```
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import numpy as np
```

```
X, y = make_regression(n_samples=200, n_features=1, noise=20, random_state=42)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
model = LinearRegression()
```

```
model.fit(X_train, y_train)
```

LinearRegression

LinearRegression()

```
y_pred = model.predict(X_test)
```

```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

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

```
Mean squared error: 437.55
R-squared score: 0.94
```

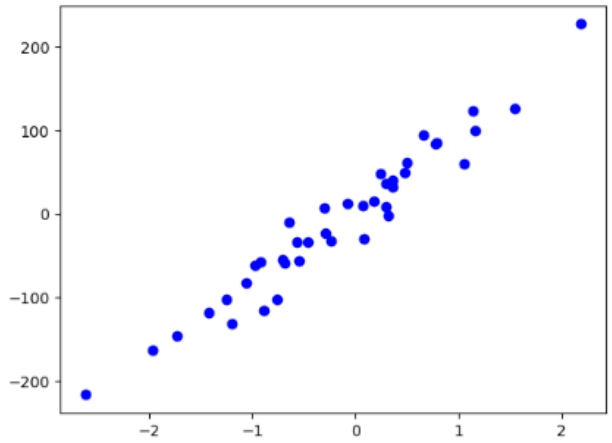
```
import seaborn as sns
```

```
plt.figure(figsize=(10, 6))
```

```
<Figure size 1000x600 with 0 Axes>
<Figure size 1000x600 with 0 Axes>
```

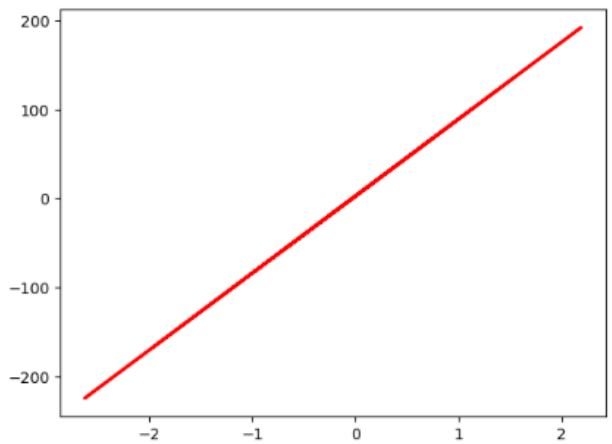
```
plt.scatter(X_test, y_test, color='blue', label='Actual')
```

```
<matplotlib.collections.PathCollection at 0x2013d318110>
```

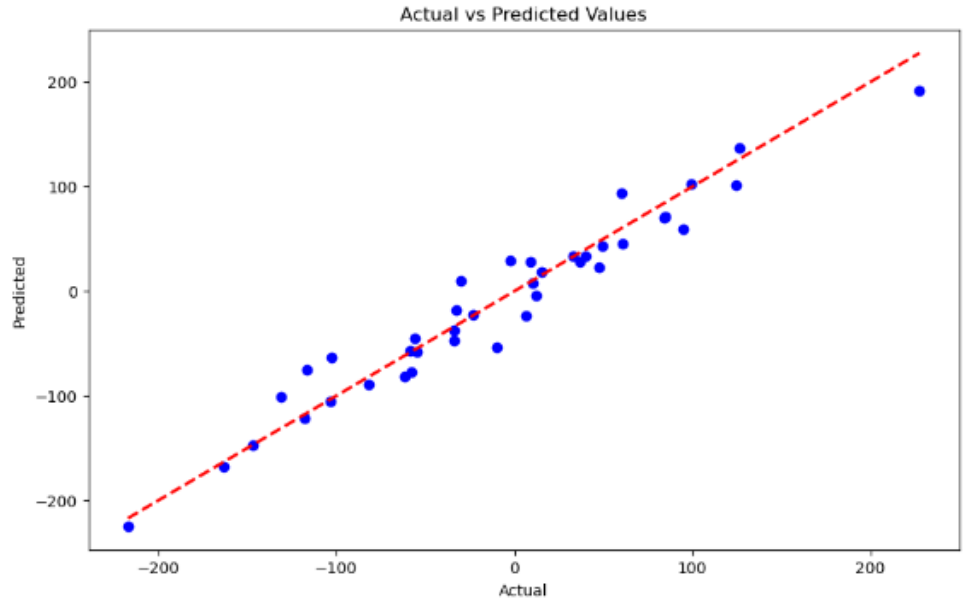



```
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Regression Line')
```

```
[<matplotlib.lines.Line2D at 0x2013d73a2d0>]
```



```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='blue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=2)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted Values')
plt.show()
```




```
residuals = y_test - y_pred
```

```
plt.figure(figsize=(21, 7))
plt.hist(residuals, bins=20, edgecolor='red')
plt.xlabel('Residuals')
plt.ylabel('Frequency')
plt.title('Histogram of Residuals')
plt.show()
```