In [8]:
```python
#Importing all the necessary modules
import pandas as pd #reading csv file and creating a dataframe
import numpy as np
import matplotlib.pyplot as plt
#for plotting data from ur1 and trained data
%matplotlib inline
ur1 = "http://bit.ly/w-data"
df = pd.read_csv(ur1)
print("Data taken in")
df.head(10)
```

Data taken in

Out[8]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25

In [9]:
```python
#plotting the data in given i in n 2-D to understand the distribution of c
df.plot(x = "Hours",y = "Scores", style = "o")
plt.title("Hours vs percentage")
plt.xlabel("Hours studied")
plt.ylabel("percentage score")
plt.show()
```


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Hours studied
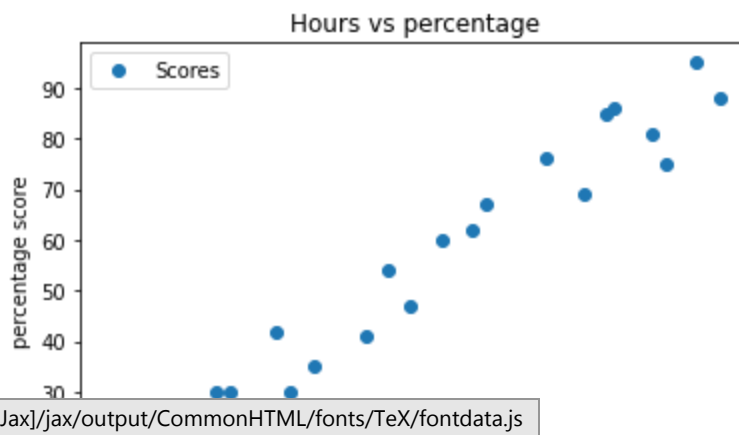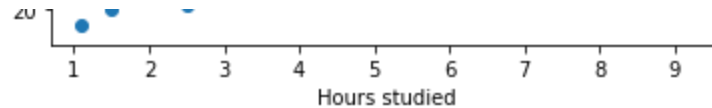
```
In [10]: #Selecting the values of data from the dataframe
         x = df.iloc[:,:-1].values
         y = df.iloc[:,1].values
```

```
In [11]: #Splitting the data values obtained into training and testing samples:
         from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_
```

```
In [12]: #Sample data taken for training are trained using Linear Regreesion model'
         from sklearn.linear_model import LinearRegression
         regressor = LinearRegression()
         regressor.fit(x_train,y_train)
         print("Trained")
```

Trained

```
In [13]: #plotting the trained regression model's output using a linear equation
         line = regressor.coef_*x + regressor.intercept_
         plt.scatter(x,y)
         plt.plot(x,line)
         plt.show()
```
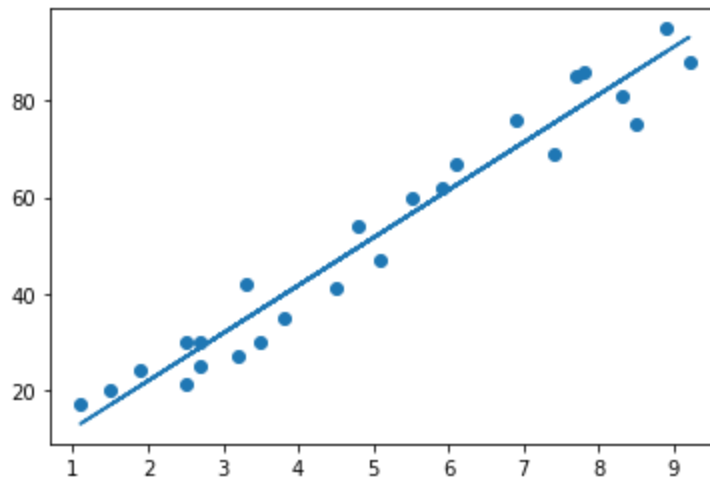


```
In [15]: y_pred = regressor.predict(x_test)
         print(x_test)
         df1 = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
         df1
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]]
```

Out[15]:

	Actual	Predicted
0	20	16.884145
1	27	33.732261
2	69	75.357018
3	30	26.794801
4	62	60.491033

In [16]:
```python
#what will the student score, if he studies for any given amount of time c
hours = 9.25
own_pred = round(regressor.predict([[9.25]]) [0], 2)
print("No of Hours={}".format(hours))
print("Predicted Score={}".format(own_pred))
```

```
No of Hours=9.25
Predicted Score=93.69
```

In [20]:

In [23]:

Out[23]: array([21, 47, 27, 75, 30, 20, 88, 60, 81, 25, 85, 62, 41, 42, 17, 95, 30,
        24, 67, 69, 30, 54, 35, 76, 86], dtype=int64)

In [ ]: