In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:

```
ds=pd.read_csv('student-mat.csv')
```

In [13]:

```
#Answer Q1

pd.set_option('display.max_columns',500)
ds.head()
```

Out[13]:

| | school | sex | age | address | famsize | Pstatus | Medu | Fedu | Mjob | Fjob | reason | guarc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | GP | F | 18 | U | GT3 | A | 4 | 4 | at_home | teacher | course | mo |
| **1** | GP | F | 17 | U | GT3 | T | 1 | 1 | at_home | other | course | fa |
| **2** | GP | F | 15 | U | LE3 | T | 1 | 1 | at_home | other | other | mo |
| **3** | GP | F | 15 | U | GT3 | T | 4 | 2 | health | services | home | mo |
| **4** | GP | F | 16 | U | GT3 | T | 3 | 3 | other | other | home | fa |

In [14]:

```
ds.columns
```

Out[14]:

```
Index(['school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fed
u',
       'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime',
       'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery',
       'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dal
c',
       'Walc', 'health', 'absences', 'G1', 'G2', 'G3'],
      dtype='object')
```

In [5]:

```
ds.shape
```

Out[5]:

```
(395, 33)
```

In [6]:

```
ds.ndim
```

Out[6]:

```
2
```

In [7]:

```
#Answer Q2
#Attributes = 33
#Dimension = 2
```

In [26]:

```
import numpy
x = pd.read_csv('student-mat.csv')
dalc = ds.iloc[:,-7].values
round(numpy.mean(dalc),2)
```

Out[26]:

```
1.48
```

In [25]:

```
import numpy
x = pd.read_csv('student-mat.csv')
walc = ds.iloc[:,-6].values
round(numpy.mean(walc),2)
```

Out[25]:

```
2.29
```

In [24]:

```
import numpy
x = pd.read_csv('student-mat.csv')
absense = ds.iloc[:,-4].values
round(numpy.mean(absense),2)
```

Out[24]:

```
5.71
```

In [15]:

```
#Answer Q3
#Dalc = 1.48
#Walc = 2.29
#absence day of = 5.71
```

In [16]:

```
numpy.amax(absense)
```

Out[16]:

```
75
```

In [17]:

```
numpy.amin(absense)
```

Out[17]:

```
0
```

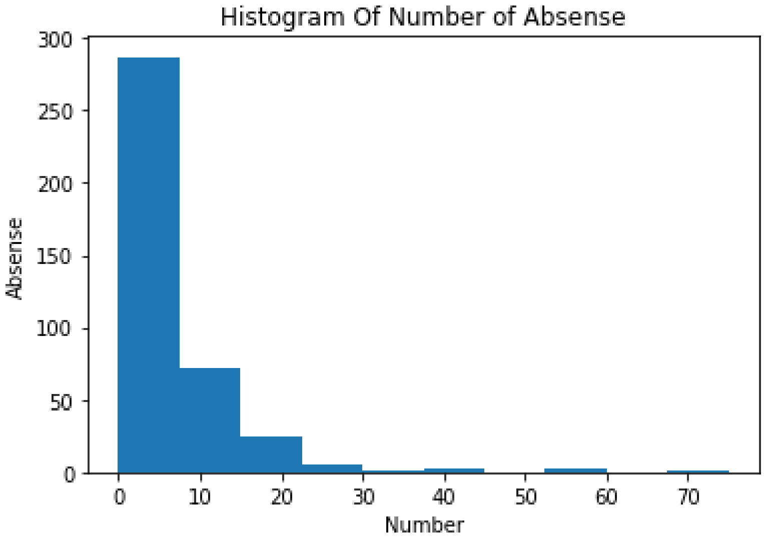
In [19]:

```
#Answer Q4
#Max = 75
#min = 0
```

In [20]:

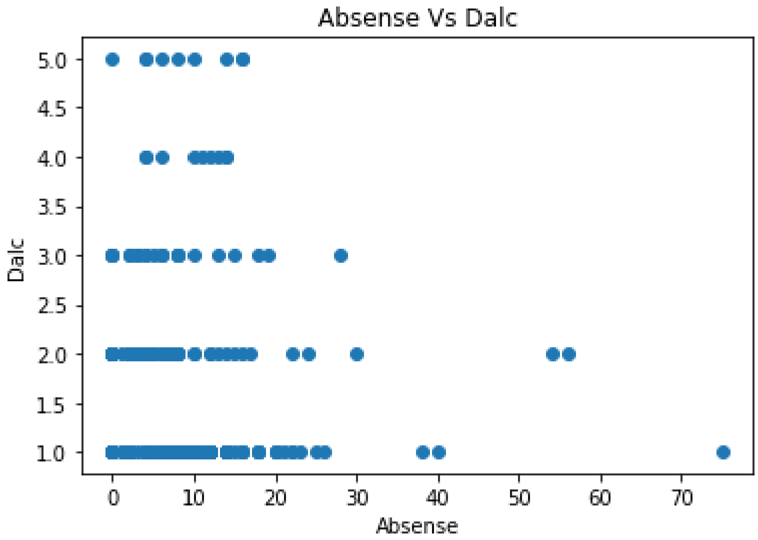
```
#Answer Q5
x=pd.read_csv('student-mat.csv')
x=ds.iloc[:,-4].values

plt.hist(x)
plt.title("Histogram Of Number of Absense")
plt.xlabel("Number")
plt.ylabel("Absense")
plt.show()
```



In [22]:

```
#Answer Q6
import matplotlib.pyplot as plt
x = absense
y = dalc
plt.scatter(x,y)
plt.title("Absense Vs Dalc")
plt.xlabel("Absense")
plt.ylabel("Dalc")
plt.show()
```



In [23]:

```
#Answer Q7
np.corrcoef(x,y)
```

Out[23]:

```
array([[1.        , 0.11190803],
       [0.11190803, 1.        ]])
```

In [27]:

```
#Answer Q8
#1. Range of days for absence = 6 days
#2. Most and Least Frequent days for absence for students = 38 days
#3. Observation on the distribution of days for absence = Absence distribution showing that
#4. Is there a linear relationship between days for absence and DALC? = NO relation
```

In [28]:

```
# SECTION B
# Naive Bayes - Step 1:

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

ds=pd.read_csv('student-mat.csv')
x=ds.iloc[:,[2,19,26,27]].values
y=ds.iloc[:,[20]].values

print(x)
print(y)
```

```
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']
 ['yes']

 ['yes']
```

In [30]:

```
#Naive Bayes - Step 5

from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X = LabelEncoder()
x[:, 1] = labelencoder_X.fit_transform(x[:, 1])
```

In [31]:

```
#Naive Bayes - Step 6
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state =
```

In [32]:

```
#Naive Bayes - Step 7
from sklearn.preprocessing import StandardScaler
sc_x=StandardScaler()
x_train=sc_x.fit_transform(x_train)
x_test=sc_x.fit_transform(x_test)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:475:
DataConversionWarning: Data with input dtype object was converted to float64
by StandardScaler.
  warnings.warn(msg, DataConversionWarning)
```

In [33]:

```
#Naive Bayes - Step 8
from sklearn.naive_bayes import GaussianNB
classifier= GaussianNB()
classifier.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:578:
DataConversionWarning: A column-vector y was passed when a 1d array was expe
cted. Please change the shape of y to (n_samples, ), for example using ravel
().
  y = column_or_1d(y, warn=True)
```

Out[33]:

```
GaussianNB(priors=None)
```

In [34]:

```
#Naive Bayes - Step 9
y_pred=classifier.predict(x_test)
```

In [35]:

```
#Naive Bayes - Step 10
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
```

In [38]:

```
print(cm)
```

```
[[ 0  7]
 [ 2 90]]
```

In [39]:

```
#Naive Bayes - Step 11
accuracy = (90/99)*100
print ('accuracy=',accuracy)
```

```
accuracy= 90.9090909090909
```

In [40]:

```
#Naive Bayes - Step 12
#Discuss = From the result it shows that most of students are agreed to have higher educati
```

In [73]:

```
# SECTION C

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

ds=pd.read_csv('student-mat.csv')
x=ds.iloc[:,[30,31]].values
y=ds.iloc[:,32].values

print(x)
print(y)
```

```
 [ 6  5]
 [10  9]
 [ 6  5]
 [ 7  5]
 [ 7  9]
 [ 6  5]
 [ 9  9]
 [14 16]
 [10  8]
 [11 12]
 [ 8  9]]
[ 6  6 10 15 10 15 11  6 19 15  9 12 14 11 16 14 14 10  5 10 15 15 16 12
  8  8 11 15 11 11 12 17 16 12 15  6 18 15 11 13 11 12 18 11  9  6 11 20
 14  7 13 13 10 11 13 10 15 15  9 16 11 11  9  9 10 15 12  6  8 16 15 10
  5 14 11 10 10 11 10  5 12 11  6 15 10  8  6 14 10  7  8 18  6 10 14 10
 15 10 14  8  5 17 14  6 18 11  8 18 13 16 19 10 13 19  9 16 14 13  8 13
 15 15 13 13  8 12 11  9  0 18  0  0 12 11  0  0  0  0 12 15  0  9 11 13
  0 11  0 11  0 10  0 14 10  0 12  8 13 10 15 12  0  7  0 10  7 12 10 16
  0 14  0 16 10  0  9  9 11  6  9 11  8 12 17  8 12 11 11 15  9 10 13  9
  8 10 14 15 16 10 18 10 16 10 10  6 11  9  7 13 10  7  8 13 14  8 10 15
```

In [74]:

```
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_state = 0
```

In [75]:

```
from sklearn.linear_model import LinearRegression
Indep_v=LinearRegression()
Indep_v.fit(x,y)
```

Out[75]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

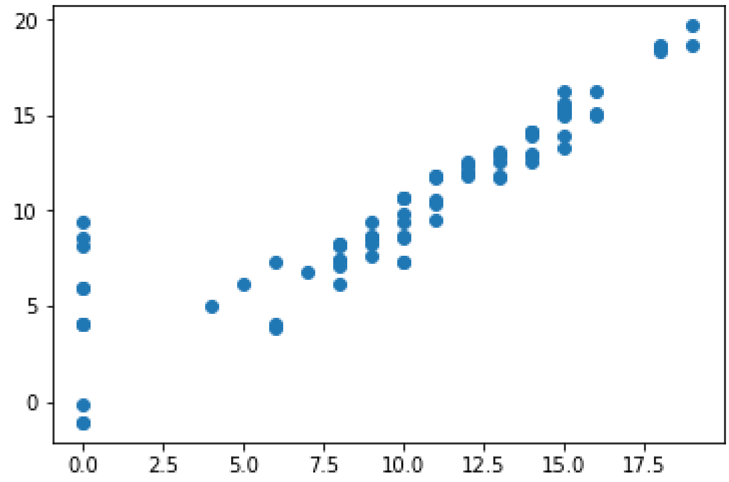
In [76]:

```
y_predict=Indep_v.predict(x_test)
print(y_predict)
```

```
[14.1318838  10.71147752  9.41807351 15.42528781 16.25888606 12.00488153
 12.83847979 10.55820894  8.27793809 15.42528781  7.29107125 11.69834436
 10.71147752  7.13780267 15.11875063 14.1318838  12.53194262  9.41807351
  8.58447526 15.11875063  5.99766724  4.02393356  6.15093583 14.96548205
 11.85161295 16.25888606  9.41807351 19.67929233 -0.1440577  12.53194262
 -1.06366921  7.29107125  7.44433984  5.0108004  13.14501696  9.87787927
 12.53194262 18.53915691  8.27793809 11.85161295  3.87066498 18.69242549
  7.44433984  8.12466951  9.5713421  -1.06366921 11.69834436  4.02393356
 11.85161295  6.8312655   8.73774385 18.38588832 10.71147752  7.59760842
 11.85161295  7.29107125  8.73774385 12.83847979 14.96548205 13.97861521
 12.99174837 13.29828554 14.1318838  15.57855639  8.58447526 15.27201922
 18.69242549 12.3114187   4.02393356  8.58447526  8.27793809 13.97861521
  6.15093583  4.02393356  8.58447526  8.12466951 10.40494035 18.53915691
  5.99766724]
```

In [77]:

```
plt.scatter(y_test,y_predict)
plt.show()
```



In [ ]:

```
#Discuss = From the scater plot showing that result is quiet OK
```