In [1]:
```
import pandas as pd
```

In [2]:
```
df=pd.read_csv('C:\\Users\\Priyo\\Downloads\\salary.csv')
df.head(5)
```

Out[2]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

In [3]:
```
df.dtypes
```

Out[3]:
```
Age                    float64
Gender                  object
Education Level         object
Job Title               object
Years of Experience    float64
Salary                 float64
dtype: object
```

In [4]:
```
df.shape
```

Out[4]:
```
(375, 6)
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
Age                    2
Gender                 2
Education Level        2
Job Title              2
Years of Experience    2
Salary                 2
dtype: int64
```

In [6]:
```
df.describe()
```

Out[6]:

	Age	Years of Experience	Salary
count	373.000000	373.000000	373.000000
mean	37.431635	10.030831	100577.345845
std	7.069073	6.557007	48240.013482
min	23.000000	0.000000	350.000000
25%	31.000000	4.000000	55000.000000
50%	36.000000	9.000000	95000.000000
75%	44.000000	15.000000	140000.000000
max	53.000000	25.000000	250000.000000

In [7]:
```
df['Education Level'].value_counts()
```

Out[7]:
```
Bachelor's    224
Master's       98
PhD            51
Name: Education Level, dtype: int64
```

In [8]:
```
df.columns
```

Out[8]:
```
Index(['Age', 'Gender', 'Education Level', 'Job Title', 'Years of Experience',
       'Salary'],
      dtype='object')
```

In [9]:
```
df['Salary'].fillna(df['Salary'].mean(),inplace=True)
df['Salary'].isnull().sum()
df['Age'].fillna(df['Age'].mean(),inplace=True)
df['Age'].isnull().sum()
df['Gender'].fillna('Male',inplace=True)
df['Gender'].isnull().sum()
df["Education Level"].fillna("Bachelor's",inplace=True)
df['Education Level'].isnull().sum()
df['Years of Experience'].fillna(df['Years of Experience'].mean(),inplace=True)
df['Years of Experience'].isnull().sum()
```

Out[9]:
```
0
```

In [10]:
```
df.head()
```

Out[10]:

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

In [11]:
```
df.pop('Job Title')
```

Out[11]:
```
0                  Software Engineer
1                       Data Analyst
2                     Senior Manager
3                    Sales Associate
4                           Director
                      ...
370          Senior Marketing Analyst
371            Director of Operations
372            Junior Project Manager
373    Senior Operations Coordinator
374           Senior Business Analyst
Name: Job Title, Length: 375, dtype: object
```

In [12]:
```
df.head()
```

Out[12]:

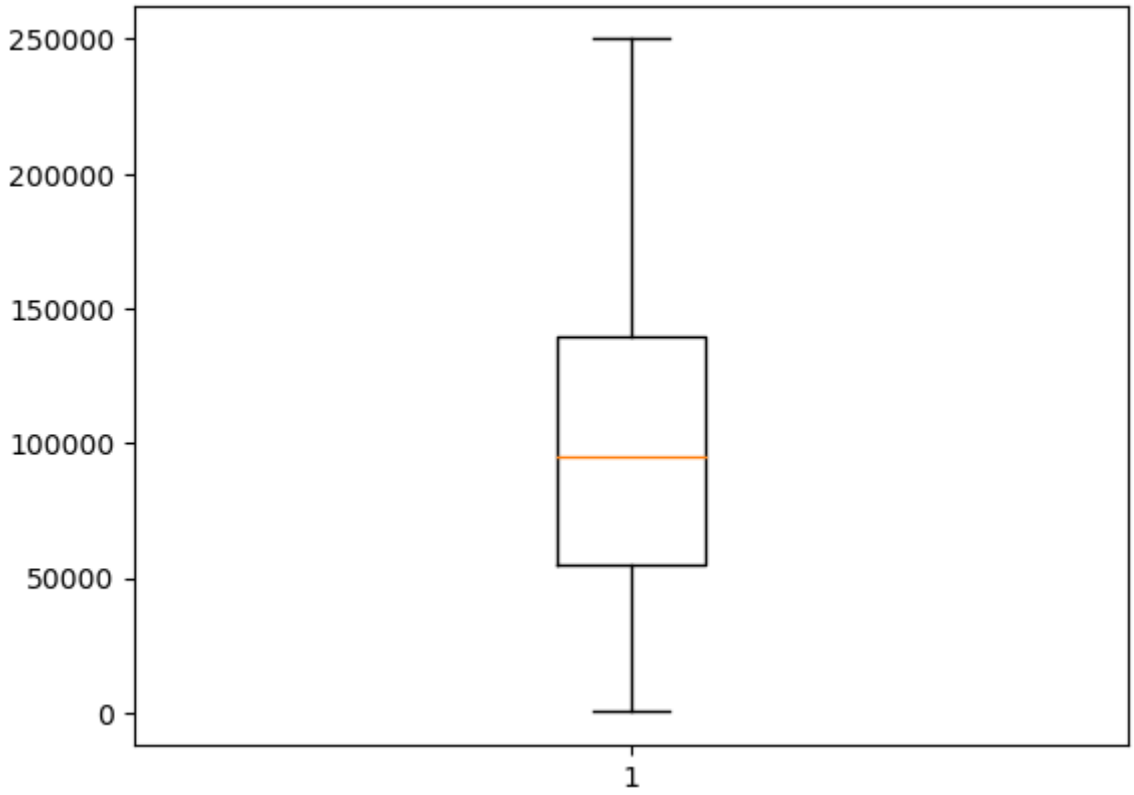
	Age	Gender	Education Level	Years of Experience	Salary
0	32.0	Male	Bachelor's	5.0	90000.0
1	28.0	Female	Master's	3.0	65000.0
2	45.0	Male	PhD	15.0	150000.0
3	36.0	Female	Bachelor's	7.0	60000.0
4	52.0	Male	Master's	20.0	200000.0

In [13]:
```
df.isnull().sum()
```

Out[13]:
```
Age                    0
Gender                 0
Education Level        0
Years of Experience    0
Salary                 0
dtype: int64
```

In [14]:
```
import matplotlib.pyplot as plt
plt.boxplot(df['Salary'])
```

Out[14]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1a90a5b6e60>,
  <matplotlib.lines.Line2D at 0x1a90a5b7100>],
 'caps': [<matplotlib.lines.Line2D at 0x1a90a5b73a0>,
  <matplotlib.lines.Line2D at 0x1a90a5b7640>],
 'boxes': [<matplotlib.lines.Line2D at 0x1a90a5b6bc0>],
 'medians': [<matplotlib.lines.Line2D at 0x1a90a5b78e0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1a90a5b7b80>],
 'means': []}
```



In [15]:
```
df.head(2)
```

Out[15]:

	Age	Gender	Education Level	Years of Experience	Salary
0	32.0	Male	Bachelor's	5.0	90000.0
1	28.0	Female	Master's	3.0	65000.0

In [16]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df['n_gender']=le.fit_transform(df['Gender'])
df['n_edu']=le.fit_transform(df['Education Level'])
```

In [17]:
```
df['n_edu'].value_counts()
```

Out[17]:
```
0    226
1     98
2     51
Name: n_edu, dtype: int64
```

In [18]:
```
df.columns
```

Out[18]:
```
Index(['Age', 'Gender', 'Education Level', 'Years of Experience', 'Salary',
       'n_gender', 'n_edu'],
      dtype='object')
```

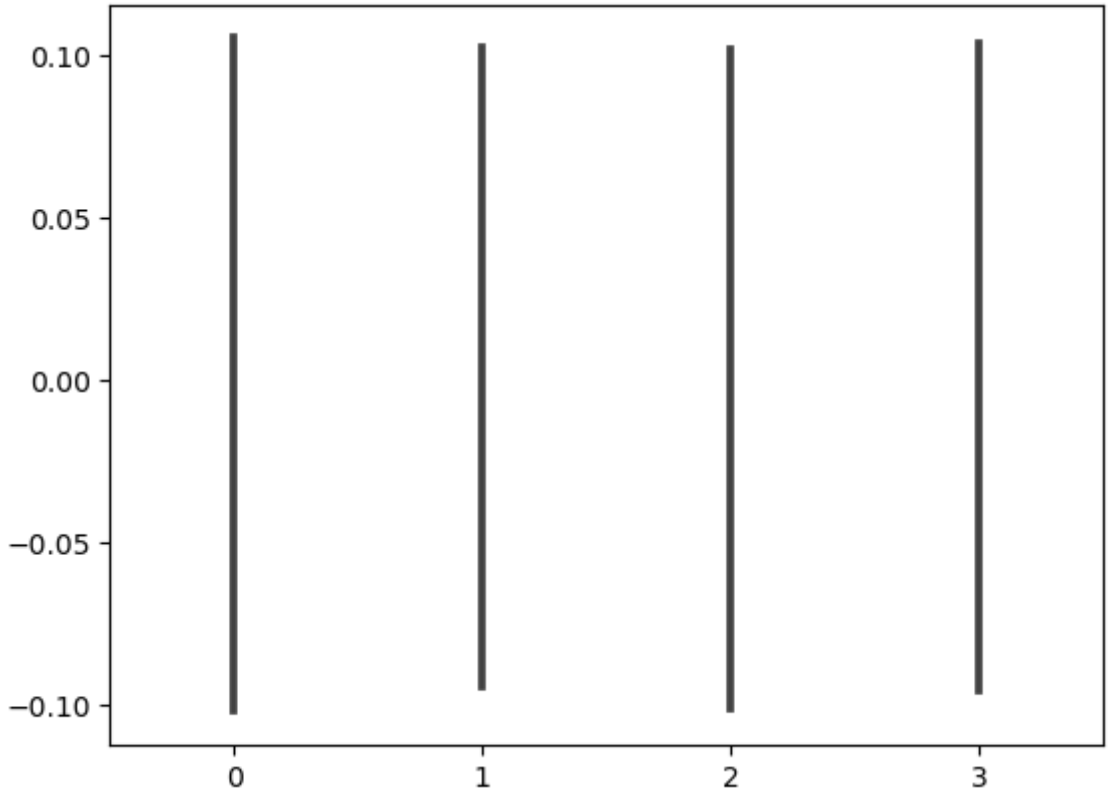
In [19]:
```
x=df[['Age','Years of Experience',
       'n_gender', 'n_edu']]
x.head()
```

Out[19]:

	Age	Years of Experience	n_gender	n_edu
0	32.0	5.0	1	0
1	28.0	3.0	0	1
2	45.0	15.0	1	2
3	36.0	7.0	0	0
4	52.0	20.0	1	1

In [20]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_sc=sc.fit_transform(x)
x_sc=pd.DataFrame(x_sc)
x_sc.head(2)
import seaborn as sns
sns.barplot(x_sc)
```

Out[20]:
```
<Axes: >
```



In [21]:
```
y=df['Salary']
y.head()
y.isnull().sum()
```

Out[21]:
```
0
```

In [22]:
```
x_sc.head(2)
x_sc.isnull().sum()
```

Out[22]:
```
0    0
1    0
2    0
3    0
dtype: int64
```

In [23]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test= train_test_split(x_sc,y,test_size=0.25)
```

In [24]:
```
from sklearn.linear_model import LinearRegression
le=LinearRegression()
```

In [25]:
```
model=le.fit(x_train,y_train)
pre=model.predict(x_test)
res=pd.DataFrame({'Actual':y_test,'Predic':pre})
res=pd.concat((x_sc,res),axis=1)
res=res[res['Actual']>=0]
res
```

Out[25]:

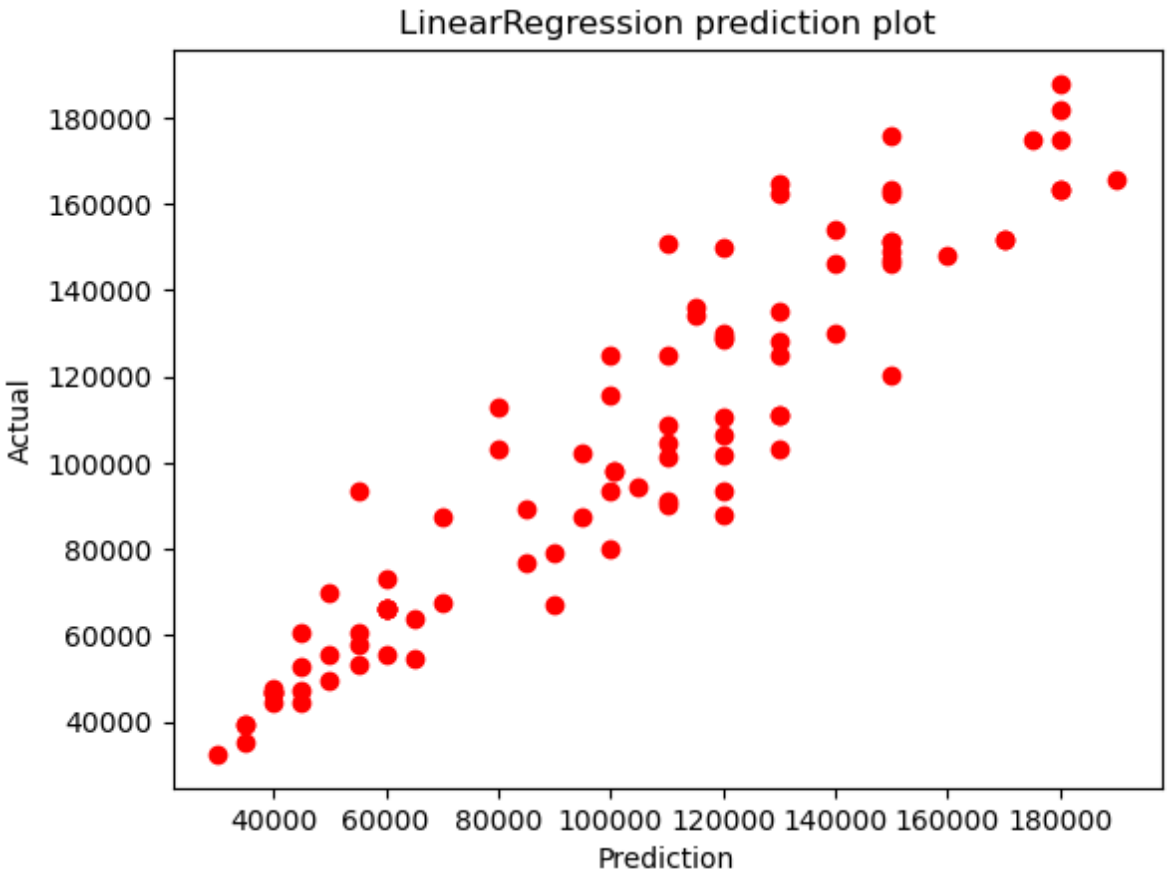
	0	1	2	3	Actual	Predic
0	-0.771458	-0.770333	0.955649	-0.738969	90000.0	67158.560494
1	-1.339580	-1.076577	-1.046409	0.646598	65000.0	54802.886037
9	0.080725	-0.004721	0.955649	2.032165	110000.0	125033.460367
17	0.222756	0.301524	0.955649	2.032165	115000.0	134329.109355
21	1.359000	1.373380	0.955649	0.646598	190000.0	165516.626195
...	...	...	...	...	...	...
353	1.501030	1.679625	0.955649	0.646598	180000.0	174812.275183
355	0.364786	0.301524	0.955649	-0.738969	130000.0	111192.188493
357	-0.629428	-0.923455	0.955649	-0.738969	60000.0	66343.688181
361	-0.487397	-0.464088	0.955649	-0.738969	90000.0	79009.510936
365	0.790878	1.220258	0.955649	0.646598	170000.0	151925.246613

94 rows × 6 columns

In [26]:
```
from sklearn.metrics import r2_score
r2_score(res['Actual'],res['Predic'])
```

Out[26]:
```
0.8746615354764269
```

In [27]:
```
plt.scatter(res['Actual'],res['Predic'],color='r')
plt.title('LinearRegression prediction plot')
plt.xlabel('Prediction')
plt.ylabel('Actual')
plt.show()
```


LinearRegression prediction plot
Actual
Prediction
180000
160000
140000
120000
100000
80000
60000
40000
40000
60000
80000
100000
120000
140000
160000
180000