Experiment title: 3

❖ **Aim/Overview of the practical:** Data analysis of any data set via graphs using linear regression.

❖ **Linear Regression – Finding a straight line of best fit through the data .This works well when the true underlying function is linear.**

A linear model makes a "hypothesis" about the true nature of the underlying function - that it is linear. We express this hypothesis in the univariate case as

$$h\theta(x)=ax+b$$

Our simple example above was an example of "univariate regression" - i.e. just one variable (or "feature") - number of hours studied. Below we will have more than one feature ("multivariate regression") which is given by

$$h\theta(\mathbf{x})=\mathbf{a}^\top\mathbf{X}$$

Here **a**

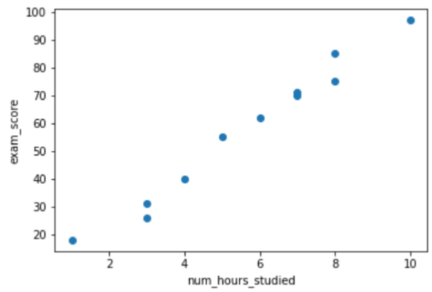
❖ is a vector of learned parameters, and **X** is the "design matrix" with all the data points. In this formulation the intercept term has been added to the design matrix as the first column (of all ones).

❖ **Code & Output:**

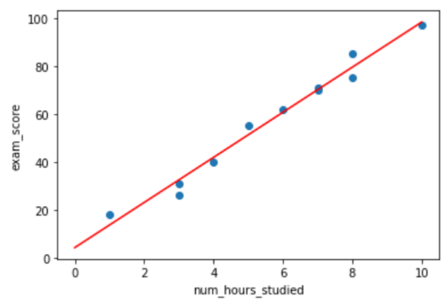
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [1]:
```python
import matplotlib.pyplot as plt
from sklearn import linear_model, metrics, model_selection
import numpy as np
import pandas as pd
```

In [2]:
```python
num_hours_studied = np.array([1, 3, 3, 4, 5, 6, 7, 7, 8, 8, 10])
exam_score = np.array([18, 26, 31, 40, 55, 62, 71, 70, 75, 85, 97])
plt.scatter(num_hours_studied, exam_score)
plt.xlabel('num_hours_studied')
plt.ylabel('exam_score')
plt.show()
```



In [4]:
```python
plt.scatter(num_hours_studied, exam_score)
x = np.linspace(0, 10)
y = a*x + b
plt.plot(x, y, 'r')
plt.xlabel('num_hours_studied')
plt.ylabel('exam_score')
plt.show()
```

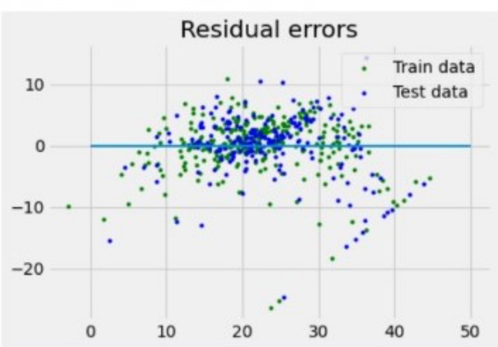


In [ ]:

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np
        from sklearn import datasets, linear_model, metrics

        # load the boston dataset
        boston = datasets.load_boston(return_X_y=False)

        # defining feature matrix(X) and response vector(y)
        X = boston.data
        y = boston.target

        # splitting X and y into training and testing sets
        from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
                                                            random_state=1)

        # create linear regression object
        reg = linear_model.LinearRegression()

        # train the model using the training sets
        reg.fit(X_train, y_train)

        # regression coefficients
        print('Coefficients: ', reg.coef_)

        # variance score: 1 means perfect prediction
        print('Variance score: {}'.format(reg.score(X_test, y_test)))

        # plot for residual error

        ## setting plot style
        plt.style.use('fivethirtyeight')

        ## plotting residual errors in training data
        plt.scatter(reg.predict(X_train), reg.predict(X_train) - y_train,
                    color = "green", s = 10, label = 'Train data')
```

```
## plotting residual errors in test data
plt.scatter(reg.predict(X_test), reg.predict(X_test) - y_test,
            color = "blue", s = 10, label = 'Test data')

## plotting line for zero residual error
plt.hlines(y = 0, xmin = 0, xmax = 50, linewidth = 2)

## plotting legend
plt.legend(loc = 'upper right')

## plot title
plt.title("Residual errors")

## method call for showing the plot
plt.show()
```

```
Coefficients:  [-8.95714048e-02  6.73132853e-02  5.04649248e-02  2.18579583e+00
 -1.72053975e+01  3.63606995e+00  2.05579939e-03 -1.36602886e+00
  2.89576718e-01 -1.22700072e-02 -8.34881849e-01  9.40360790e-03
 -5.04008320e-01]
Variance score: 0.7209056672661758
```




```
In [ ]:
```

❖ Learning outcomes (What I have learnt):

1. We learned about data analysis and data handling in python.
2. We learned about various basic functions and libraries required for data analysis using python.
3. We learned graphically analyze data functions of matplotlib library in python.
4. We learned about linear regression and its implementation.

Evaluation Grid :

s.no	Parameters	Marks Obtained	Maximum Marks
1.	**Student Performance (Conduct of experiment) objectives/Outcomes.**		**12**
2.	**Viva Voce**		**10**
3.	**Submission of Work Sheet (Record)**		**8**
	Total		**30**