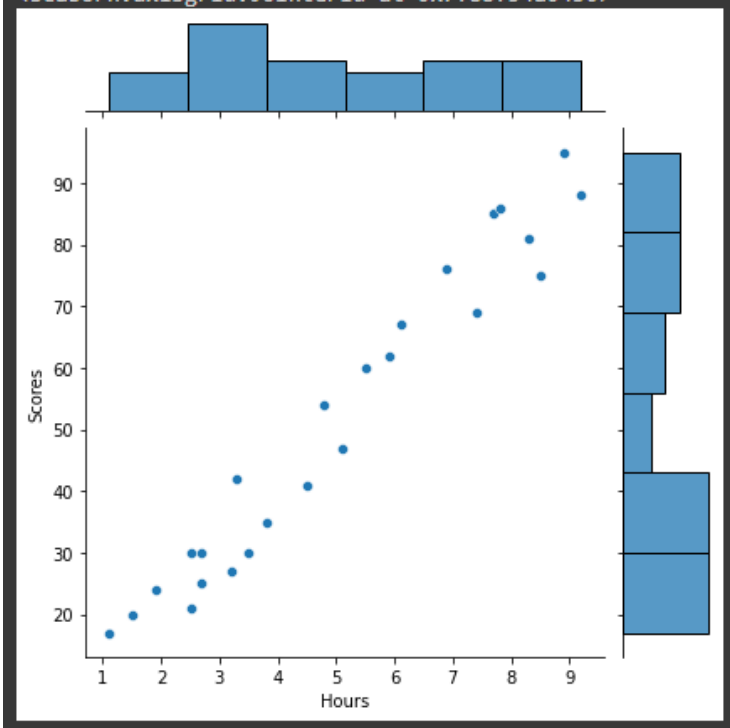
```
#INDIVIDUAL PREDICTION
#I want to know the Scores for 3.8 study hour
model.predict([[3.8]])
```

```
array([34.55786174])
```

MAJOR PROJECT 1 (COMPULSORY)

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.jointplot(data=df,x="Hours", y="Scores")
```

```
<seaborn.axisgrid.JointGrid at 0x7f88f54a0430>
```



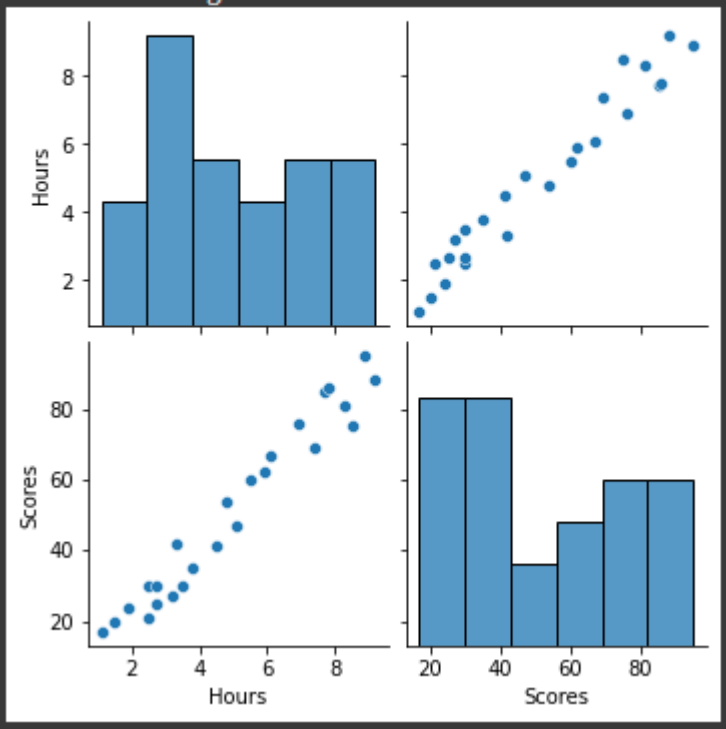
```
sns.heatmap(df.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f88f4f5a580>
```

MAJOR PROJECT 1 (COMPULSORY)

```
sns.pairplot(df)
```

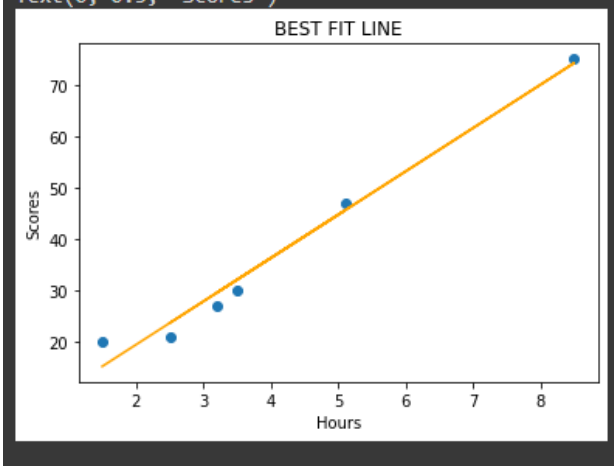
```
<seaborn.axisgrid.PairGrid at 0x7f88f4ecdac0>
```




```
#FINAL VISUALISATION - BEST FIT LINE

plt.scatter(x,y) #ACTUAL VALUES
plt.plot(x,y_pred,c = 'orange') #PREDICTED VALUES
plt.title('BEST FIT LINE')
plt.xlabel('Hours')
plt.ylabel('Scores')
```

```
Text(0, 0.5, 'Scores')
```



MAJOR PROJECT 1 (COMPULSORY)

```
#CROSS VERIFICATION
#y = mx + C #Equation of a Straight line
#m - Slope
#C - Constant/y-intercept
#y - dependant variable
#x - independant variable


#From excel m = 8.435, C = 2.5040
```

```
m = model.coef_ #slope(m)
m
```

```
array([8.43521972])
```

```
C = model.intercept_ # constant/y-intercept
C
```

```
2.5040268093616618
```

```
#y = mx +C
m*3.8 + C
```

```
array([34.55786174])
```

MAJOR PROJECT 2

2. Choose any dataset of your choice and Perform Exploratory Data Analysis for Atleast 15 different facts/Conclusions.

```
#EXPLORATORY DATA ANALYSIS - EDA(PRE MACHINE LEARNING)
```

```
#1.Take the Data and create DataFrame
import pandas as pd
df = pd.read_csv('/content/List of most expensive films.csv')
df
```

	Rank	Title	Year	Est Cost (inmillions)
0	1	Pirates of the Caribbean: On Stranger Tides	2011	$456
1	2	Avengers: Age of Ultron	2015	$417
2	3	Pirates of the Caribbean: At World's End	2007	$392
3	4	Avengers: Endgame	2019	$377
4	5	Avengers: Infinity War	2018	$351
5	6	Avatar: The Way of Water	2022	$350
6	7	Titanic	1997	$338
7	8	Spider-Man 3	2007	$337
8	9	Justice League	2017	$332
9	10	Tangled	2010	$323
10	11	Harry Potter and the Half-Blood Prince	2009	$316
11	12	John Carter	2012	$311
12	13	Waterworld	1995	$306
13	14	Pirates of the Caribbean: Dead Man's Chest	2006	$302
14	15	Avatar	2009	$299
15	16	Batman v Superman: Dawn of Justice	2016	$297
16	17	Solo: A Star Wars Story	2018	$297
17	18	Star Wars: The Force Awakens	2015	$296
18	19	Star Wars: The Rise of Skywalker	2019	$291
19	20	Star Wars: The Last Jedi	2017	$290
20	21	King Kong	2005	$287
21	22	Spider-Man 2	2004	$287
22	23	Furious 7	2015	$286
23	24	The Chronicles of Narnia: Prince Caspian	2008	$283
24	25	X-Men: The Last Stand	2006	$282
25	26	Beauty and the Beast	2017	$282
26	27	Spectre	2015	$280
27	28	Wild Wild West	1999	$277
28	29	The Fate of the Furious	2017	$276
29	30	The Lion King	2019	$276

MAJOR PROJECT 2

```
print ("Type : ", type(df), "\n\n")
```

```
Type :  <class 'pandas.core.frame.DataFrame'>
```

```
df.describe()
```

	Rank	Year
count	30.000000	30.000000
mean	15.500000	2011.466667
std	8.803408	6.996222
min	1.000000	1995.000000
25%	8.250000	2007.000000
50%	15.500000	2013.500000
75%	22.750000	2017.000000
max	30.000000	2022.000000

```
df.shape #(30,6) - 30 rows, 4 columns
```

```
(30, 4)
```

```
df.size #Total no of elements present
```

```
120
```

```
df.info() #Provides information about the dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 4 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   Rank                   30 non-null     int64 
 1   Title                  30 non-null     object
 2   Year                   30 non-null     int64 
 3   Est Cost (inmillions)  30 non-null     object
dtypes: int64(2), object(2)
memory usage: 1.1+ KB
```

MAJOR PROJECT 2

```
print ("Head -- \n",df.head(10))
```

```
Head --
    Rank                                        Title  Year  \
0     1  Pirates of the Caribbean: On Stranger Tides  2011
1     2                    Avengers: Age of Ultron  2015
2     3    Pirates of the Caribbean: At World's End  2007
3     4                          Avengers: Endgame  2019
4     5                    Avengers: Infinity War  2018
5     6                  Avatar: The Way of Water  2022
6     7                                   Titanic  1997
7     8                              Spider-Man 3  2007
8     9                            Justice League  2017
9    10                                   Tangled  2010

  Est Cost (inmillions)
0                  $456
1                  $417
2                  $392
3                  $377
4                  $351
5                  $350
6                  $338
7                  $337
8                  $332
9                  $323
```

```
df.nunique()
```

```
Rank                    30
Title                   30
Year                    18
Est Cost (inmillions)   26
dtype: int64
```

MAJOR PROJECT 1 (COMPULSORY)

```
df['Title'].unique()
```

```
array(['Pirates of the Caribbean: On Stranger Tides',
       'Avengers: Age of Ultron',
       "Pirates of the Caribbean: At World's End", 'Avengers: Endgame',
       'Avengers: Infinity War', 'Avatar: The Way of Water', 'Titanic',
       'Spider-Man 3', 'Justice League', 'Tangled',
       'Harry Potter and the Half-Blood Prince', 'John Carter',
       'Waterworld', "Pirates of the Caribbean: Dead Man's Chest",
       'Avatar', 'Batman v Superman: Dawn of Justice',
       'Solo: A Star Wars Story', 'Star Wars: The Force Awakens',
       'Star Wars: The Rise of Skywalker', 'Star Wars: The Last Jedi',
       'King Kong', 'Spider-Man 2', 'Furious 7',
       'The Chronicles of Narnia: Prince Caspian',
       'X-Men: The Last Stand', 'Beauty and the Beast', 'Spectre',
       'Wild Wild West', 'The Fate of the Furious', 'The Lion King'],
      dtype=object)
```

```
df['Year'].unique()
```

```
array([2011, 2015, 2007, 2019, 2018, 2022, 1997, 2017, 2010, 2009, 2012,
       1995, 2006, 2016, 2005, 2004, 2008, 1999])
```

```
df['Est Cost (inmillions)'].unique()
```

```
array(['$456', '$417', '$392', '$377', '$351', '$350', '$338', '$337',
       '$332', '$323', '$316', '$311', '$306', '$302', '$299', '$297',
       '$296', '$291', '$290', '$287', '$286', '$283', '$282', '$280',
       '$277', '$276'], dtype=object)
```

MAJOR PROJECT 1 (COMPULSORY)

```
df["Title"].value_counts()
```

```
Pirates of the Caribbean: On Stranger Tides    1
Avengers: Age of Ultron                        1
The Fate of the Furious                        1
Wild Wild West                                 1
Spectre                                        1
Beauty and the Beast                           1
X-Men: The Last Stand                          1
The Chronicles of Narnia: Prince Caspian       1
Furious 7                                      1
Spider-Man 2                                   1
King Kong                                      1
Star Wars: The Last Jedi                       1
Star Wars: The Rise of Skywalker               1
Star Wars: The Force Awakens                   1
Solo: A Star Wars Story                        1
Batman v Superman: Dawn of Justice             1
Avatar                                         1
Pirates of the Caribbean: Dead Man's Chest     1
Waterworld                                     1
John Carter                                    1
Harry Potter and the Half-Blood Prince         1
Tangled                                        1
Justice League                                 1
Spider-Man 3                                   1
Titanic                                        1
Avatar: The Way of Water                       1
Avengers: Infinity War                         1
Avengers: Endgame                              1
Pirates of the Caribbean: At World's End       1
The Lion King                                  1
Name: Title, dtype: int64
```

MAJOR PROJECT 1 (COMPULSORY)

```
df.groupby(['Title', 'Rank']).mean()
```

Title	Rank	Year
Avatar	15	2009.0
Avatar: The Way of Water	6	2022.0
Avengers: Age of Ultron	2	2015.0
Avengers: Endgame	4	2019.0
Avengers: Infinity War	5	2018.0
Batman v Superman: Dawn of Justice	16	2016.0
Beauty and the Beast	26	2017.0
Furious 7	23	2015.0
Harry Potter and the Half-Blood Prince	11	2009.0
John Carter	12	2012.0
Justice League	9	2017.0
King Kong	21	2005.0
Pirates of the Caribbean: At World's End	3	2007.0
Pirates of the Caribbean: Dead Man's Chest	14	2006.0
Pirates of the Caribbean: On Stranger Tides	1	2011.0
Solo: A Star Wars Story	17	2018.0
Spectre	27	2015.0
Spider-Man 2	22	2004.0
Spider-Man 3	8	2007.0
Star Wars: The Force Awakens	18	2015.0
Star Wars: The Last Jedi	20	2017.0
Star Wars: The Rise of Skywalker	19	2019.0
Tangled	10	2010.0
The Chronicles of Narnia: Prince Caspian	24	2008.0
The Fate of the Furious	29	2017.0
The Lion King	30	2019.0
Titanic	7	1997.0
Waterworld	13	1995.0
Wild Wild West	28	1999.0
X-Men: The Last Stand	25	2006.0

MAJOR PROJECT 1 (COMPULSORY)

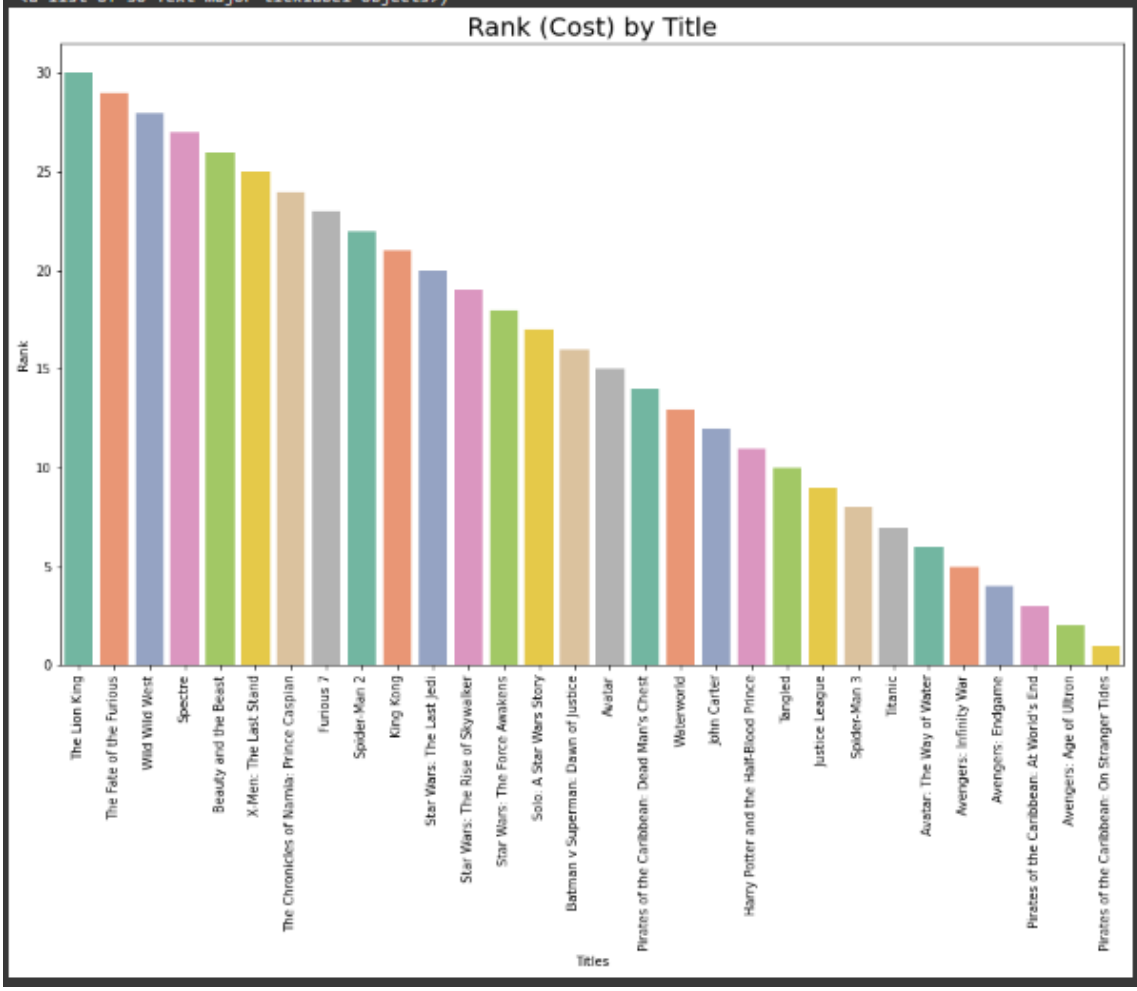
```
fig, ax1 = plt.subplots()
fig.set_size_inches(15, 9)


ax1 = sns.barplot(x ="Title", y ="Rank",
        data = df.sort_values('Est Cost (inmillions)'),
                    palette ="Set2")

ax1.set(xlabel ='Titles', ylabel ='Rank')
ax1.set_title('Rank (Cost) by Title', size = 20)

plt.xticks(rotation =90)
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]),
 <a list of 30 Text major ticklabel objects>)
```



MAJOR PROJECT 1 (COMPULSORY)

```
#Slicing
#Slice row indexes from 15 to 30 and column indexes 1 and 2.
df.iloc[15:31,1:3]
```

	Title	Year
15	Avatar	2009.0
16	Batman v Superman: Dawn of Justice	2016.0
17	Solo: A Star Wars Story	2018.0
18	Star Wars: The Force Awakens	2015.0
19	Star Wars: The Rise of Skywalker	2019.0
20	Star Wars: The Last Jedi	2017.0
21	King Kong	2005.0
22	Spider-Man 2	2004.0
23	Furious 7	2015.0
24	The Chronicles of Narnia: Prince Caspian	2008.0
25	X-Men: The Last Stand	2006.0
26	Beauty and the Beast	2017.0
27	Spectre	2015.0
28	Wild Wild West	1999.0
29	The Fate of the Furious	2017.0
30	The Lion King	2019.0

GITHUB ACCOUNT LINK :

https://github.com/RahulBhoo7