Project Title : Predicting Weight from Height Using Linear Regression

Objective : Build a linear regression model to predict weight based on height and evaluate its performance.

Step 1: Import Libraries

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

Step 2: Create the Dataset

In [2]:
```
# Manually input the dataset
data = {
"Height": [63, 64, 66, 69, 70, 65, 68, 72, 71, 67],
"Weight": [127, 121, 142, 157, 162, 135, 149, 175, 168, 145]
}
# Convert to DataFrame
df = pd.DataFrame(data)
```

In [3]:
```
df
```

Out[3]:

	Height	Weight
0	63	127
1	64	121
2	66	142
3	69	157
4	70	162
5	65	135
6	68	149
7	72	175
8	71	168
9	67	145

Step 3: Exploratory Data Analysis (EDA)

In [4]:
```
# Display first 5 rows
print("First 5 rows of the dataset:")
print(df.head())
```

```
First 5 rows of the dataset:
   Height  Weight
0      63     127
1      64     121
2      66     142
3      69     157
4      70     162
```

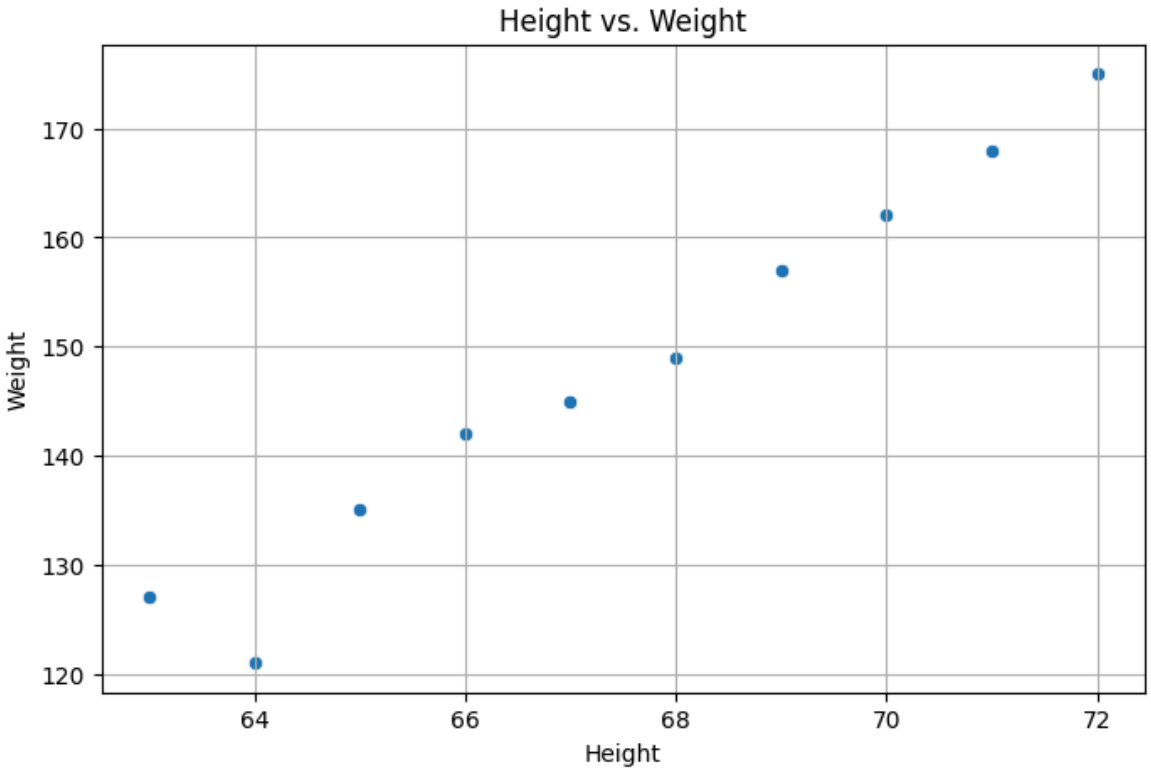
In [5]:
```
# Summary statistics
print("\nSummary Statistics:")
print(df.describe())
```

```
Summary Statistics:
         Height      Weight
count  10.00000   10.000000
mean   67.50000  148.100000
std     3.02765   17.609656
min    63.00000  121.000000
25%    65.25000  136.750000
50%    67.50000  147.000000
75%    69.75000  160.750000
max    72.00000  175.000000
```

In [6]:
```python
# Check correlation
print("\nCorrelation between Height and Weight:")
print(df.corr())
```

```
Correlation between Height and Weight:
          Height    Weight
Height  1.000000  0.984697
Weight  0.984697  1.000000
```

In [7]:
```python
# Visualize the relationship
plt.figure(figsize=(8, 5))
sns.scatterplot(x="Height", y="Weight", data=df)
plt.title("Height vs. Weight")
plt.grid(True)
plt.show()
```



Step 4: Prepare Data for Modeling

In [8]:
```python
# Define features (X) and target (y)
X = df[["Height"]] # 2D array required by scikit-learn
y = df["Weight"]
```

Step 5: Train the Linear Regression Model

In [9]:
```python
# Initialize and fit the model
model = LinearRegression()
model.fit(X, y)
# Extract coefficients
beta_0 = model.intercept_
beta_1 = model.coef_[0]
print(f"\nModel Equation: Weight = {beta_0:.2f} + {beta_1:.2f} * Height")
```

```
Model Equation: Weight = -238.49 + 5.73 * Height
```

Step 6: Make Predictions

In [11]:
```
# Predict on the same dataset (for visualization)
df["Predicted Weight"] = model.predict(X)
# Predict for a new height (e.g., 70 inches)
new_height = np.array([[70]])
predicted_weight = model.predict(new_height)
print(f"\nPredicted weight for 70 inches: {predicted_weight[0]:.2f} pounds")
```

```
Predicted weight for 70 inches: 162.42 pounds
```

```
c:\Users\sachi\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.py:493: UserWarning:
X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

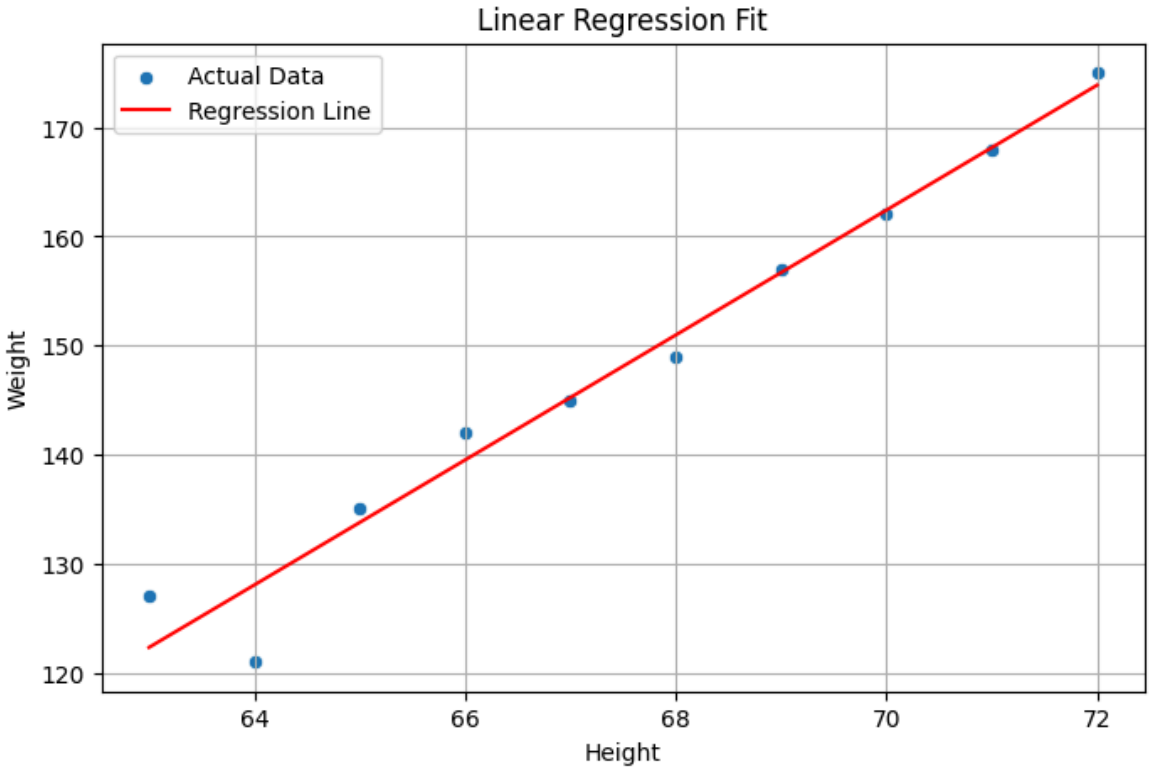
Step 7: Evaluate the Model

In [13]:
```
# Calculate metrics
mse = mean_squared_error(y, df["Predicted Weight"])
rmse = np.sqrt(mse)
r2 = r2_score(y, df["Predicted Weight"])
print(f"\nMSE: {mse:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
MSE: 8.48
RMSE: 2.91
R-squared: 0.97
```

Step 8: Visualize the Regression Line

In [15]:
```
plt.figure(figsize=(8, 5))
sns.scatterplot(x="Height", y="Weight", data=df, label="Actual Data")
sns.lineplot(x="Height", y="Predicted Weight", data=df, color="red", label="Regression Line")
plt.title("Linear Regression Fit")
plt.grid(True)
plt.legend()
plt.show()
```



Step 9: Key Takeaways

1. Model Performance : R-squared = 0.97 (97% of variance in weight explained by height). RMSE = 2.91 (predictions are off by ~3 pounds on average).
2. Interpretation : For every additional inch in height, weight increases by 5.73 pounds .