# 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
```

## Boston House Price Dataset

In [2]:

```
df = pd.read_csv("E:\MCA\sem_3\ML_Lab\programs\week4\\boston_house_prices.csv")
df.head()
```

Out[2]:

| | CRIM | ZN | INDUS | CHAS | NOX | RM | AGE | DIS | RAD | TAX | PTRATIO | B | LSTAT | MEDV |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.00632 | 18.0 | 2.31 | 0 | 0.538 | 6.575 | 65.2 | 4.0900 | 1 | 296 | 15.3 | 396.90 | 4.98 | 24.0 |
| **1** | 0.02731 | 0.0 | 7.07 | 0 | 0.469 | 6.421 | 78.9 | 4.9671 | 2 | 242 | 17.8 | 396.90 | 9.14 | 21.6 |
| **2** | 0.02729 | 0.0 | 7.07 | 0 | 0.469 | 7.185 | 61.1 | 4.9671 | 2 | 242 | 17.8 | 392.83 | 4.03 | 34.7 |
| **3** | 0.03237 | 0.0 | 2.18 | 0 | 0.458 | 6.998 | 45.8 | 6.0622 | 3 | 222 | 18.7 | 394.63 | 2.94 | 33.4 |
| **4** | 0.06905 | 0.0 | 2.18 | 0 | 0.458 | 7.147 | 54.2 | 6.0622 | 3 | 222 | 18.7 | 396.90 | 5.33 | 36.2 |

In [3]:

```
df.isnull().sum()
```

Out[3]:

```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
MEDV       0
dtype: int64
```

In [4]:

```
df.shape
```

Out[4]:

```
(506, 14)
```

In [5]:

```
x = df.drop(['MEDV'],axis = 1).values
x
```

Out[5]:

```
array([[6.3200e-03, 1.8000e+01, 2.3100e+00, ..., 1.5300e+01, 3.9690e+02,
        4.9800e+00],
       [2.7310e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9690e+02,
        9.1400e+00],
       [2.7290e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9283e+02,
        4.0300e+00],
       ...,
       [6.0760e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
        5.6400e+00],
       [1.0959e-01, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9345e+02,
        6.4800e+00],
       [4.7410e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
        7.8800e+00]])
```

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

Out[6]:
```
array([24. , 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9, 15. ,
       18.9, 21.7, 20.4, 18.2, 19.9, 23.1, 17.5, 20.2, 18.2, 13.6, 19.6,
       15.2, 14.5, 15.6, 13.9, 16.6, 14.8, 18.4, 21. , 12.7, 14.5, 13.2,
       13.1, 13.5, 18.9, 20. , 21. , 24.7, 30.8, 34.9, 26.6, 25.3, 24.7,
       21.2, 19.3, 20. , 16.6, 14.4, 19.4, 19.7, 20.5, 25. , 23.4, 18.9,
       35.4, 24.7, 31.6, 23.3, 19.6, 18.7, 16. , 22.2, 25. , 33. , 23.5,
       19.4, 22. , 17.4, 20.9, 24.2, 21.7, 22.8, 23.4, 24.1, 21.4, 20. ,
       20.8, 21.2, 20.3, 28. , 23.9, 24.8, 22.9, 23.9, 26.6, 22.5, 22.2,
       23.6, 28.7, 22.6, 22. , 22.9, 25. , 20.6, 28.4, 21.4, 38.7, 43.8,
       33.2, 27.5, 26.5, 18.6, 19.3, 20.1, 19.5, 19.5, 20.4, 19.8, 19.4,
       21.7, 22.8, 18.8, 18.7, 18.5, 18.3, 21.2, 19.2, 20.4, 19.3, 22. ,
       20.3, 20.5, 17.3, 18.8, 21.4, 15.7, 16.2, 18. , 14.3, 19.2, 19.6,
       23. , 18.4, 15.6, 18.1, 17.4, 17.1, 13.3, 17.8, 14. , 14.4, 13.4,
       15.6, 11.8, 13.8, 15.6, 14.6, 17.8, 15.4, 21.5, 19.6, 15.3, 19.4,
       17. , 15.6, 13.1, 41.3, 24.3, 23.3, 27. , 50. , 50. , 50. , 22.7,
       25. , 50. , 23.8, 23.8, 22.3, 17.4, 19.1, 23.1, 23.6, 22.6, 29.4,
       23.2, 24.6, 29.9, 37.2, 39.8, 36.2, 37.9, 32.5, 26.4, 29.6, 50. ,
       32. , 29.8, 34.9, 37. , 30.5, 36.4, 31.1, 29.1, 50. , 33.3, 30.3,
       34.6, 34.9, 32.9, 24.1, 42.3, 48.5, 50. , 22.6, 24.4, 22.5, 24.4,
       20. , 21.7, 19.3, 22.4, 28.1, 23.7, 25. , 23.3, 28.7, 21.5, 23. ,
       26.7, 21.7, 27.5, 30.1, 44.8, 50. , 37.6, 31.6, 46.7, 31.5, 24.3,
       31.7, 41.7, 48.3, 29. , 24. , 25.1, 31.5, 23.7, 23.3, 22. , 20.1,
       22.2, 23.7, 17.6, 18.5, 24.3, 20.5, 24.5, 26.2, 24.4, 24.8, 29.6,
       42.8, 21.9, 20.9, 44. , 50. , 36. , 30.1, 33.8, 43.1, 48.8, 31. ,
       36.5, 22.8, 30.7, 50. , 43.5, 20.7, 21.1, 25.2, 24.4, 35.2, 32.4,
       32. , 33.2, 33.1, 29.1, 35.1, 45.4, 35.4, 46. , 50. , 32.2, 22. ,
       20.1, 23.2, 22.3, 24.8, 28.5, 37.3, 27.9, 23.9, 21.7, 28.6, 27.1,
       20.3, 22.5, 29. , 24.8, 22. , 26.4, 33.1, 36.1, 28.4, 33.4, 28.2,
       22.8, 20.3, 16.1, 22.1, 19.4, 21.6, 23.8, 16.2, 17.8, 19.8, 23.1,
       21. , 23.8, 23.1, 20.4, 18.5, 25. , 24.6, 23. , 22.2, 19.3, 22.6,
       19.8, 17.1, 19.4, 22.2, 20.7, 21.1, 19.5, 18.5, 20.6, 19. , 18.7,
       32.7, 16.5, 23.9, 31.2, 17.5, 17.2, 23.1, 24.5, 26.6, 22.9, 24.1,
       18.6, 30.1, 18.2, 20.6, 17.8, 21.7, 22.7, 22.6, 25. , 19.9, 20.8,
       16.8, 21.9, 27.5, 21.9, 23.1, 50. , 50. , 50. , 50. , 50. , 13.8,
       13.8, 15. , 13.9, 13.3, 13.1, 10.2, 10.4, 10.9, 11.3, 12.3,  8.8,
        7.2, 10.5,  7.4, 10.2, 11.5, 15.1, 23.2,  9.7, 13.8, 12.7, 13.1,
       12.5,  8.5,  5. ,  6.3,  5.6,  7.2, 12.1,  8.3,  8.5,  5. , 11.9,
       27.9, 17.2, 27.5, 15. , 17.2, 17.9, 16.3,  7. ,  7.2,  7.5, 10.4,
        8.8,  8.4, 16.7, 14.2, 20.8, 13.4, 11.7,  8.3, 10.2, 10.9, 11. ,
        9.5, 14.5, 14.1, 16.1, 14.3, 11.7, 13.4,  9.6,  8.7,  8.4, 12.8,
       10.5, 17.1, 18.4, 15.4, 10.8, 11.8, 14.9, 12.6, 14.1, 13. , 13.4,
       15.2, 16.1, 17.8, 14.9, 14.1, 12.7, 13.5, 14.9, 20. , 16.4, 17.7,
       19.5, 20.2, 21.4, 19.9, 19. , 19.1, 19.1, 20.1, 19.9, 19.6, 23.2,
       29.8, 13.8, 13.3, 16.7, 12. , 14.6, 21.4, 23. , 23.7, 25. , 21.8,
       20.6, 21.2, 19.1, 20.6, 15.2,  7. ,  8.1, 13.6, 20.1, 21.8, 24.5,
       23.1, 19.7, 18.3, 21.2, 17.5, 16.8, 22.4, 20.6, 23.9, 22. , 11.9])
```

# Iteration 1

### i. test_size=0.3 random_state=0

In [7]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.3,random_state = 0)
```

In [8]:
```
model= LinearRegression()
```

In [9]:
```
model.fit(x_train,y_train)
```

Out[9]:
```
▾ LinearRegression
LinearRegression()
```

In [10]:
```
y_predict = model.predict(x_test)
```

In [11]:
```
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: 27.20
R-squared: 0.67
```

In [12]:
```
model.score(x,y)
```

Out[12]:
```
0.7378284679886178
```

In [13]:
```
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, "Differe
pred_y_df
```

Out[13]:

| | Actual Values | predicted values | Difference |
|---|---|---|---|
| **0** | 22.6 | 24.935708 | -2.335708 |
| **1** | 50.0 | 23.751632 | 26.248368 |
| **2** | 23.0 | 29.326383 | -6.326383 |
| **3** | 8.3 | 11.975346 | -3.675346 |
| **4** | 21.2 | 21.372725 | -0.172725 |
| **...** | ... | ... | ... |
| **147** | 36.2 | 27.811077 | 8.388923 |
| **148** | 11.0 | 14.506816 | -3.506816 |
| **149** | 7.2 | 7.573699 | -0.373699 |
| **150** | 22.8 | 28.334807 | -5.534807 |
| **151** | 28.7 | 25.043412 | 3.656588 |

152 rows × 3 columns

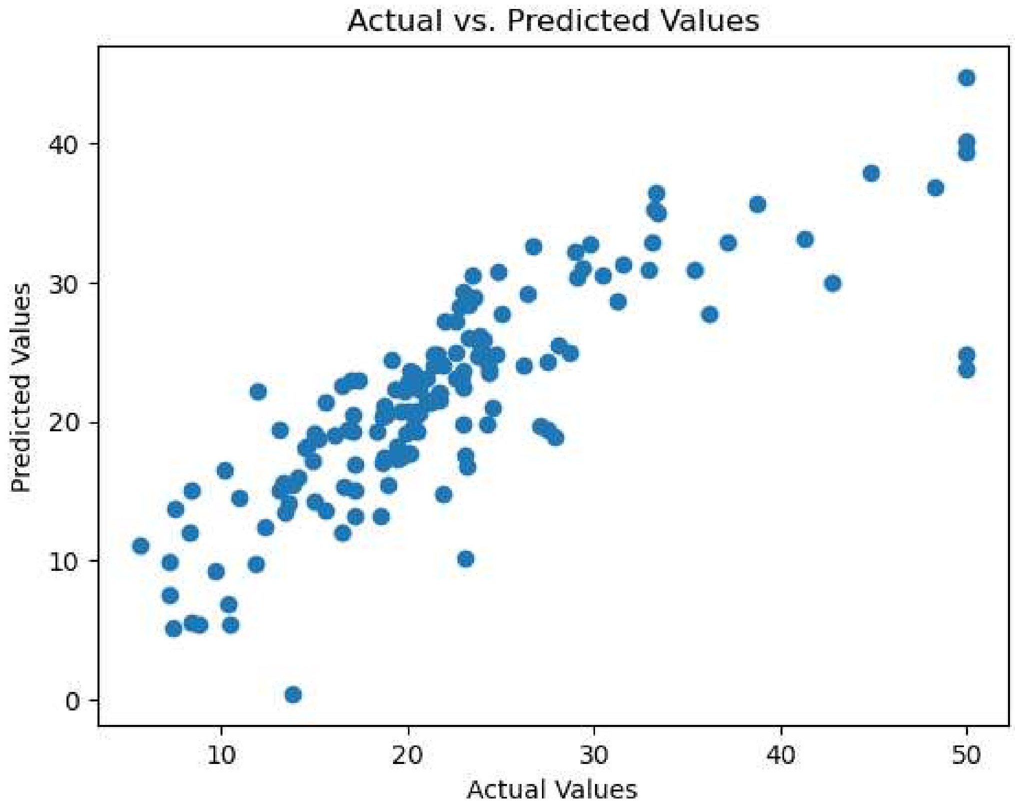
In [14]:
```
model.coef_
```

Out[14]:
```
array([-1.21310401e-01,  4.44664254e-02,  1.13416945e-02,  2.51124642e+00,
       -1.62312529e+01,  3.85906801e+00, -9.98516565e-03, -1.50026956e+00,
        2.42143466e-01, -1.10716124e-02, -1.01775264e+00,  6.81446545e-03,
       -4.86738066e-01])
```

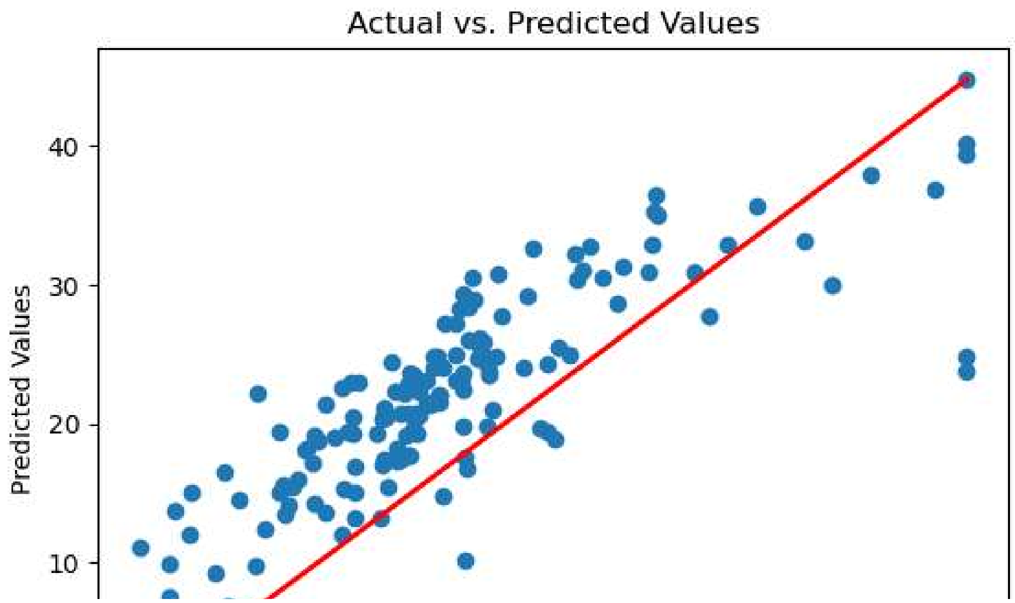
In [15]:
```
model.intercept_
```

Out[15]:
```
37.93710774183272
```

In [16]:
```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [17]:
```
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 [18]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.3,random_state = 42
```

In [19]:
```
model= LinearRegression()
```

In [20]:
```
model.fit(x_train,y_train)
```

Out[20]:
```
▾ LinearRegression
LinearRegression()
```

In [21]:
```
y_predict = model.predict(x_test)
```

In [22]:
```
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: 21.52
R-squared: 0.71
```

In [23]:
```
model.score(x,y)
```

Out[23]:
```
0.7365934508044951
```

In [24]:
```
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, "Differe
pred_y_df
```

Out[24]:
```
     Actual Values  predicted values  Difference
0             23.6         28.648960   -5.048960
1             32.4         36.495014   -4.095014
2             13.6         15.411193   -1.811193
3             22.8         25.403213   -2.603213
4             16.1         18.855280   -2.755280
...            ...               ...         ...
147           17.1         17.403672   -0.303672
148           14.5         13.385941    1.114059
149           50.0         39.983425   10.016575
150           14.3         16.682863   -2.382863
151           12.6         18.285618   -5.685618

152 rows × 3 columns
```

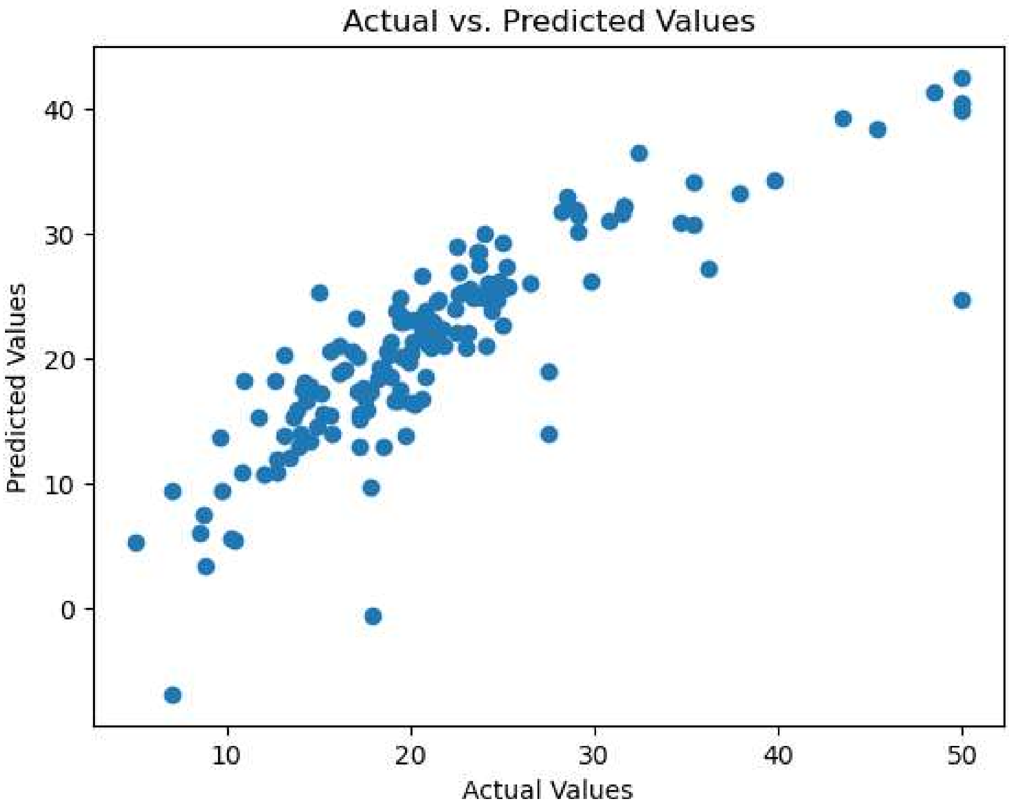
In [25]:
```
model.coef_
```

Out[25]:
```
array([-1.33470103e-01,  3.58089136e-02,  4.95226452e-02,  3.11983512e+00,
       -1.54170609e+01,  4.05719923e+00, -1.08208352e-02, -1.38599824e+00,
        2.42727340e-01, -8.70223437e-03, -9.10685208e-01,  1.17941159e-02,
       -5.47113313e-01])
```

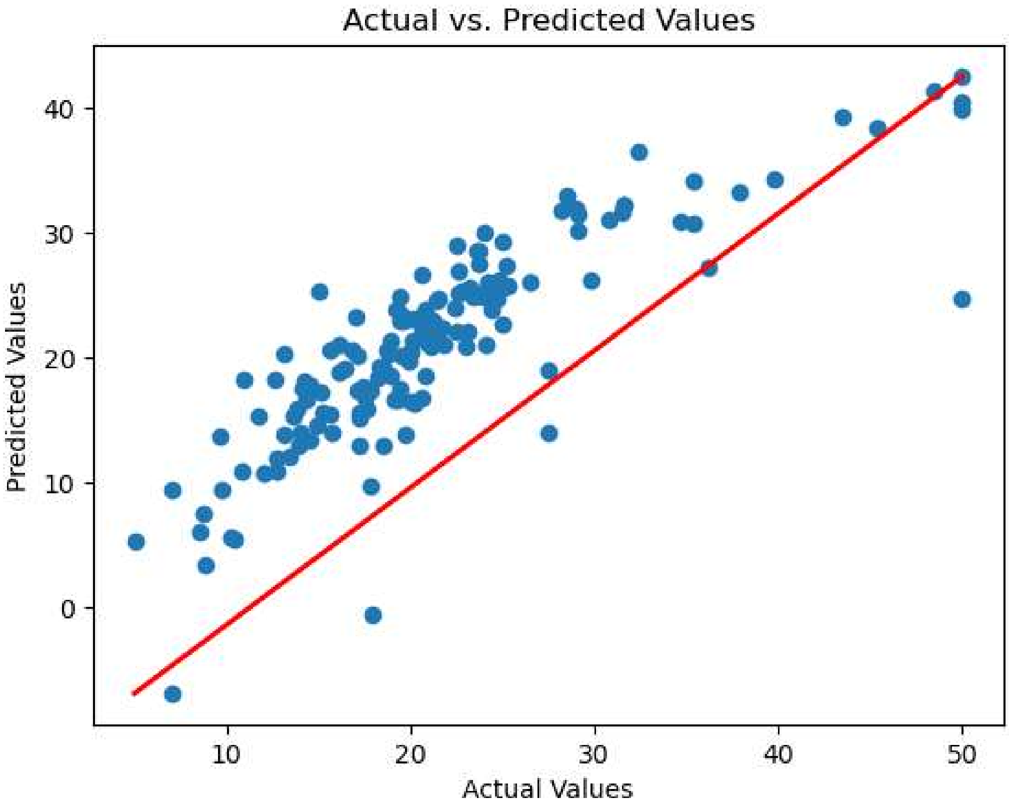
In [26]:
```
model.intercept_
```

Out[26]:
```
31.63108403569373
```

In [27]:
```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [28]:
```
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 [29]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.4,random_state = 0)
```

In [30]:
```
model= LinearRegression()
```

In [31]:
```
model.fit(x_train,y_train)
```

Out[31]:
```
▾ LinearRegression
LinearRegression()
```

In [32]:
```
y_predict = model.predict(x_test)
```

In [33]:

```
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: 25.79
R-squared: 0.69
```

In [34]:

```
model.score(x,y)
```

Out[34]:

```
0.7361196190442469
```

In [35]:

```
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, "Differe
pred_y_df
```

Out[35]:

| | Actual Values | predicted values | Difference |
|---|---|---|---|
| **0** | 22.6 | 24.581552 | -1.981552 |
| **1** | 50.0 | 24.516293 | 25.483707 |
| **2** | 23.0 | 29.713799 | -6.713799 |
| **3** | 8.3 | 12.511327 | -4.211327 |
| **4** | 21.2 | 21.349654 | -0.149654 |
| **...** | ... | ... | ... |
| **198** | 22.1 | 27.219549 | -5.119549 |
| **199** | 46.0 | 40.065076 | 5.934924 |
| **200** | 22.9 | 29.489264 | -6.589264 |
| **201** | 20.2 | 16.588711 | 3.611289 |
| **202** | 43.1 | 37.210154 | 5.889846 |

203 rows × 3 columns

In [36]:

```
model.coef_
```

Out[36]:

```
array([-1.03747356e-01,  5.58589924e-02,  5.88240770e-02,  2.50523544e+00,
       -1.90284888e+01,  3.25353601e+00, -3.22150522e-03, -1.57603462e+00,
        2.58716068e-01, -1.14681299e-02, -1.10777478e+00,  5.50051783e-03,
       -5.59569992e-01])
```

In [37]:

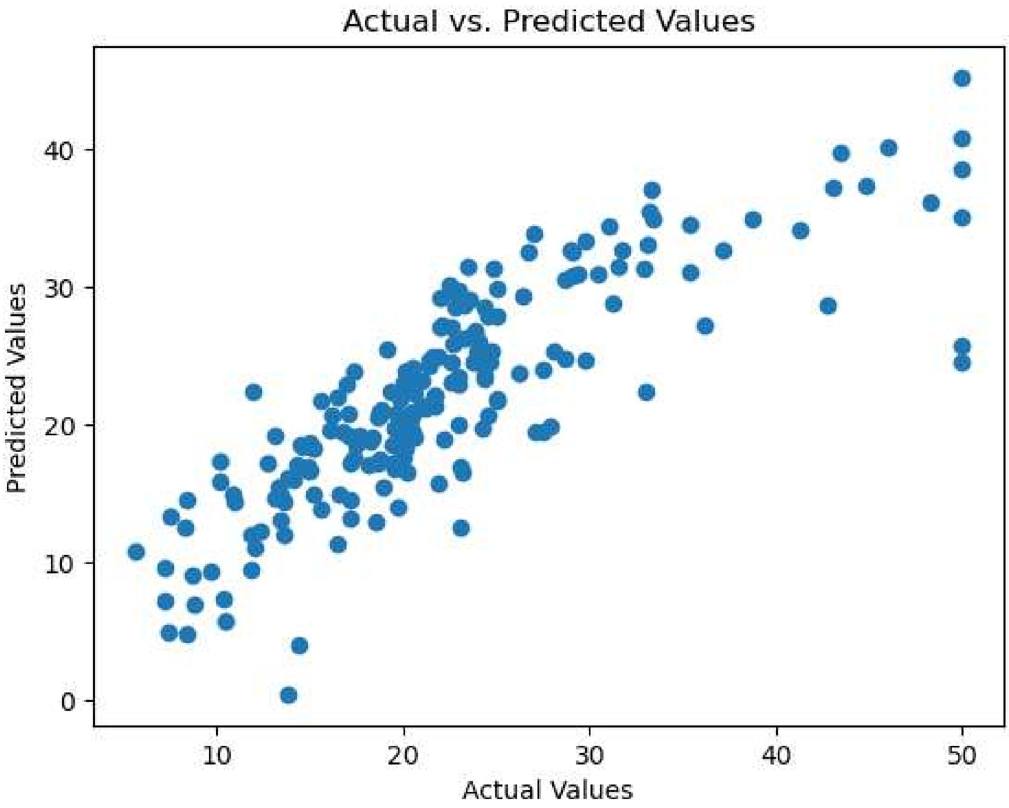
```
model.intercept_
```

Out[37]:

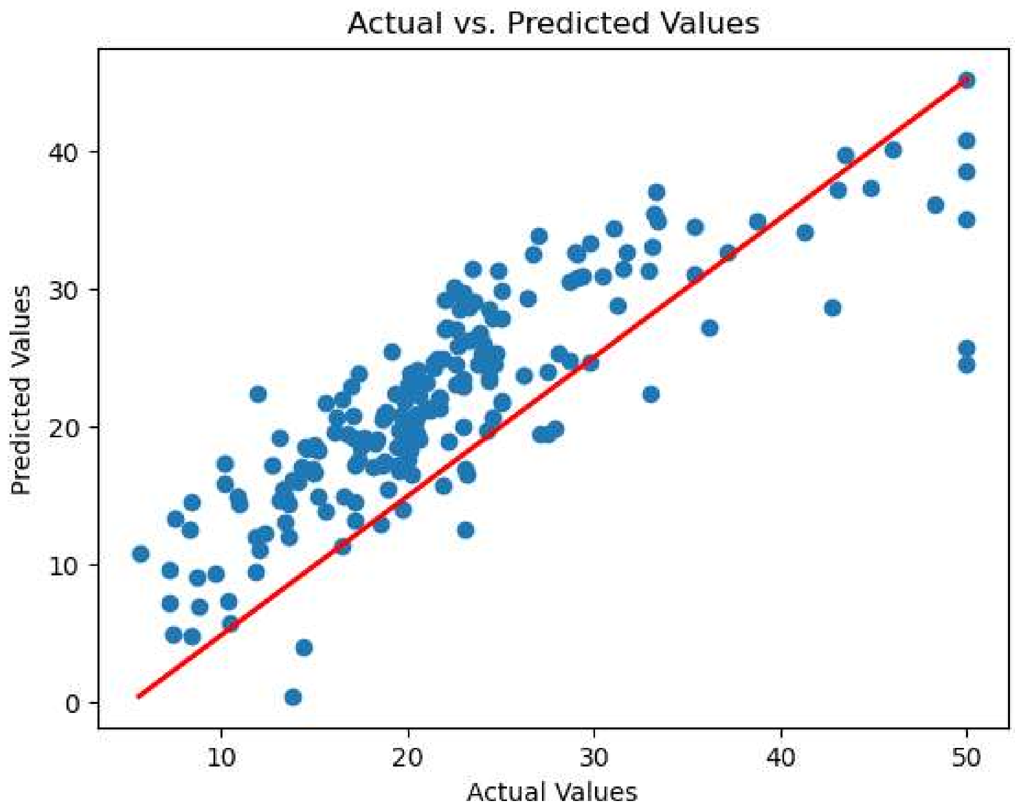
```
45.481419593251005
```

In [38]:

```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [39]:
```
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 [40]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.4,random_state = 42
```

In [41]:
```
model= LinearRegression()
```

In [42]:
```
model.fit(x_train,y_train)
```

Out[42]:
```
▾ LinearRegression
LinearRegression()
```

In [43]:
```
y_predict = model.predict(x_test)
```

In [44]:
```
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: 21.83
R-squared: 0.71
```

In [45]:
```
model.score(x,y)
```

Out[45]:
```
0.7362442925015322
```

In [46]:
```
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, "Differe
pred_y_df
```

Out[46]:

| | Actual Values | predicted values | Difference |
|---|---|---|---|
| **0** | 23.6 | 28.700257 | -5.100257 |
| **1** | 32.4 | 37.275424 | -4.875424 |
| **2** | 13.6 | 14.450663 | -0.850663 |
| **3** | 22.8 | 25.734356 | -2.934356 |
| **4** | 16.1 | 18.497639 | -2.397639 |
| **...** | ... | ... | ... |
| **198** | 16.7 | 19.366259 | -2.666259 |
| **199** | 13.8 | 11.536608 | 2.263392 |
| **200** | 22.9 | 28.982763 | -6.082763 |
| **201** | 15.3 | 21.744787 | -6.444787 |
| **202** | 27.5 | 32.575230 | -5.075230 |

203 rows × 3 columns

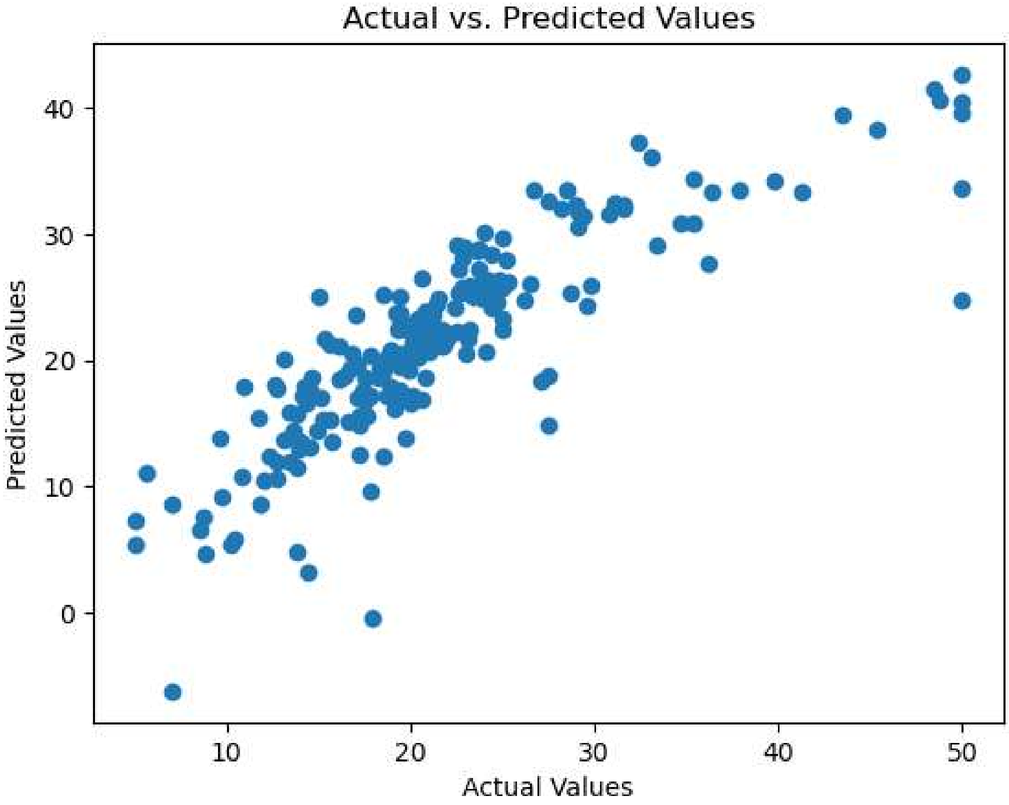
In [47]:
```
model.coef_
```

Out[47]:
```
array([-1.13498178e-01,  3.79976868e-02,  2.99876109e-02,  3.51131977e+00,
       -1.44513558e+01,  3.80534175e+00, -1.79553906e-02, -1.48968845e+00,
        2.53016064e-01, -1.00262729e-02, -8.86408743e-01,  1.08462004e-02,
       -5.75917903e-01])
```

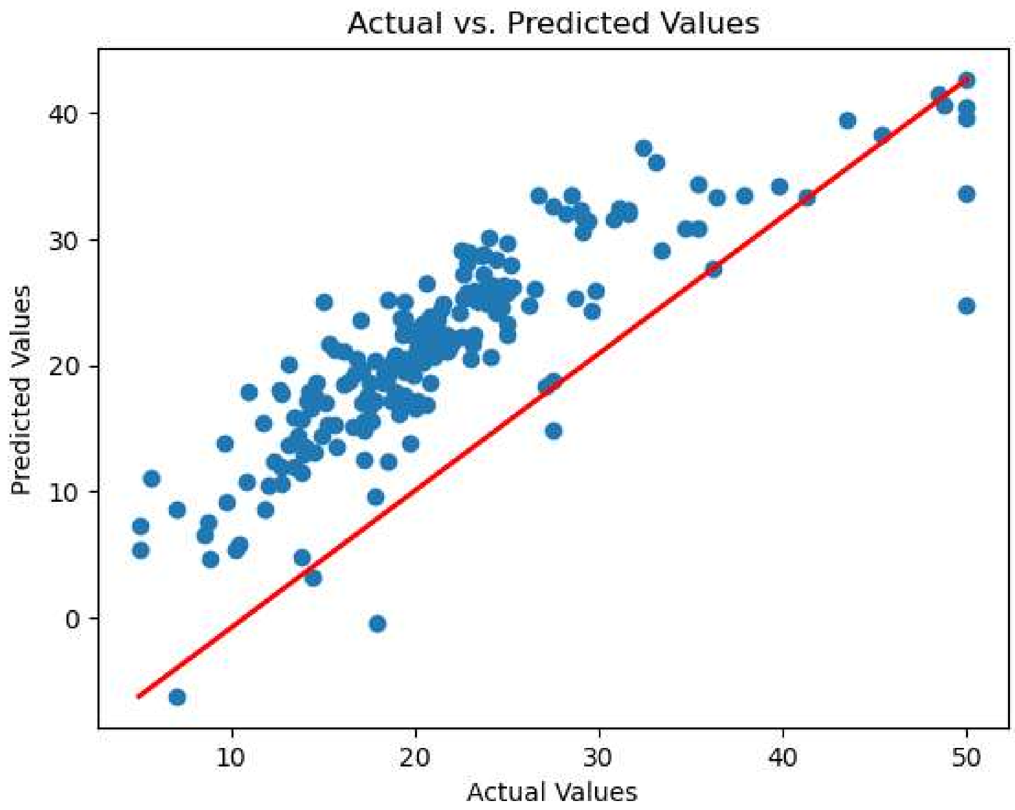
In [48]:
```
model.intercept_
```

Out[48]:
```
34.33026076546706
```

In [49]:
```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [50]:
```
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 [51]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.5,random_state = 0)
```

In [52]:
```
model= LinearRegression()
```

In [53]:
```
model.fit(x_train,y_train)
```

Out[53]:
```
▾ LinearRegression
LinearRegression()
```

In [54]:
```
y_predict = model.predict(x_test)
```

In [55]:
```
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: 25.30
R-squared: 0.67
```

In [56]:
```
model.score(x,y)
```

Out[56]:
```
0.733888612575677
```

In [57]:
```
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, "Differe
pred_y_df
```

Out[57]:

| | Actual Values | predicted values | Difference |
|---|---|---|---|
| **0** | 22.6 | 24.709603 | -2.109603 |
| **1** | 50.0 | 25.126314 | 24.873686 |
| **2** | 23.0 | 30.765033 | -7.765033 |
| **3** | 8.3 | 12.414055 | -4.114055 |
| **4** | 21.2 | 21.295995 | -0.095995 |
| **...** | ... | ... | ... |
| **248** | 32.2 | 32.604891 | -0.404891 |
| **249** | 13.5 | 12.787651 | 0.712349 |
| **250** | 17.9 | 1.206608 | 16.693392 |
| **251** | 13.3 | 19.442771 | -6.142771 |
| **252** | 11.7 | 15.319865 | -3.619865 |

253 rows × 3 columns

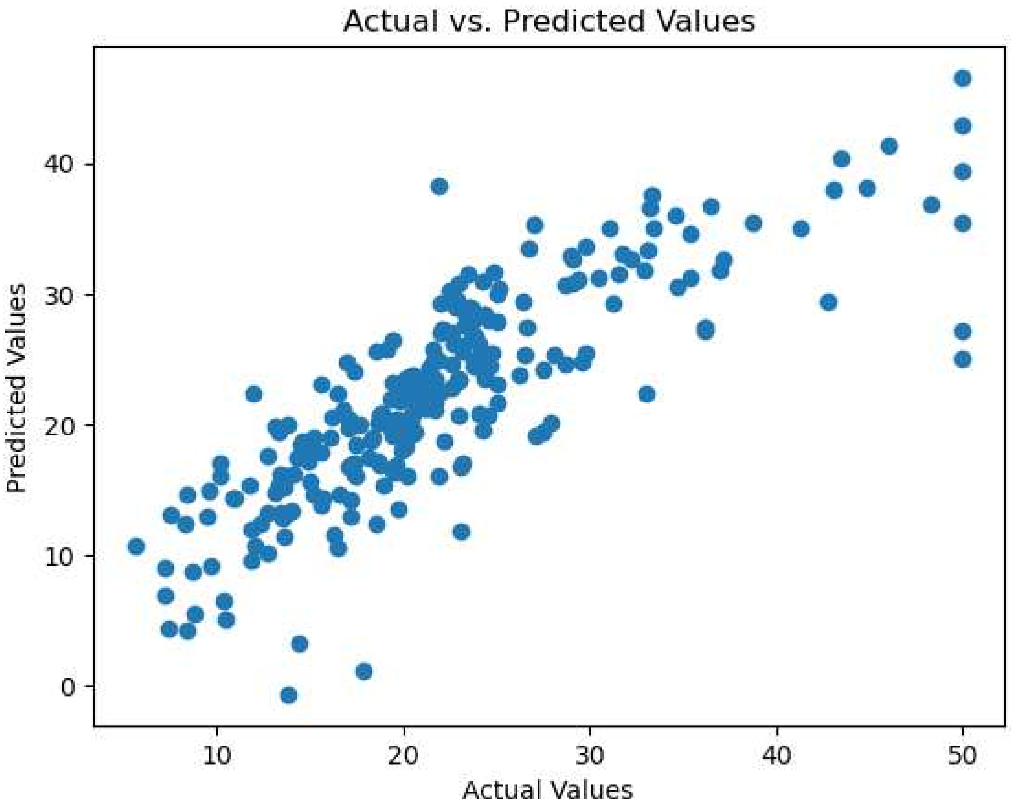
In [58]:
```
model.coef_
```

Out[58]:
```
array([-1.22697052e-01,  5.76835439e-02,  7.42047961e-02,  3.38949970e+00,
       -1.65391519e+01,  3.57730248e+00, -2.98033902e-03, -1.55759520e+00,
        2.50098217e-01, -9.73850155e-03, -1.12527834e+00,  6.85073312e-03,
       -5.92410460e-01])
```

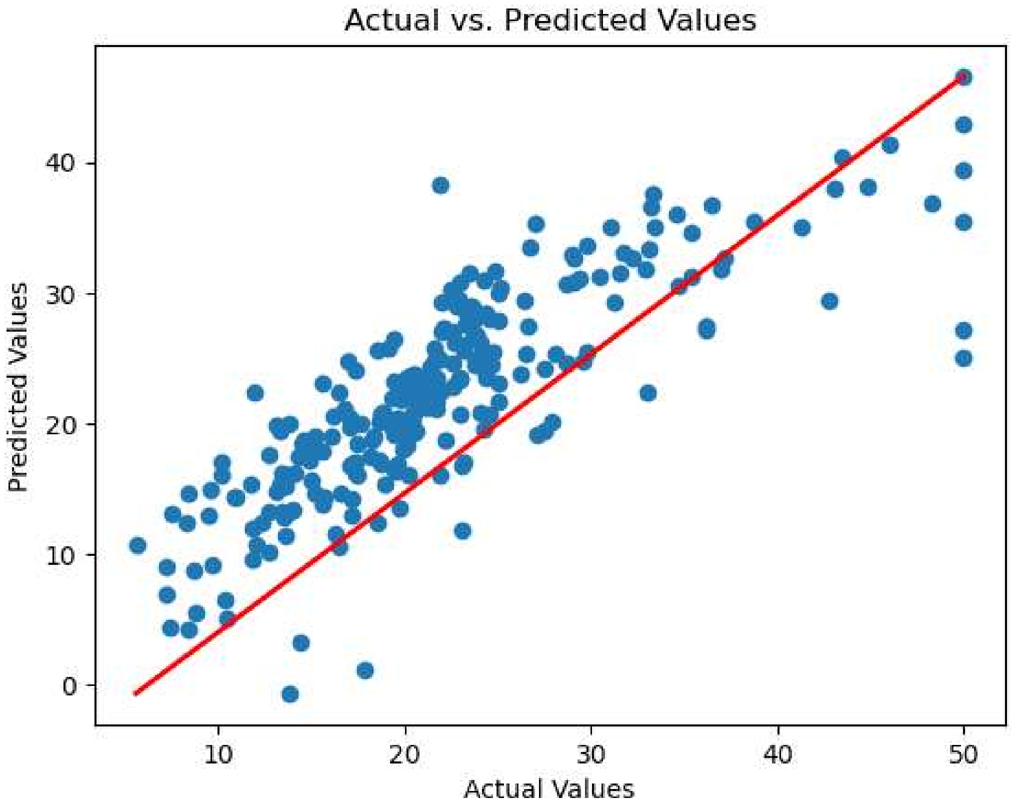
In [59]:
```
model.intercept_
```

Out[59]:
```
41.58607683099513
```

In [60]:
```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [61]:
```
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 [62]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.5,random_state = 42
```

In [63]:
```
model= LinearRegression()
```

In [64]:
```
model.fit(x_train,y_train)
```

Out[64]:
```
▾ LinearRegression
LinearRegression()
```

In [65]:
```
y_predict = model.predict(x_test)
```

In [66]:
```
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: 25.18
R-squared: 0.69
```

In [67]:
```
model.score(x,y)
```

Out[67]:
```
0.7325693559556025
```

In [68]:
```
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, "Differe
pred_y_df
```

Out[68]:

| | Actual Values | predicted values | Difference |
|---|---|---|---|
| **0** | 23.6 | 28.280203 | -4.680203 |
| **1** | 32.4 | 37.739436 | -5.339436 |
| **2** | 13.6 | 15.407600 | -1.807600 |
| **3** | 22.8 | 25.441452 | -2.641452 |
| **4** | 16.1 | 18.605815 | -2.505815 |
| **...** | ... | ... | ... |
| **248** | 19.3 | 20.360373 | -1.060373 |
| **249** | 23.9 | 26.435753 | -2.535753 |
| **250** | 24.7 | 23.326693 | 1.373307 |
| **251** | 19.8 | 18.110622 | 1.689378 |
| **252** | 23.8 | 22.463210 | 1.336790 |

253 rows × 3 columns

In [69]:
```
model.coef_
```

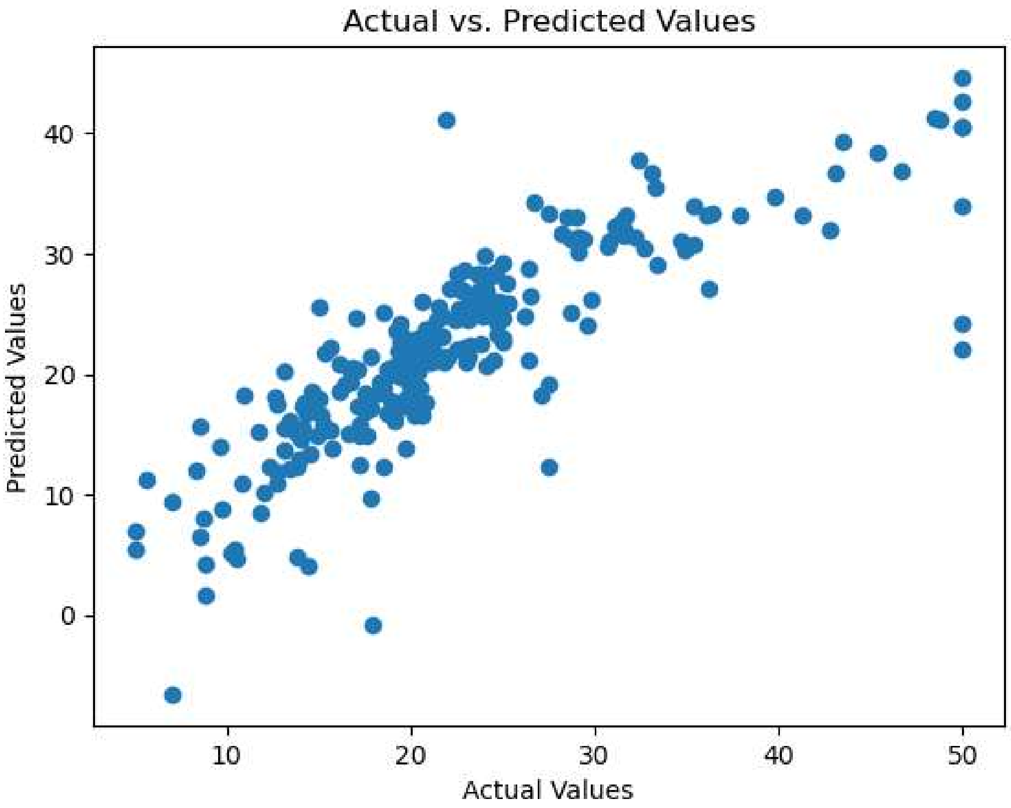
Out[69]:
```
array([-1.17646736e-01,  3.32466019e-02,  3.05213436e-02,  4.40864134e+00,
       -1.34354334e+01,  4.54696921e+00, -2.19317442e-02, -1.44431337e+00,
        1.98900056e-01, -8.07440421e-03, -8.09040596e-01,  1.11002543e-02,
       -5.15546338e-01])
```

In [70]:
```
model.intercept_
```

Out[70]:
```
26.60863720541145
```

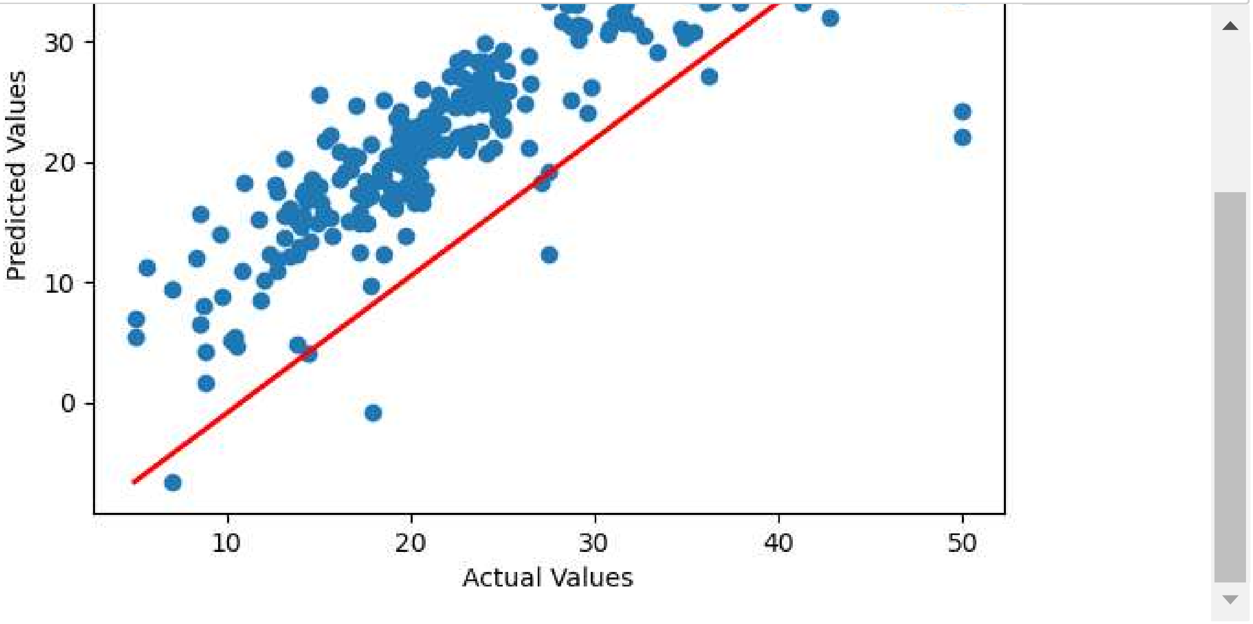
In [71]:

```
plt.scatter(y_test, y_predict)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



In [72]:

```
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()
```


Predicted Values
Actual Values