Linear Regression

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog). There are two main types:

Simple regression - only one independent variable (e.g. sales)

Multiple regression - more than one independent variables (e.g. sales, price)

Simple Linear Regression

Simple linear regression is an approach for predicting a response using a single feature. It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x).

In [1]:

```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

In [2]:

```
# load the data tips from sns
df = sns.load_dataset('tips')
df.head()
```

Out[2]:

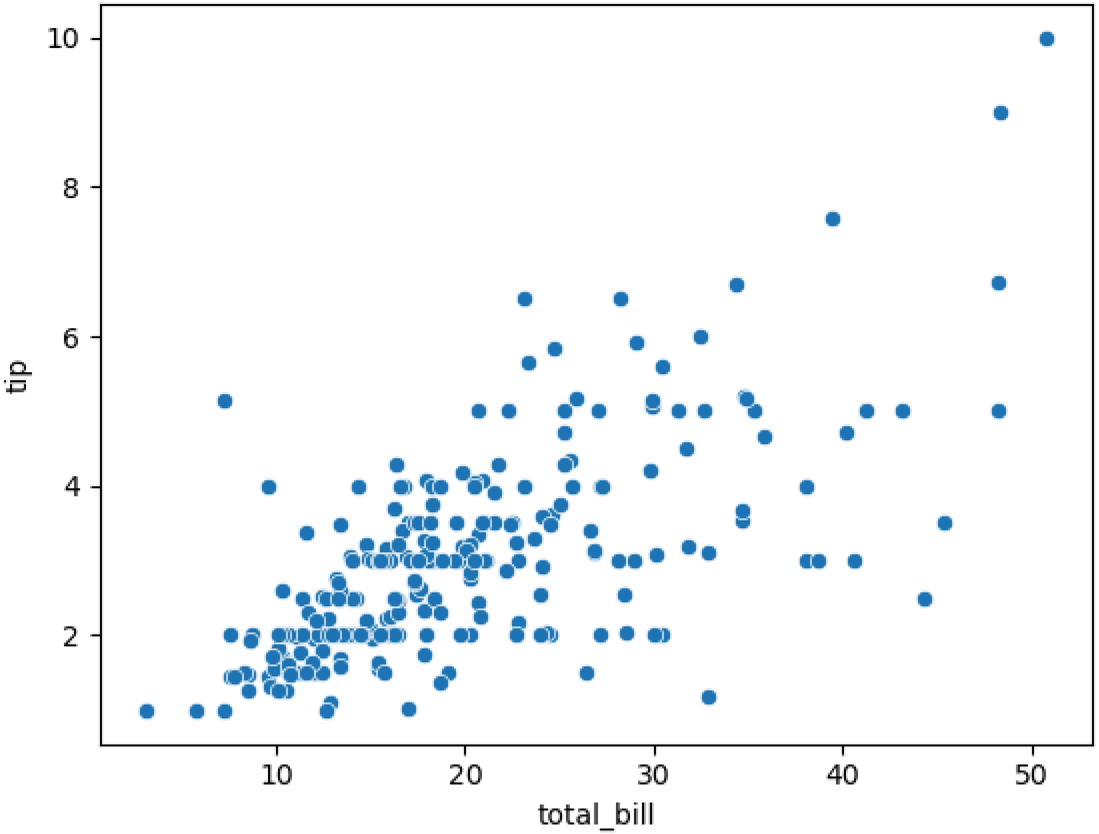
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [3]:

```
sns.scatterplot(x='total_bill', y='tip', data=df)
```

Out[3]:

```
<Axes: xlabel='total_bill', ylabel='tip'>
```



In [4]:
```
# split the data into X and y
X = df[['total_bill']]
# scalar = MinMaxScaler()
# X = scalar.fit_transform(X)
y = df['tip']
```

In [5]:
```
# split the data into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [6]:
```
# call the model
model = LinearRegression()
```

In [7]:
```
# train the model
model.fit(X_train, y_train)
```

Out[7]:
```
▾ LinearRegression
LinearRegression()
```

In [8]:
```
# take out model intercept and slop, make an equation
print(model.intercept_)
print(model.coef_)
print('y = ', model.intercept_, '+', model.coef_, '* X')
```

```
0.9018607268977337
[0.10670471]
y =  0.9018607268977337 + [0.10670471] * X
```

In [9]:
```
model.predict([[5]])
```

```
C:\Users\ustb\.anaconda\anwaar\Lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature name
s, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[9]:
```
array([1.43538425])
```

In [10]:
```
# predict
y_pred = model.predict(X_test)
```

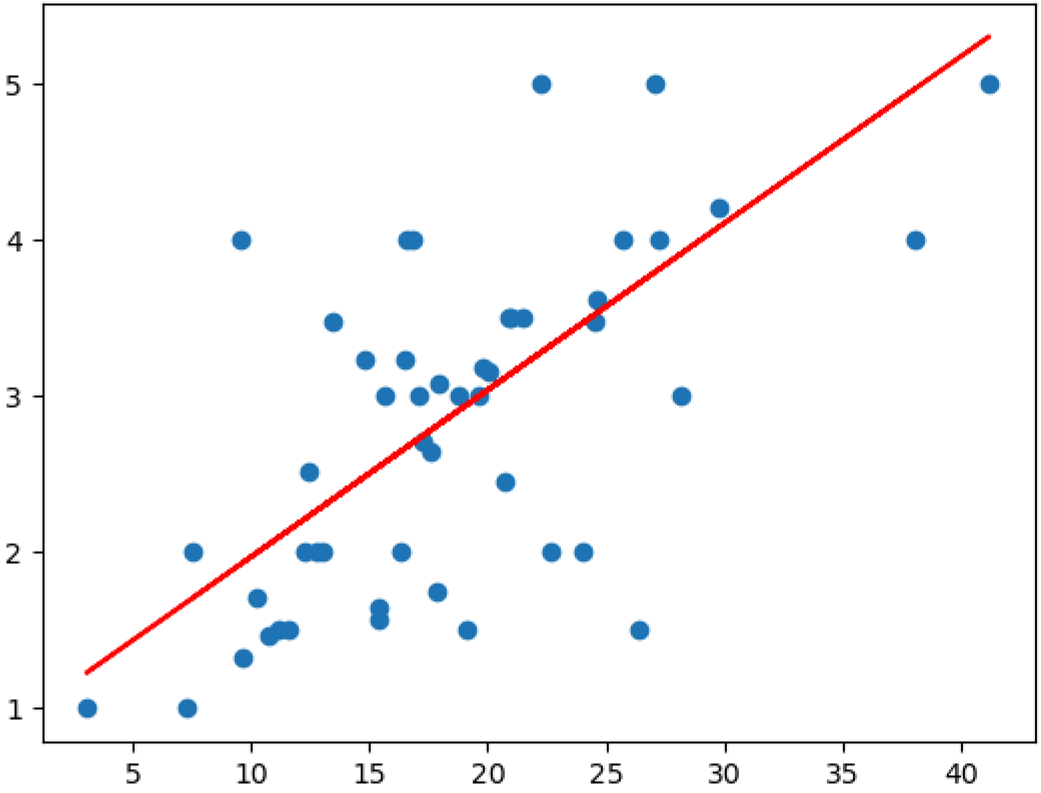
In [11]:

```
# evaluate the model
print('MSE = ', mean_squared_error(y_test, y_pred))
print('R2 = ', r2_score(y_test, y_pred))
print('RMSE = ', np.sqrt(mean_squared_error(y_test, y_pred)))
```

```
MSE =  0.6984756329422364
R2 =  0.39026682745460495
RMSE =  0.8357485464792842
```

In [12]:

```
# plot the model and data
plt.scatter(X_test, y_test)
plt.plot(X_test, y_pred, color='red')
plt.show()
```



In [ ]:

```
```