Michael Kojo Ampah 1019090042 ML

In [199]:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import scipy.stats as stats
```

In [176]:

```
df = pd.read_csv(r'C:\Users\M_Ampah\Downloads\Performance.csv')
```

In [201]:

```
#No1

print("Male =",df['gender'].value_counts()['M'])
print("Female =",df['gender'].value_counts()['F'])
```

```
Male = 19
Female = 21
```

In [177]:

```
#No2
np.average(df.age)
print("Average is :",np.average)
```

```
Average is : <function average at 0x000001B4F360C8B8>
```

#NO3

There are no missing values

In [50]:

```
#No4
range = np.max(df.english_score) - np.min(df.english_score)
print("range is :", range)
```

```
range is : 30
```

In [53]:

```
#No5
columns = [english_score
, 'science_score']
data[columns].corr()
```

Out[53]:

	english_score	science_score
english_score	1.000000	0.629384
science_score	0.629384	1.000000

In [58]:

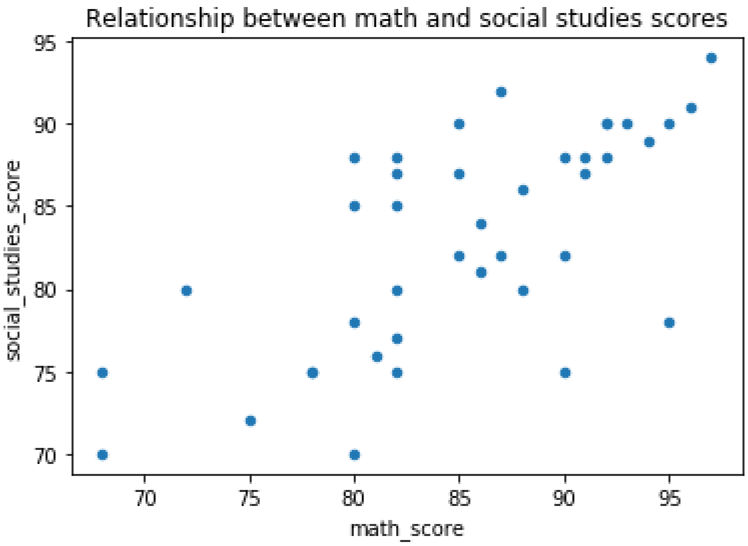
```
#No6

data.plot.scatter(x='math_score',y='social_studies_score')
plt.title('Relationship between math and social studies scores')


#Observation : Mathe scores are positively correlated to social studies
```

Out[58]:

```
Text(0.5, 1.0, 'Relationship between math and social studies scores')
```



In [98]:

```
#No7

df['Overall_score'] = df['english_score'] + df['math_score'] + df['scie
maxOverall = np.max(df.Overall_score)
HighestScoringStudent = df.loc[df['Overall_score'] == maxOverall]
print(HighestScoringStudent)
```

```
    student_id gender  age  grade_level  english_score  math_score  \
31          32      F   15           10             95          97   

    science_score  social_studies_score  Overall_score  Overall  
31             96                    94            382      382  
```

In [103]:

```
#No8
df.describe()[['english_score', 'math_score', 'science_score', 'social_
```

Out[103]:

	english_score	math_score	science_score	social_studies_score
count	40.000000	40.000000	40.000000	40.000000
mean	82.675000	85.175000	86.650000	83.000000
std	8.150468	7.242636	6.435279	6.575011
min	65.000000	68.000000	70.000000	70.000000
25%	78.000000	80.750000	83.500000	77.750000
50%	84.000000	85.500000	88.000000	84.500000
75%	89.000000	91.000000	92.000000	88.000000
max	95.000000	97.000000	96.000000	94.000000

In [104]:

```
#No9
englishScoreStd = df['english_score'].std()
print(englishScoreStd)
```

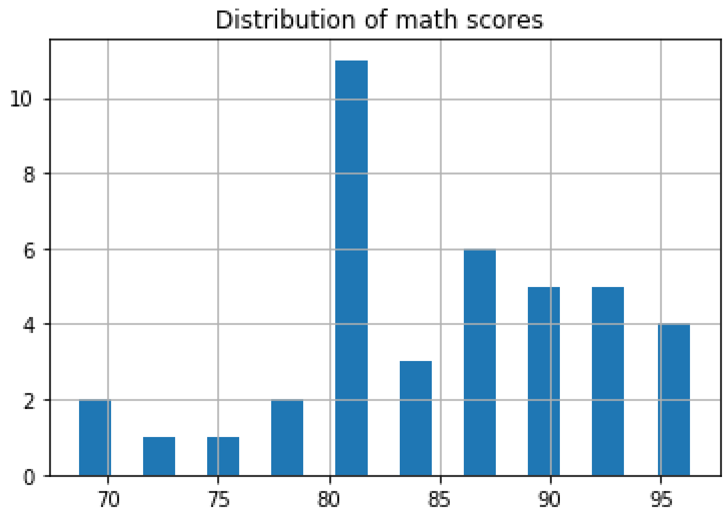
```
8.150467974609077
```

In [113]:

```
#No10
df.hist('math_score',  rwidth= 0.5)
plt.title('Distribution of math scores ')
```

Out[113]:

```
Text(0.5, 1.0, 'Distribution of math scores ')
```



In [117]:

```
#No11
np.median(df.science_score)
```

Out[117]:

```
88.0
```

In [127]:

```
#No12

q1 ,q3 = np.percentile(df['english_score'], [25, 75])
iqr = q3 -q1
print(iqr)
```

```
11.0
```

In [128]:

```
#No13
df.describe()[['english_score', 'math_score', 'science_score', 'social_

# Math has the highest overall score (97)
```

Out[128]:

	english_score	math_score	science_score	social_studies_score
count	40.000000	40.000000	40.000000	40.000000
mean	82.675000	85.175000	86.650000	83.000000
std	8.150468	7.242636	6.435279	6.575011
min	65.000000	68.000000	70.000000	70.000000
25%	78.000000	80.750000	83.500000	77.750000
50%	84.000000	85.500000	88.000000	84.500000
75%	89.000000	91.000000	92.000000	88.000000
max	95.000000	97.000000	96.000000	94.000000

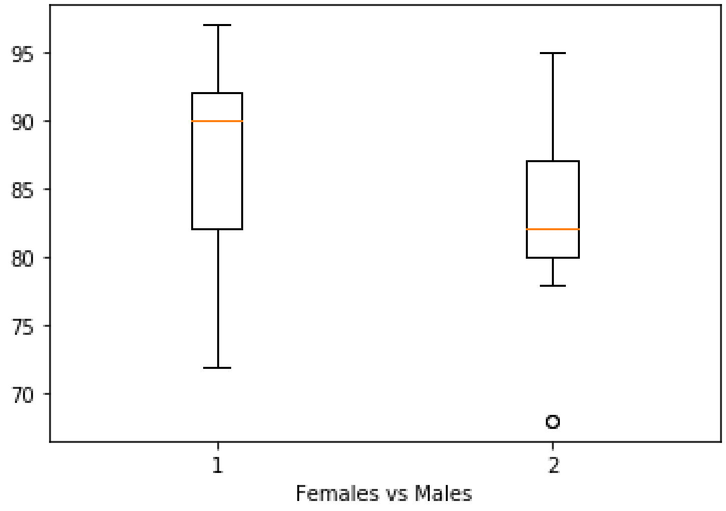
In [181]:

```
#No14
females = df[df['gender'] == 'F']
males = df[df['gender'] == 'M']
femaleMathScores = females.math_score
maleMathScores = males.math_score
plotData = [femaleMathScores, maleMathScores]
fig = plt.figure()

ax = fig.add_subplot(111)
ax.boxplot(plotData)
ax.set_xlabel('Females vs Males')
```

Out[181]:

```
Text(0.5, 0, 'Females vs Males')
```



In [192]:

```
#No15
grade, count = np.unique(df.grade_level, return_counts=True)
mode_value = np.argwhere(count == np.max(count))
print(grade[mode_value].flatten().tolist())
```

```
[11]
```

In [183]:

```
#No16
#there are no missing values
```

In [185]:

```
#No17
df.corr()
```

Out[185]:

	student_id	age	grade_level	english_score	math_score	scienc
student_id	1.000000	0.032300	-0.045710	0.387646	0.250597	0
age	0.032300	1.000000	0.965963	0.284062	0.113057	0
grade_level	-0.045710	0.965963	1.000000	0.305335	0.129292	0
english_score	0.387646	0.284062	0.305335	1.000000	0.701187	0
math_score	0.250597	0.113057	0.129292	0.701187	1.000000	0
science_score	0.159167	0.314896	0.310005	0.629384	0.615301	1
social_studies_score	0.191478	0.348830	0.406362	0.746895	0.673596	0

In [189]:

```
#No18
pd.plotting.scatter_matrix(df, alpha=0.1)
```

Out[189]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001B4
FE2408C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B4
FE64E308>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B4
FE663548>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B4
FE679B08>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B4
FE6AF548>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B4
FE6E4F08>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B4
FE71E888>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001B4
FE75EF88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B4
FE767088>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B4
FE79D248>
```

In [193]:

```
#No19
ageRange = np.max(df.age) - np.min(df.age)
print("range is :", ageRange)
```

```
range is : 3
```

In [194]:

```
#No20
df['Overall_score'] = df['english_score'] + df['math_score'] + df['scie
maxOverall = np.min(df.Overall_score)
HighestScoringStudent = df.loc[df['Overall_score'] == maxOverall]
print(HighestScoringStudent)
```

```
    student_id gender  age  grade_level  english_score  math_score  \
12          13      M   14            9             65          68   

    science_score  social_studies_score  Overall_score  
12             75                    70            278  
```

In [196]:

```
#N021
meanMathScore = np.mean(df.math_score)
medianMathScore = np.median(df.math_score)
print("Mean math score: ", meanMathScore)
print("Median math score: ", medianMathScore)
print("Difference: ", meanMathScore - medianMathScore)
# data is not skewed as the difference is negligle
```

```
Mean math score:  85.175
Median math score:  85.5
Difference:  -0.3250000000000284
```

In [200]:

```
#No22
df['social_studies_zscore'] = stats.zscore(df['social_studies_score'])
student15 = df.loc[df['student_id'] == 15]
print(student15)
```

```
    student_id gender  age  grade_level  english_score  math_score  \
14          15      F   15           10             92          90   

    science_score  social_studies_score  Overall_score  social_studies
_zscore  
14             70                    82            334              -
0.154029  
```

In [ ]:

```
```