Multiple Linear Regression

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score,accuracy_score
```

CCPP Dataset

In [2]:
```
df = pd.read_csv("E:\MCA\sem_3\ML_Lab\programs\week4\ccpp.csv")
df.head()
```

Out[2]:

	AT	V	AP	RH	PE
0	8.34	40.77	1010.84	90.01	480.48
1	23.64	58.49	1011.40	74.20	445.75
2	29.74	56.90	1007.15	41.91	438.76
3	19.07	49.69	1007.22	76.79	453.09
4	11.80	40.66	1017.13	97.20	464.43

In [3]:
```
df.isnull().sum()
```

Out[3]:
```
AT    0
V     0
AP    0
RH    0
PE    0
dtype: int64
```

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

Out[4]:
```
(9568, 5)
```

In [5]:
```
x = df.drop(['PE'],axis = 1).values
x
```

Out[5]:
```
array([[   8.34,   40.77, 1010.84,   90.01],
       [  23.64,   58.49, 1011.4 ,   74.2 ],
       [  29.74,   56.9 , 1007.15,   41.91],
       ...,
       [  15.99,   43.34, 1014.2 ,   78.66],
       [  17.65,   59.87, 1018.58,   94.65],
       [  23.68,   51.3 , 1011.86,   71.24]])
```

In [6]:
```
y = df['PE'].values
y
```

Out[6]:
```
array([480.48, 445.75, 438.76, ..., 465.96, 450.93, 451.67])
```

In [7]:
```
x = pd.DataFrame(x)
x
```

Out[7]:

	0	1	2	3
0	8.34	40.77	1010.84	90.01
1	23.64	58.49	1011.40	74.20
2	29.74	56.90	1007.15	41.91
3	19.07	49.69	1007.22	76.79
4	11.80	40.66	1017.13	97.20
...	...	...	...	...
9563	15.12	48.92	1011.80	72.93
9564	33.41	77.95	1010.30	59.72
9565	15.99	43.34	1014.20	78.66
9566	17.65	59.87	1018.58	94.65
9567	23.68	51.30	1011.86	71.24

9568 rows × 4 columns

In [8]:
```
y = pd.DataFrame(y)
y
```

Out[8]:

	0
0	480.48
1	445.75
2	438.76
3	453.09
4	464.43
...	...
9563	462.59
9564	432.90
9565	465.96
9566	450.93
9567	451.67

9568 rows × 1 columns

Iteration 1

i. test_size=0.3 random_state=0

In [9]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.3,random_state = 0
```

In [10]:
```
model= LinearRegression()
```

In [11]:
```
model.fit(x_train,y_train)
```

Out[11]:
```
LinearRegression
LinearRegression()
```

In [12]:
```
y_predict = model.predict(x_test)
```

In [13]:
```
#test_data
model.predict([[14.96,41.76,1024.07,73.17]])
```

Out[13]:
```
array([[467.35592062]])
```

In [14]:
```
mse = mean_squared_error(y_test, y_predict)
r2 = r2_score(y_test, y_predict)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 20.60
R-squared: 0.93
```

In [15]:
```
model.score(x,y)
```

Out[15]:
```
0.9286890531329729
```

In [16]:
```
y_test = np.array(y_test).flatten()
y_predict = np.array(y_predict).flatten()
pred_y_df = pd.DataFrame({"Actual Values":y_test, "predicted values": y_predict, "Differ
pred_y_df
```

Out[16]:

	Actual Values	predicted values	Difference
0	426.18	433.332595	-7.152595
1	451.10	448.379693	2.720307
2	442.87	445.912478	-3.042478
3	443.70	445.986055	-2.286055
4	460.59	462.822292	-2.232292
...	...	...	...
2866	440.27	443.739019	-3.469019
2867	438.42	442.129053	-3.709053
2868	442.88	444.058091	-1.178091
2869	456.64	459.347457	-2.707457
2870	478.19	472.915320	5.274680

2871 rows × 3 columns

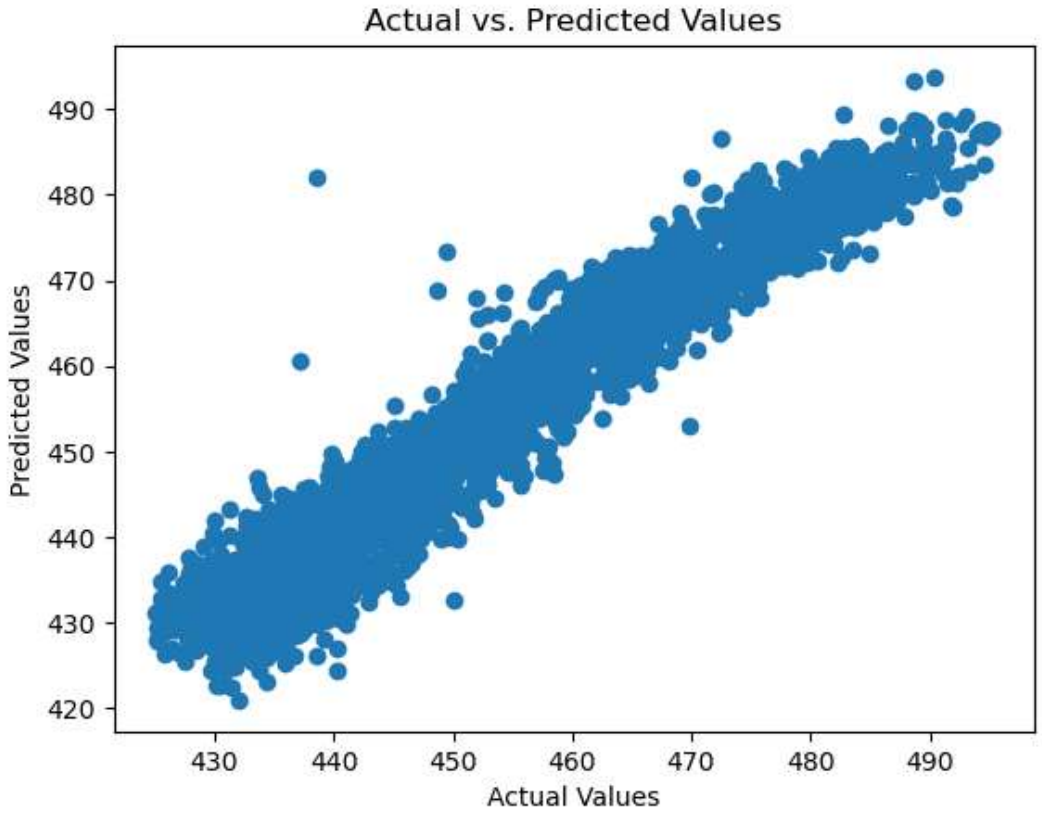
In [17]:
```
model.coef_
```

Out[17]:
```
array([[-1.9797936 , -0.23300225,  0.06812315, -0.15839461]])
```

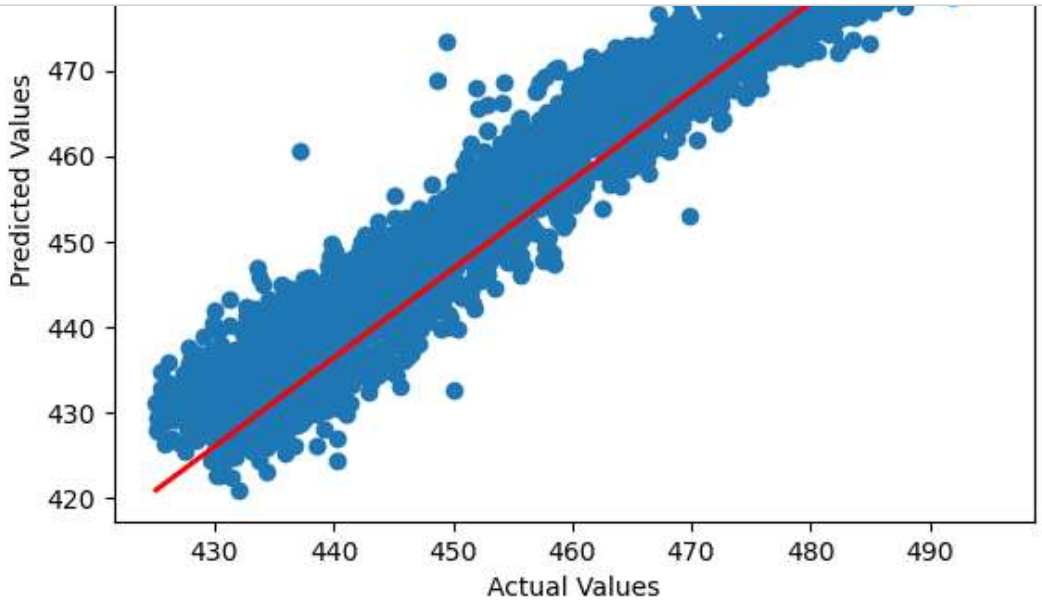
In [18]:
```
model.intercept_
```

Out[18]:
```
array([448.53067141])
```

In [19]:
```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [20]:
```
plt.scatter(y_test, y_predict)
plt.plot([min(y_test), max(y_test)], [min(y_predict), max(y_predict)], color='red', linew
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



ii. test_size=0.3 random_state=42

In [21]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.3,random_state = 4
```

In [22]:
```
model= LinearRegression()
```

In [23]:
```
model.fit(x_train,y_train)
```

Out[23]:
```
LinearRegression
LinearRegression()
```

In [24]:
```
y_predict = model.predict(x_test)
```

In [25]:
```
mse = mean_squared_error(y_test, y_predict)
r2 = r2_score(y_test, y_predict)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 20.00
R-squared: 0.93
```

In [26]:
```
model.score(x,y)
```

Out[26]:
```
0.9286920827769402
```

In [27]:
```
y_test = np.array(y_test).flatten()
y_predict = np.array(y_predict).flatten()
pred_y_df = pd.DataFrame({"Actual Values":y_test, "predicted values": y_predict, "Differ
pred_y_df
```

Out[27]:

	Actual Values	predicted values	Difference
0	433.27	435.844874	-2.574874
1	438.16	437.447577	0.712423
2	458.42	461.252149	-2.832149
3	480.82	476.059882	4.760118
4	441.41	435.811718	5.598282
...	...	...	...
2866	480.31	480.039097	0.270903
2867	446.77	453.837232	-7.067232
2868	454.66	450.978874	3.681126
2869	483.77	483.406809	0.363191
2870	441.51	435.479268	6.030732

2871 rows × 3 columns

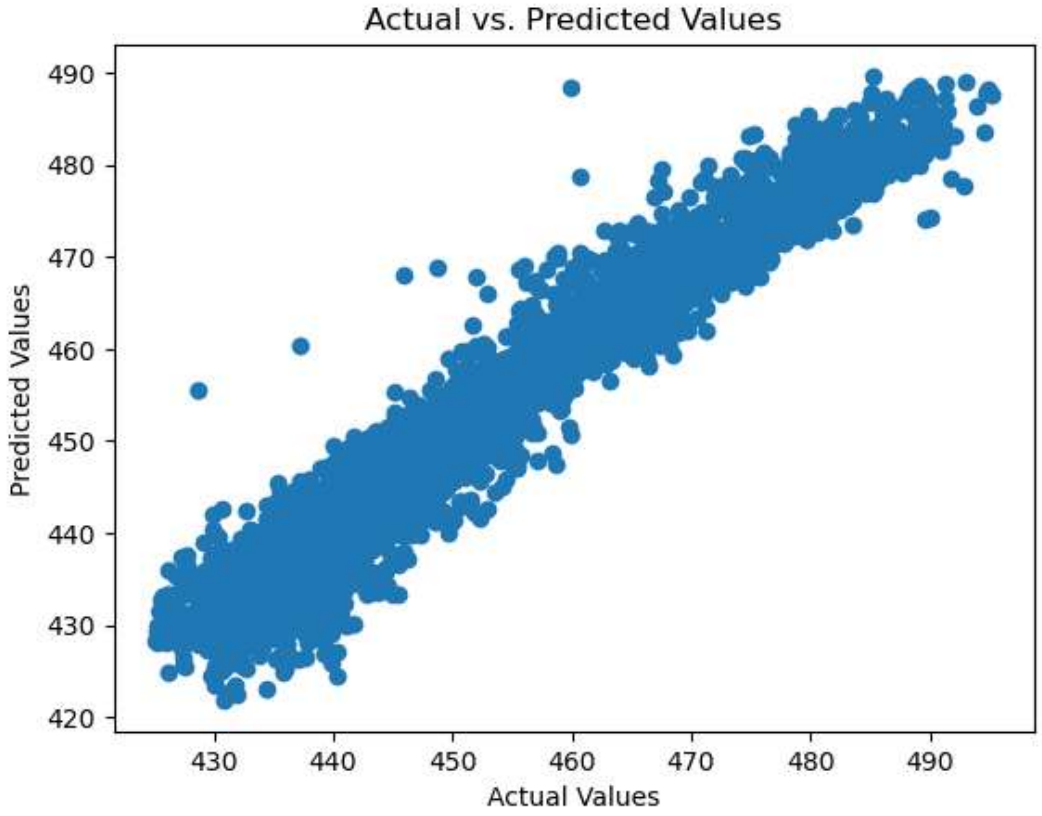
In [28]:
```
model.coef_
```

Out[28]:
```
array([[-1.98460442, -0.23146881,  0.05596081, -0.15973814]])
```

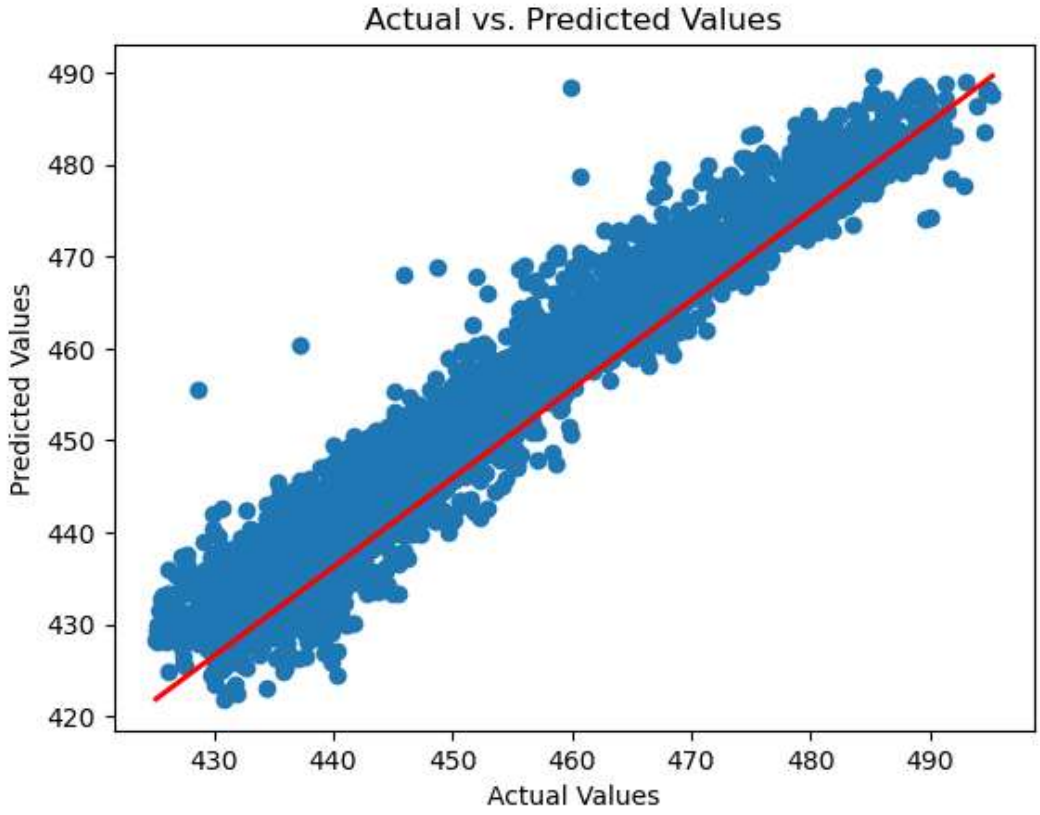
In [29]:
```
model.intercept_
```

Out[29]:
```
array([460.94747177])
```

In [30]:
```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [31]:
```
plt.scatter(y_test, y_predict)
plt.plot([min(y_test), max(y_test)], [min(y_predict), max(y_predict)], color='red', linew
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



Iteration 2

i. test_size=0.4 random_state=0

In [32]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.4,random_state = 0
```

In [33]:
```
model= LinearRegression()
```

In [34]:
```
model.fit(x_train,y_train)
```

Out[34]:
```
LinearRegression
LinearRegression()
```

In [35]:
```
y_predict = model.predict(x_test)
```

In [36]:
```
mse = mean_squared_error(y_test, y_predict)
r2 = r2_score(y_test, y_predict)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 20.69
R-squared: 0.93
```

In [37]:
```
model.score(x,y)
```

Out[37]:
```
0.9286867872621933
```

In [38]:
```
y_test = np.array(y_test).flatten()
y_predict = np.array(y_predict).flatten()
pred_y_df = pd.DataFrame({"Actual Values":y_test, "predicted values": y_predict, "Differ
pred_y_df
```

Out[38]:

	Actual Values	predicted values	Difference
0	426.18	433.338695	-7.158695
1	451.10	448.381768	2.718232
2	442.87	445.984692	-3.114692
3	443.70	445.996297	-2.296297
4	460.59	462.790519	-2.200519
...	...	...	...
3823	434.21	440.100327	-5.890327
3824	431.19	431.972177	-0.782177
3825	450.86	448.296619	2.563381
3826	456.44	452.628175	3.811825
3827	467.21	469.691145	-2.481145

3828 rows × 3 columns

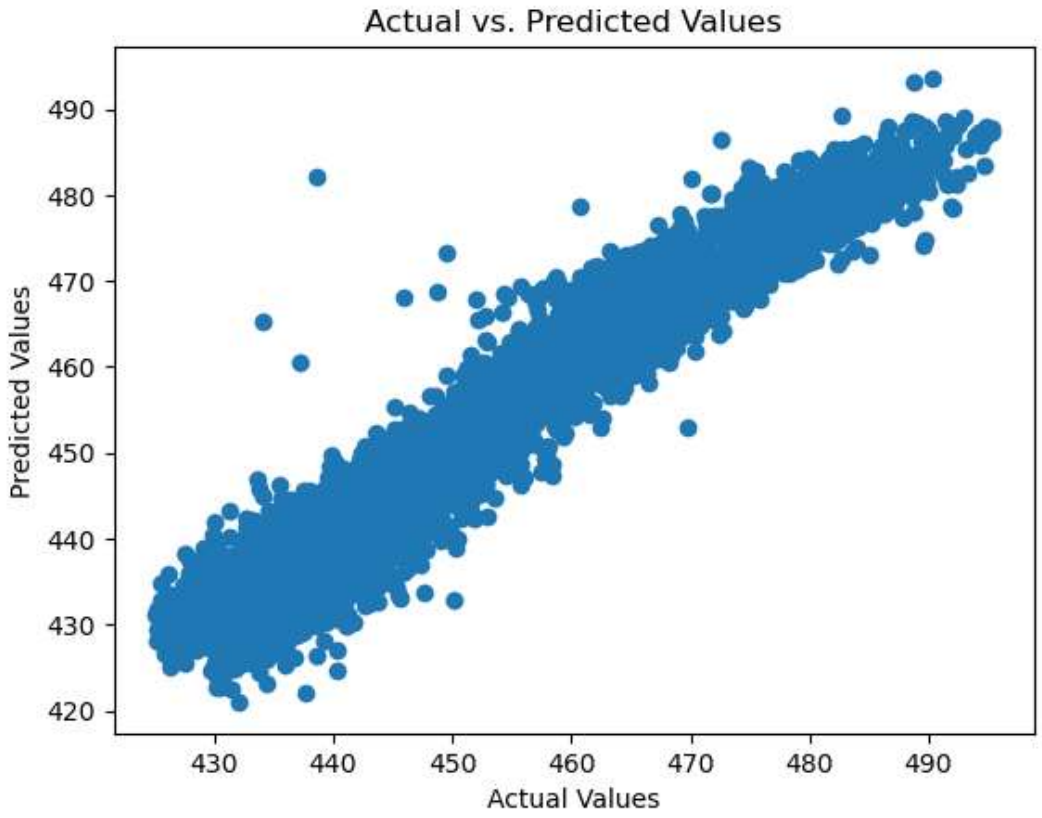
In [39]:
```
model.coef_
```

Out[39]:
```
array([[-1.96843024, -0.23826082,  0.0676831 , -0.15907504]])
```

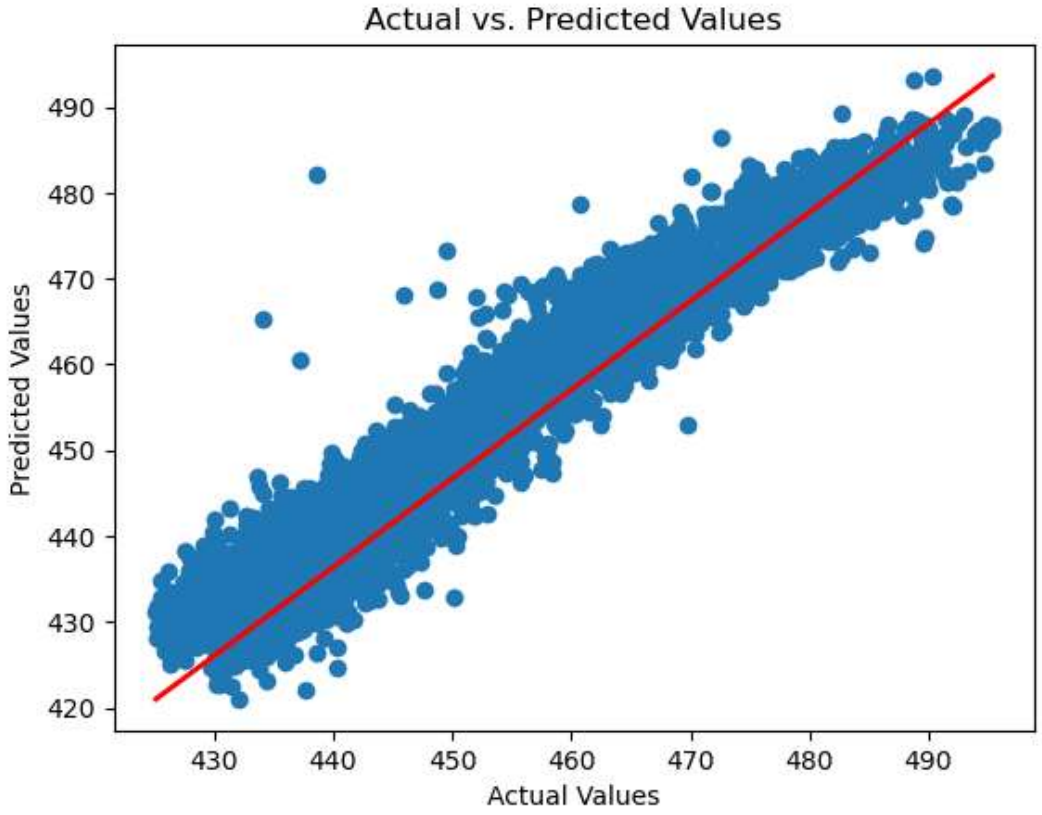
In [40]:
```
model.intercept_
```

Out[40]:
```
array([449.08571405])
```

In [41]:
```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [42]:
```
plt.scatter(y_test, y_predict)
plt.plot([min(y_test), max(y_test)], [min(y_predict), max(y_predict)], color='red', linew
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



ii.test_size=0.4 random_state=42

In [43]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.4,random_state = 4
```

In [44]:
```
model= LinearRegression()
```

In [45]:
```
model.fit(x_train,y_train)
```

Out[45]:
```
LinearRegression
LinearRegression()
```

In [46]:
```
y_predict = model.predict(x_test)
```

In [47]:
```
mse = mean_squared_error(y_test, y_predict)
r2 = r2_score(y_test, y_predict)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 20.28
R-squared: 0.93
```

In [48]:
```
model.score(x,y)
```

Out[48]:
```
0.9286932753846601
```

In [49]:
```
y_test = np.array(y_test).flatten()
y_predict = np.array(y_predict).flatten()
pred_y_df = pd.DataFrame({"Actual Values":y_test, "predicted values": y_predict, "Differ
pred_y_df
```

Out[49]:

	Actual Values	predicted values	Difference
0	433.27	435.785672	-2.515672
1	438.16	437.460918	0.699082
2	458.42	461.318754	-2.898754
3	480.82	476.151231	4.668769
4	441.41	435.825833	5.584167
...	...	...	...
3823	484.04	476.298532	7.741468
3824	438.54	440.974779	-2.434779
3825	441.90	440.366504	1.533496
3826	428.27	429.717123	-1.447123
3827	447.42	447.873832	-0.453832

3828 rows × 3 columns

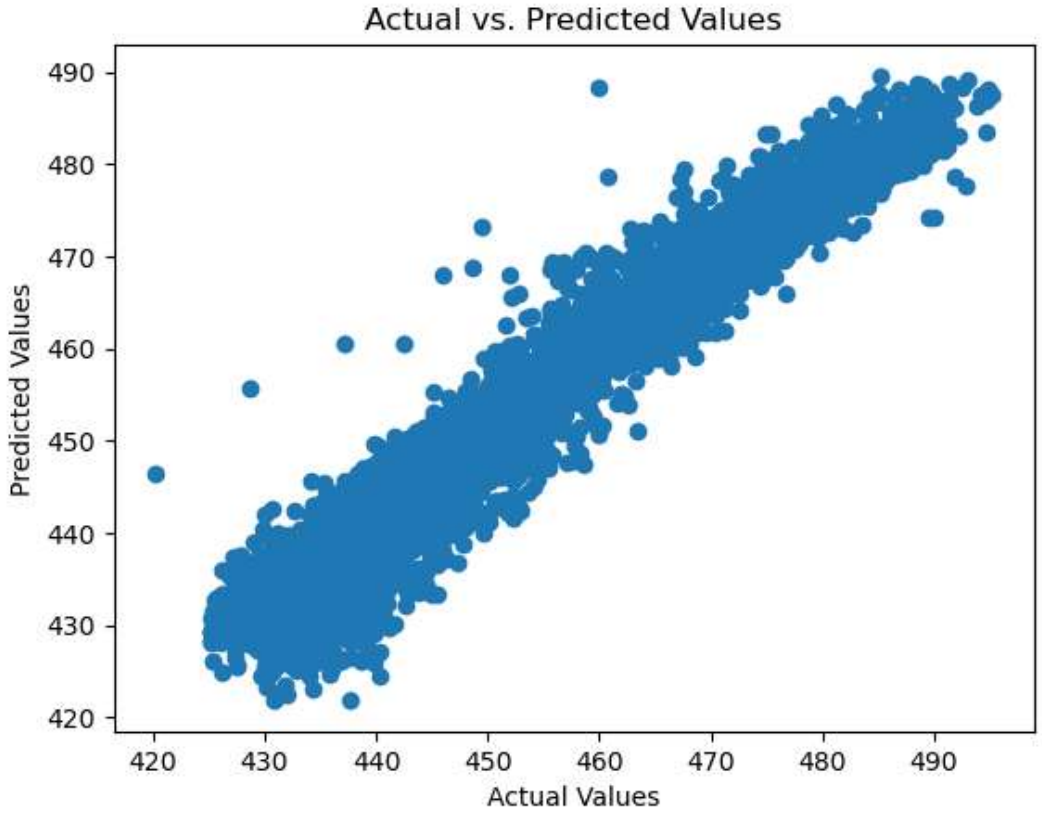
In [50]:
```
model.coef_
```

Out[50]:
```
array([[-1.98272338, -0.2301616 ,  0.06312151, -0.15781872]])
```

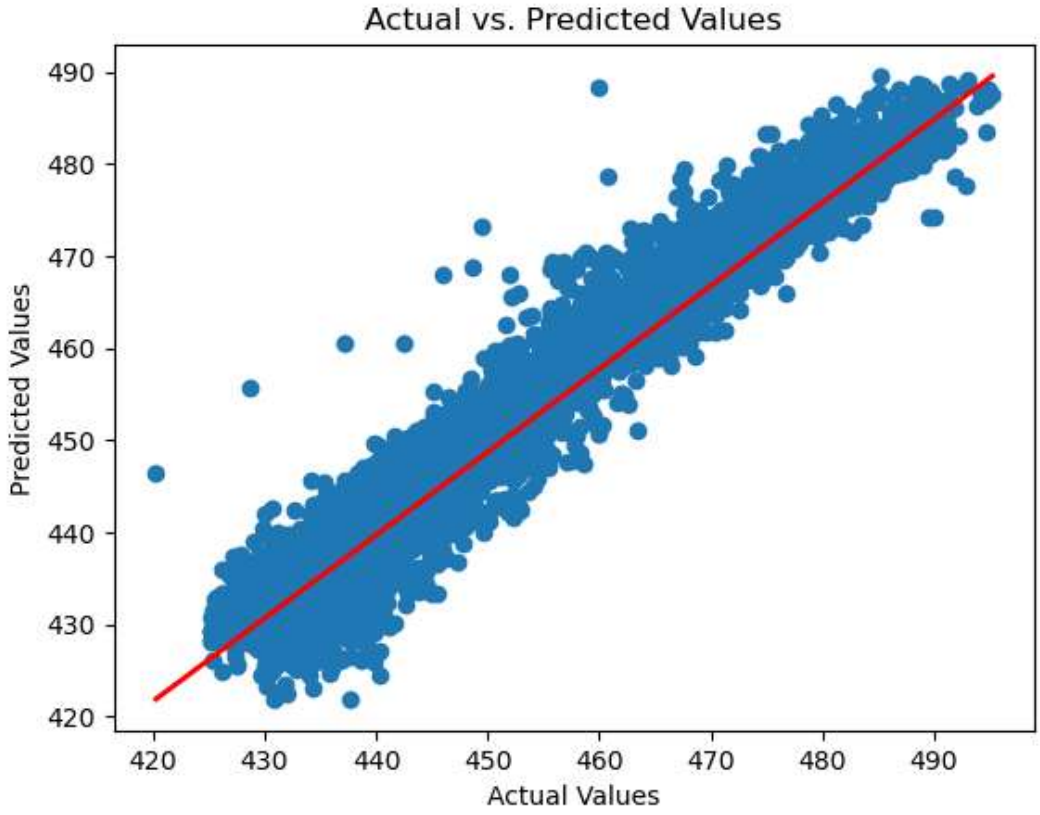
In [51]:
```
model.intercept_
```

Out[51]:
```
array([453.44634076])
```

In [52]:
```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [53]:
```
plt.scatter(y_test, y_predict)
plt.plot([min(y_test), max(y_test)], [min(y_predict), max(y_predict)], color='red', linew
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



Iteration 3

test_size=0.5 random_state=0

In [54]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.5,random_state = 0
```

In [55]:
```
model= LinearRegression()
```

In [56]:
```
model.fit(x_train,y_train)
```

Out[56]:
```
▾ LinearRegression
LinearRegression()
```

In [57]:
```
y_predict = model.predict(x_test)
```

In [58]:
```
mse = mean_squared_error(y_test, y_predict)
r2 = r2_score(y_test, y_predict)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 20.33
R-squared: 0.93
```

In [59]:
```
model.score(x,y)
```

Out[59]:
```
0.9286328322723267
```

In [60]:
```
y_test = np.array(y_test).flatten()
y_predict = np.array(y_predict).flatten()
pred_y_df = pd.DataFrame({"Actual Values":y_test, "predicted values": y_predict, "Differ
pred_y_df
```

Out[60]:

	Actual Values	predicted values	Difference
0	426.18	433.225912	-7.045912
1	451.10	448.368325	2.731675
2	442.87	446.055372	-3.185372
3	443.70	445.989381	-2.289381
4	460.59	462.727582	-2.137582
...	...	...	...
4779	470.07	469.438402	0.631598
4780	460.87	464.333831	-3.463831
4781	444.90	440.589084	4.310916
4782	465.63	466.204272	-0.574272
4783	430.32	433.461686	-3.141686

4784 rows × 3 columns

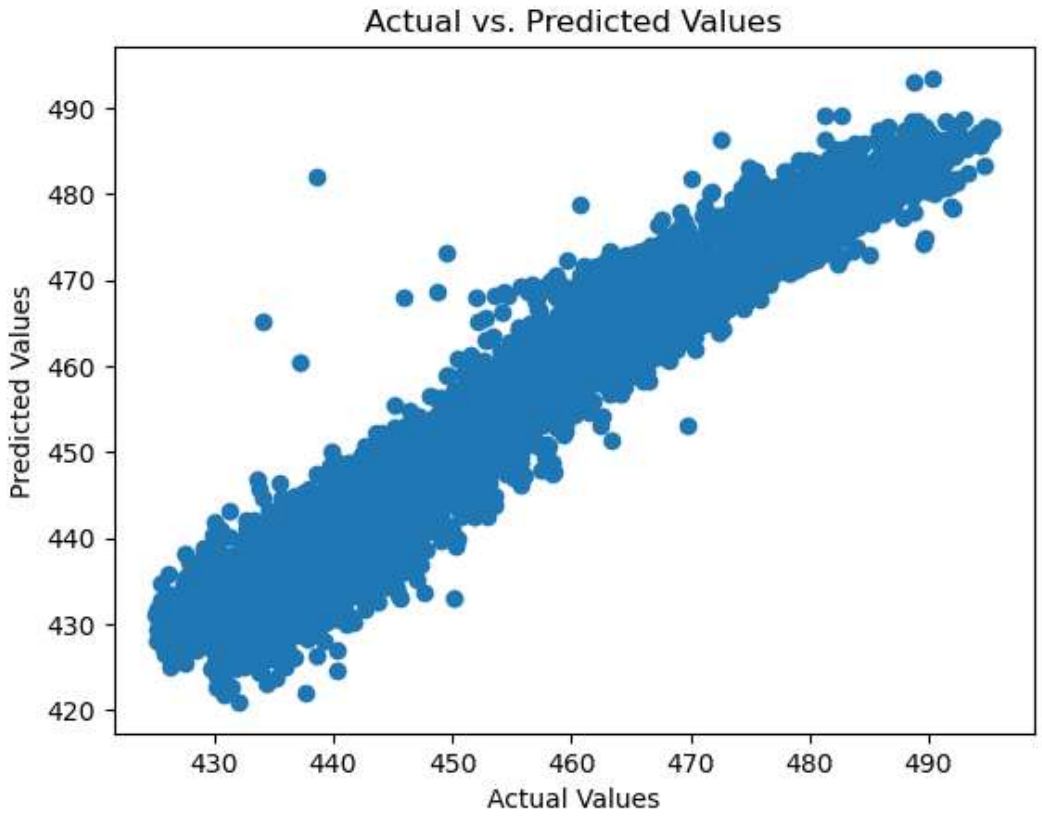
In [61]:
```
model.coef_
```

Out[61]:
```
array([[-1.94756722, -0.25103187,  0.07246607, -0.15942899]])
```

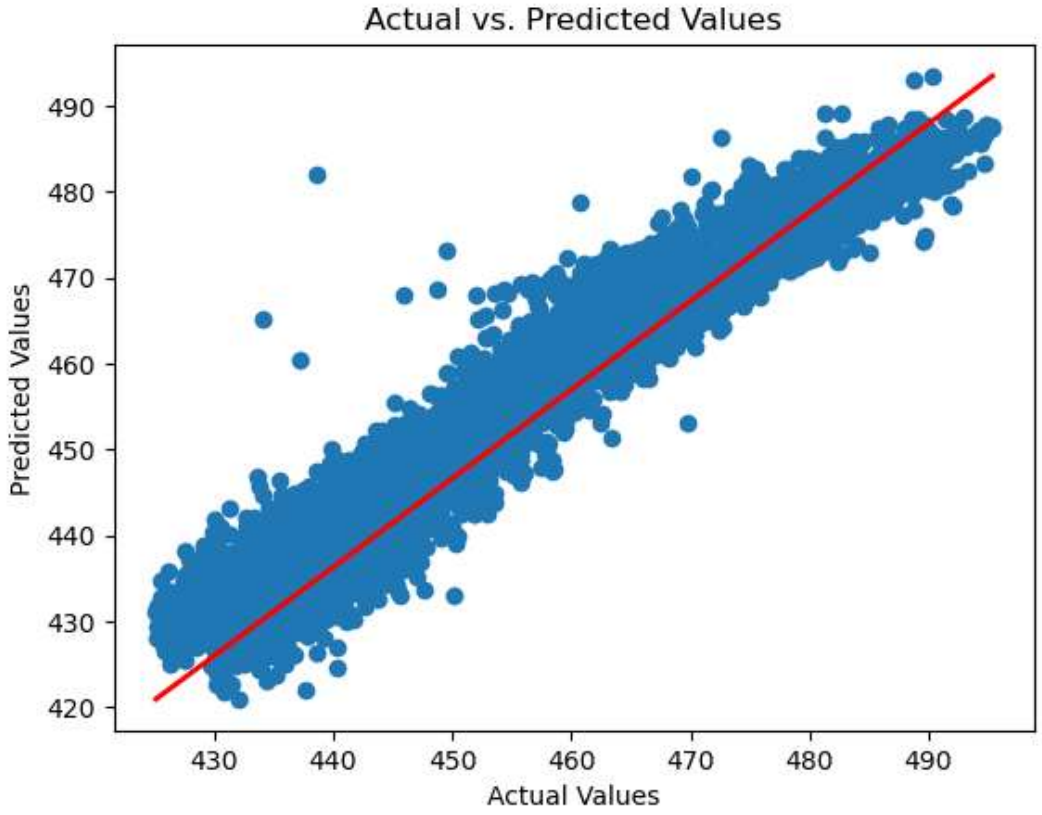
In [62]:
```
model.intercept_
```

Out[62]:
```
array([444.52367365])
```

In [63]:
```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [64]:
```
plt.scatter(y_test, y_predict)
plt.plot([min(y_test), max(y_test)], [min(y_predict), max(y_predict)], color='red', linew
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



ii.test_size=0.5 random_state=42

In [65]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.5,random_state = 4
```

In [66]:
```
model= LinearRegression()
```

In [67]:
```
model.fit(x_train,y_train)
```

Out[67]:
```
LinearRegression
LinearRegression()
```

In [68]:
```
y_predict = model.predict(x_test)
```

In [69]:
```
mse = mean_squared_error(y_test, y_predict)
r2 = r2_score(y_test, y_predict)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 21.16
R-squared: 0.93
```

In [70]:
```
model.score(x,y)
```

Out[70]:
```
0.9286714393010711
```

In [71]:
```
y_test = np.array(y_test).flatten()
y_predict = np.array(y_predict).flatten()
pred_y_df = pd.DataFrame({"Actual Values":y_test, "predicted values": y_predict, "Differ
pred_y_df
```

Out[71]:

	Actual Values	predicted values	Difference
0	433.27	435.655150	-2.385150
1	438.16	437.401889	0.758111
2	458.42	461.231409	-2.811409
3	480.82	476.159251	4.660749
4	441.41	435.673627	5.736373
...	...	...	...
4779	448.65	444.594920	4.055080
4780	464.17	468.026056	-3.856056
4781	436.55	439.304387	-2.754387
4782	471.87	471.869121	0.000879
4783	440.58	440.818089	-0.238089

4784 rows × 3 columns

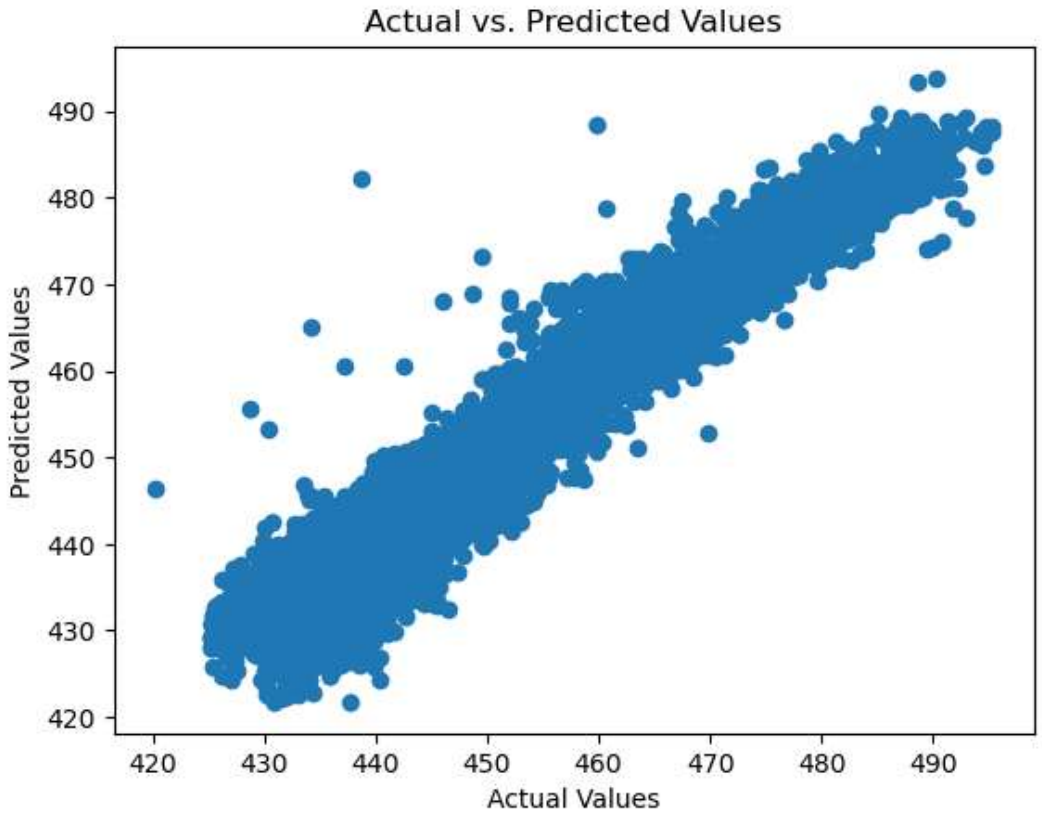
In [72]:
```
model.coef_
```

Out[72]:
```
array([[-1.99900652, -0.22689299,  0.05832641, -0.15905849]])
```

In [73]:
```
model.intercept_
```

Out[73]:
```
array([458.50516928])
```

In [74]:
```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [75]:
```
plt.scatter(y_test, y_predict)
plt.plot([min(y_test), max(y_test)], [min(y_predict), max(y_predict)], color='red', linew
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```