```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
data=pd.read_csv('/content/student_data.csv')
print(data)
```

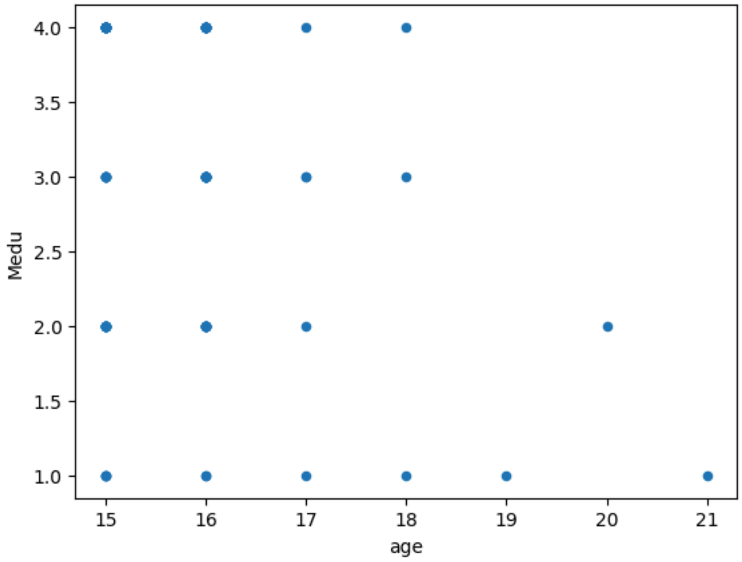
```
   school sex  age address famsize Pstatus  Medu  Fedu      Mjob      Fjob  \
0      GP   F   18       U     GT3       A     4     4   at_home   teacher   
1      GP   F   17       U     GT3       T     1     1   at_home     other   
2      GP   F   15       U     LE3       T     1     1   at_home     other   
3      GP   F   15       U     GT3       T     4     2    health  services   
4      GP   F   16       U     GT3       T     3     3     other     other   
..    ...  ..  ...     ...     ...     ...   ...   ...       ...       ...   
95     MS   M   20       U     LE3       A     2     2  services  services   
96     MS   M   17       U     LE3       T     3     1  services  services   
97     MS   M   21       R     GT3       T     1     1     other     other   
98     MS   M   18       R     LE3       T     3     2  services     other   
99     MS   M   19       U     LE3       T     1     1     other   at_home   

    ... famrel freetime  goout  Dalc  Walc health absences  G1  G2  G3  
0   ...      4        3      4     1     1      3        6   5   6   6  
1   ...      5        3      3     1     1      3        4   5   5   6  
2   ...      4        3      2     2     3      3       10   7   8  10  
3   ...      3        2      2     1     1      5        2  15  14  15  
4   ...      4        3      2     1     2      5        4   6  10  10  
..  ...    ...      ...    ...   ...   ...    ...      ...  ..  ..  ..  
95  ...      5        5      4     4     5      4       11   9   9   9  
96  ...      2        4      5     3     4      2        3  14  16  16  
97  ...      5        5      3     3     3      3        3  10   8   7  
98  ...      4        4      1     3     4      5        0  11  12  10  
99  ...      3        2      3     3     3      5        5   8   9   9  

[100 rows x 33 columns]
```

```
data.head()
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4

5 rows × 33 columns

```
data.plot.scatter(x='age', y='Medu');
```



```
data.shape
```

```
(100, 33)
```

```
x=data['age'].values.reshape(-1,1)
y=data['Medu'].values.reshape(-1,1)#depending
```

```
x.shape
```

```
(100, 1)
```

```
SEED = 30
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 30)
```

```
print(x_train)
print(y_train)
```

```
 [4]
 [3]
 [4]
 [4]
 [4]
 [4]
 [4]
 [2]
 [2]
 [4]
 [2]
 [2]
 [4]
 [4]
 [4]
 [4]
 [2]
 [3]
 [2]
 [4]
 [3]
 [2]
 [3]
 [4]
 [3]
 [2]
 [4]
 [2]
 [4]
 [1]
 [4]
 [3]
 [2]
 [3]
 [2]
 [4]
 [1]
 [4]
 [2]
 [3]
 [4]
 [1]
 [3]
 [4]
 [1]
 [2]
 [4]
 [4]
 [2]
 [3]
 [2]
 [1]
 [3]
 [4]
 [4]
 [4]
 [4]
 [4]
```

```
from sklearn.linear_model import LinearRegression
```

```
regressor = LinearRegression()
```

```
regressor.fit(x_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred = regressor.predict(x_test)
```

```
df_preds = pd.DataFrame({'Actual': y_test.squeeze(), 'Predicted': y_pred.squeeze()})
print(df_preds)
```

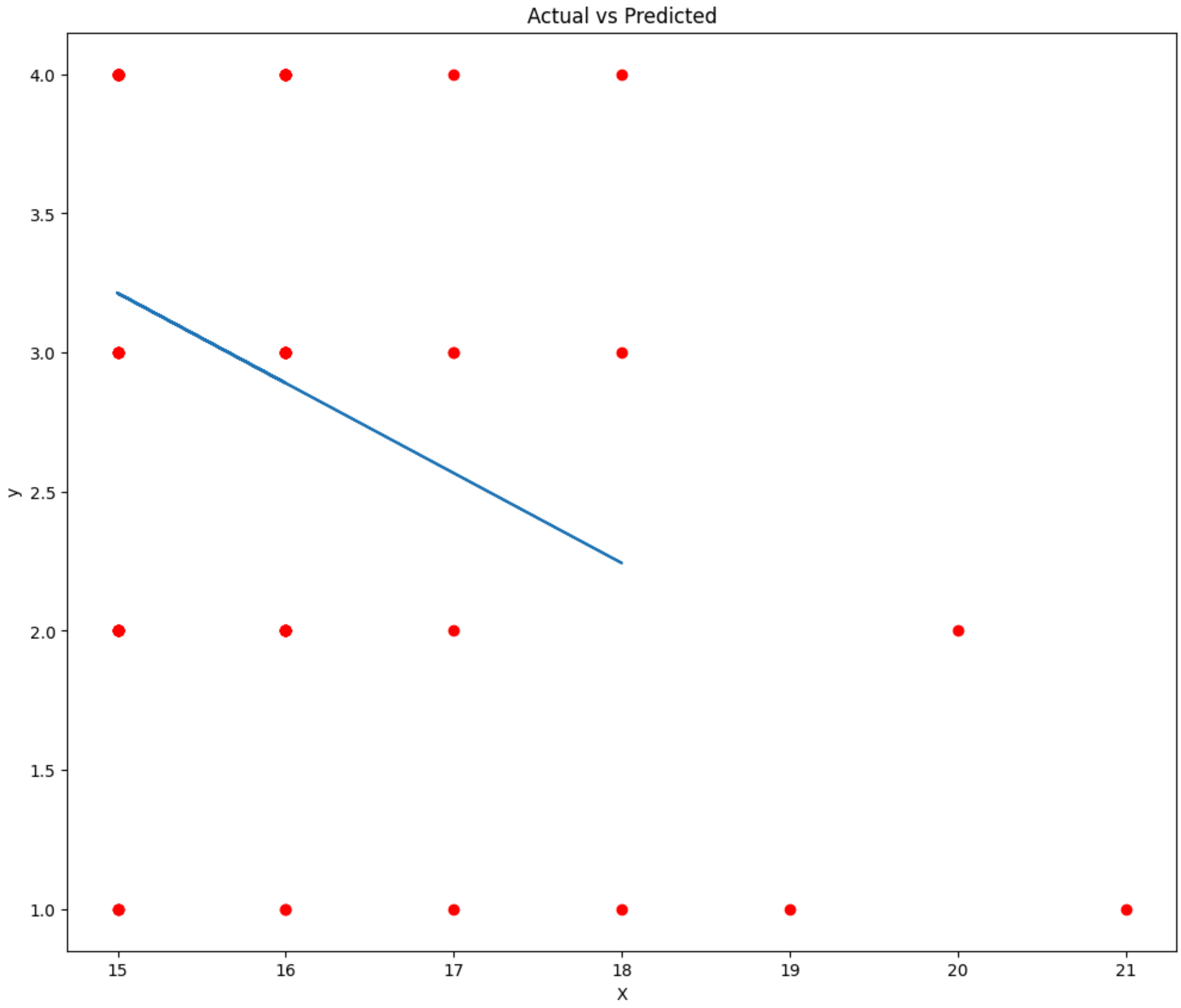
```
    Actual  Predicted
0        4   3.213119
1        4   3.213119
2        3   2.889740
3        4   3.213119
4        3   3.213119
5        3   2.889740
6        4   3.213119
7        2   2.889740
8        2   3.213119
9        4   2.889740
10       3   3.213119
11       3   3.213119
12       4   2.242981
13       2   2.889740
14       4   3.213119
15       3   2.889740
16       2   3.213119
17       3   2.889740
18       4   3.213119
19       4   3.213119
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
```

```
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Mean absolute error: 0.68
Mean squared error: 0.67
Root mean squared error: 0.82
```

```
plt.figure(figsize=(12,10))
plt.plot(x,y,'ro')     # regression line
plt.plot(x_test,y_pred)   # scatter plot showing actual data
plt.title('Actual vs Predicted')
plt.xlabel('X')
plt.ylabel('y')
plt.show()
```


Actual vs Predicted
4.0
3.5
3.0
2.5
2.0
1.5
1.0
y
15
16
17
18
19
20
21
X

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
data=pd.read_csv('/content/2014.csv')
print(data)
```

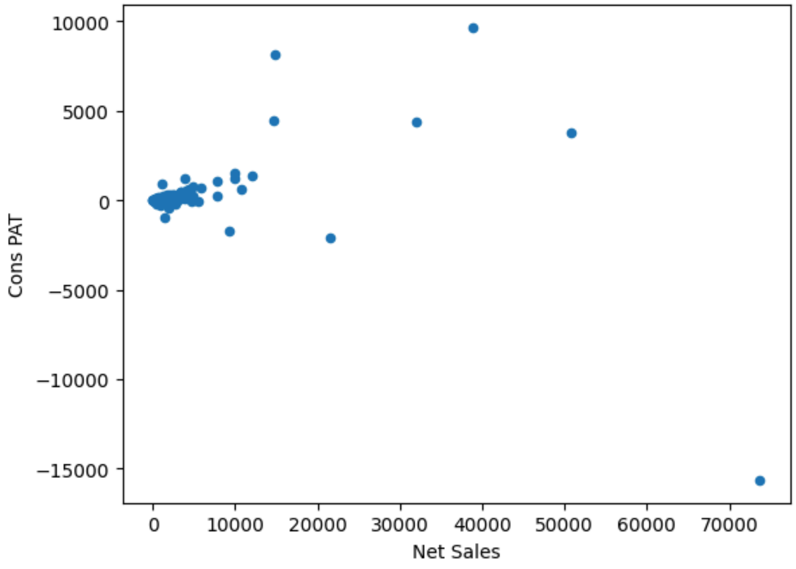
```
                Company Name  Net Sales  Cons PAT  Cons OCF
0            3I Infotech Ltd.       1344      -976       -51
1               3M India Ltd.       1840       108       177
2      Aavas Financiers Ltd.        104        19      -370
3              ABB India Ltd.       7733       229       479
4           Abbott India Ltd.       2289       229       215
..                        ...        ...       ...       ...
151          Wabco India Ltd.       1348       121       124
152   Whirlpool Of India Ltd.       3294       211       300
153              Wintac Ltd.          29        -6         6
154  Xchanging Solutions Ltd.        287        25         1
155     Zim Laboratories Ltd.        266       -22         6

[156 rows x 4 columns]
```

```
data.head()
```

	Company Name	Net Sales	Cons PAT	Cons OCF
0	3I Infotech Ltd.	1344	-976	-51
1	3M India Ltd.	1840	108	177
2	Aavas Financiers Ltd.	104	19	-370
3	ABB India Ltd.	7733	229	479
4	Abbott India Ltd.	2289	229	215

Next steps: Generate code with data View recommended plots

```
data.plot.scatter(x='Net Sales', y='Cons PAT');
```



```
data.shape
```

```
(156, 4)
```

```
x=data['Net Sales'].values.reshape(-1,1)
y=data['Cons PAT'].values.reshape(-1,1)#depending
```

```
x.shape
```

```
     (156, 1)
```

```
SEED = 30
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 30)
```

```
print(x_train)
print(y_train)
```

```
      [   -14]
      [   -60]
      [   229]
      [    -6]
      [    20]
      [    45]
      [     0]
      [   -71]
      [   158]
      [    18]
      [    22]
      [   149]
      [    35]
      [   114]
      [    22]
      [    66]
      [   -53]
      [    22]
      [    69]
      [    40]
      [     5]
      [  -459]
      [   943]
      [    88]
      [   153]
      [    68]
      [  1338]
      [    -3]
      [  1185]
      [  -262]
      [    25]
      [   129]
      [   -31]
      [   383]
      [  1066]
      [   160]
      [   247]
      [   268]
      [   186]
      [   -56]
      [   264]
      [    34]
      [   -22]
      [    32]
      [    79]
      [   301]
      [    42]
      [    47]
      [   584]
      [   113]
      [     5]
      [   196]
      [   177]
      [    42]
      [   -51]
      [  1487]
      [    70]
      [   103]
```

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
```

```
regressor.fit(x_train, y_train)
```

```
     ▾ LinearRegression
     LinearRegression()
```

```
y_pred = regressor.predict(x_test)
```

```
df_preds = pd.DataFrame({'Actual': y_test.squeeze(), 'Predicted': y_pred.squeeze()})
print(df_preds)
```

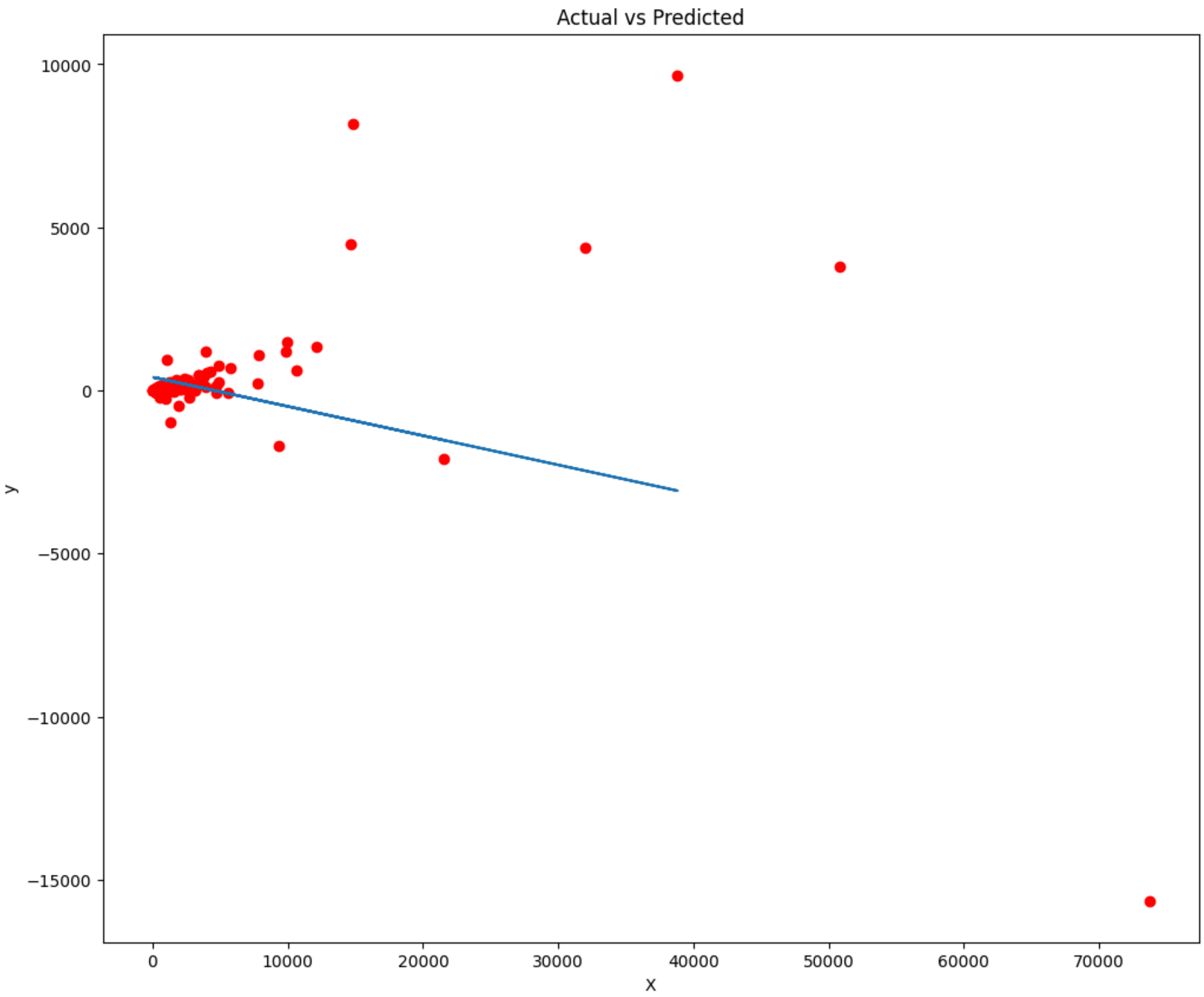
```
    Actual    Predicted
0        4   326.274229
1     9663 -3075.930497
2     4363 -2460.486468
3       37   338.470118
4        7   396.221241
5       33   361.965140
6       67   379.451893
7        6   396.759295
8        1   389.047188
9      229  -286.838234
10      19   223.685278
11    -233   361.696113
12      49   382.590541
13     -67   -15.390022
14      33   385.191135
15     274    84.957038
16   -2097 -1526.873210
17     211   111.232005
18     133   256.685920
19     108   241.620409
20      78   265.563810
21       7   401.063726
22       7   394.696755
23       8   395.952214
24      12   397.476700
25    -976   286.099535
26      78   348.513792
27      11   298.923154
28     114   191.043339
29      53   378.824163
30      35   343.312604
31      19   397.297349
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
```

```
print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Mean absolute error: 931.79
Mean squared error: 6679322.02
Root mean squared error: 2584.44
```

```
plt.figure(figsize=(12,10))
plt.plot(x,y,'ro')     # regression line
plt.plot(x_test,y_pred)   # scatter plot showing actual data
plt.title('Actual vs Predicted')
plt.xlabel('X')
plt.ylabel('y')
plt.show()
```


Actual vs Predicted
10000
5000
0
−5000
−10000
−15000
y
0
10000
20000
30000
40000
50000
60000
70000
X