Andoch ML_Assignment2

May 20, 2023

[6]:
```python
from matplotlib import pyplot as plot
```

[7]:
```python
import numpy as np
```

[8]:
```python
import pandas as pd
```

[9]:
```python
import scipy.stats as stats
```

[10]:
```python
document = pd.read_csv(r'C:\Users\bonin\Downloads\Performance.csv')
```

[11]:
```python
df = pd.DataFrame(document)
```

[12]:
```python
df.head()
```

[12]:
```
   student_id gender  age  grade_level  english_score  math_score  \
0           1      M   16           11             80          90
1           2      F   15           10             70          80
2           3      F   17           12             88          72
3           4      M   16           11             65          82
4           5      F   14            9             75          88

   science_score  social_studies_score
0             85                    75
1             92                    78
2             90                    80
3             78                    85
4             85                    80
```

[13]:
```python
#Question 1
numberOfStudents = len(df.index)
print(numberOfStudents)
```

```
40
```

[14]:
```python
#Question 2
averageAgeOfStudents = np.mean(document.age)
print(averageAgeOfStudents)
```

```
15.675
```

[15]:

```
#Question 3
numberOfMissingRecords = df.isna().sum().sum()
print(numberOfMissingRecords)
```

```
0
```

[16]:

```
#Question 4
englishScores = np.array(document.english_score)
rangeOfEnglishScores = np.max(englishScores) - np.min(englishScores)
print(rangeOfEnglishScores)
```

```
30
```

[17]:

```
#Question 5
correlationCoefficient = df['english_score'].corr(df['science_score'])
print(correlationCoefficient)
```

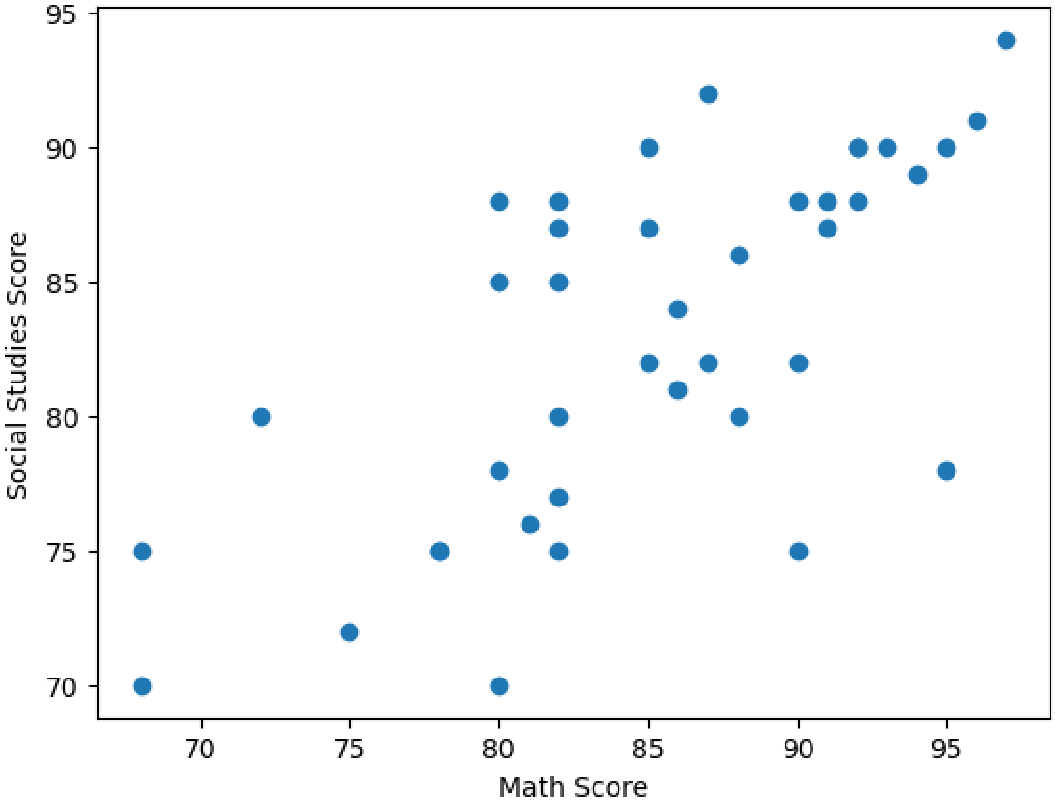
```
0.6293841680797964
```

[18]:

```
#Question 6
plot.scatter(df.math_score, df.social_studies_score)
plot.xlabel("Math Score")
plot.ylabel("Social Studies Score")

#Observation: math scores are positively correlated with social studies scores
```

[18]:

```
Text(0, 0.5, 'Social Studies Score')
```



[19]:
```
#Question 7
df['overall_score'] = df['math_score'] + df['english_score'] +␣
 ↪df['science_score'] + df['social_studies_score']
maxOverallScore = np.max(df.overall_score)
highestScoringStudent = df.loc[df['overall_score'] == maxOverallScore]
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
31             96                    94            382
```

[20]:
```
#Question 8
```

[21]:
```
#Question 9
englishScoresSTD = df['english_score'].std()
print(englishScoresSTD)
```

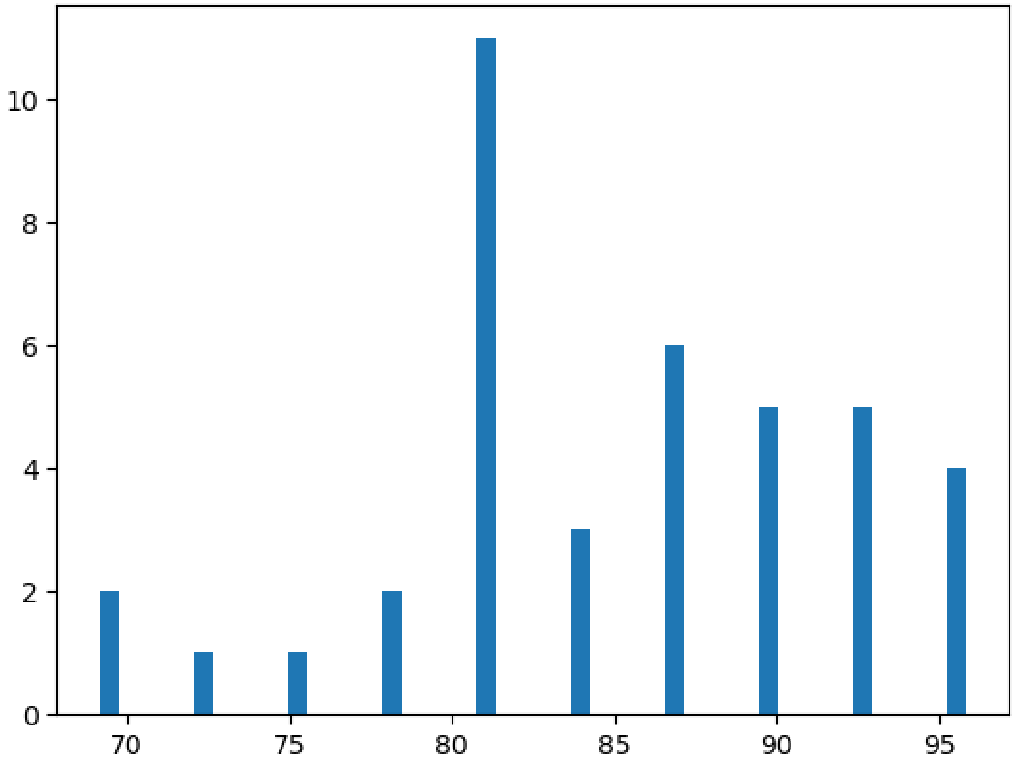
```
8.150467974609077
```

[22]:

```
#Question 10
plot.hist(df.math_score, rwidth=0.2)
#Observations: most people scored above 80, with very few people scoring below␣
 ↪80. 82 was the most common score
```

[22]:

```
(array([ 2.,  1.,  1.,  2., 11.,  3.,  6.,  5.,  5.,  4.]),
 array([68. , 70.9, 73.8, 76.7, 79.6, 82.5, 85.4, 88.3, 91.2, 94.1, 97. ]),
 <BarContainer object of 10 artists>)
```



[23]:

```
#Question 11
medianScienceScore = np.median(document.science_score)
print(medianScienceScore)
```

```
88.0
```

[24]:

```
#Question 12
percentile75 = np.percentile(document.english_score, 75)
percentile25 = np.percentile(document.english_score, 25)
englishIQR = percentile75 - percentile25
print(englishIQR)
```

```
11.0
```

[25]:

```
#Question 13
df.describe()[['english_score', 'math_score', 'science_score',␣
 ↪'social_studies_score' ]]

# Math has the highest overall score
```

[25]:

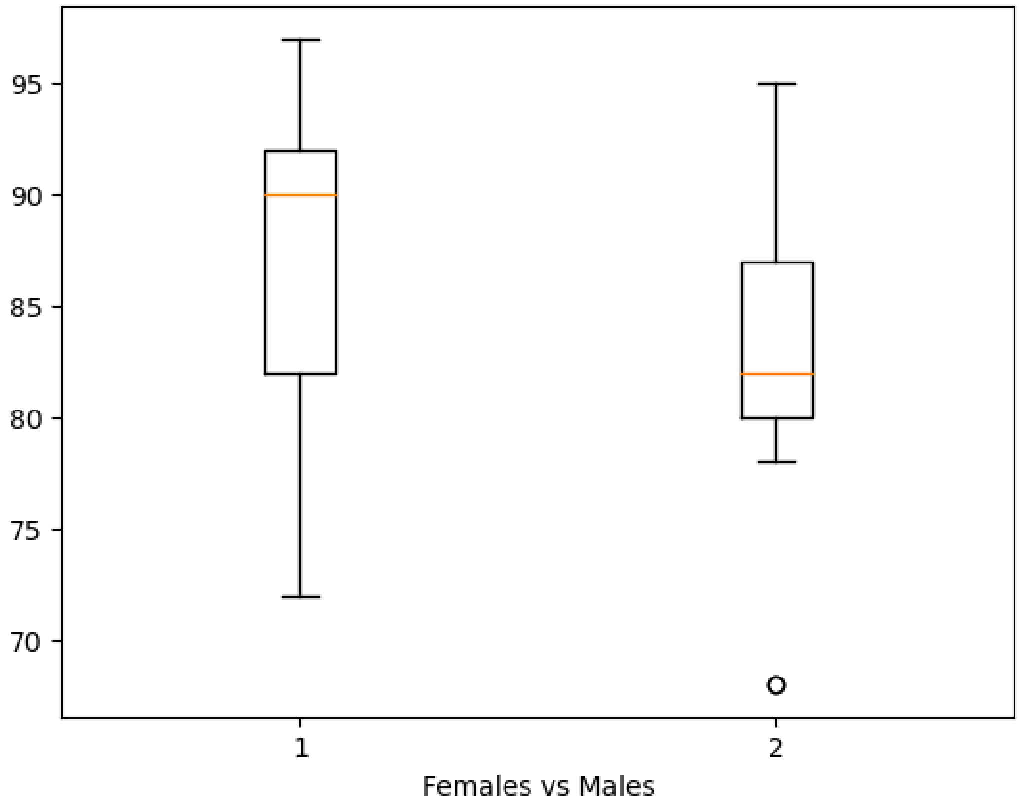
```
       english_score  math_score  science_score  social_studies_score
count      40.000000   40.000000      40.000000             40.000000
mean       82.675000   85.175000      86.650000             83.000000
std         8.150468    7.242636       6.435279              6.575011
min        65.000000   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
```

[26]:

```
#Question 14
females = df[df['gender'] == 'F']
males = df[df['gender'] == 'M']
femaleMathScores = females.math_score
maleMathScores = males.math_score
plotData = [femaleMathScores, maleMathScores]
fig = plot.figure()

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

[26]:

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



[27]:
```
#Question 15
grade, count = np.unique(document.grade_level, return_counts=True)
mode_value = np.argwhere(count == np.max(count))
print(grade[mode_value].flatten().tolist())
```

```
[11]
```

[28]:
```
#Question 16
# no missing values
```

[29]:
```
#Question 17
df.corr()
```

```
C:\Users\bonin\AppData\Local\Temp\ipykernel_10112\3156044343.py:2:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  df.corr()
```

[29]:
```
                    student_id       age  grade_level  english_score  \
student_id            1.000000  0.032300    -0.045710       0.387646
age                   0.032300  1.000000     0.965963       0.284062
```

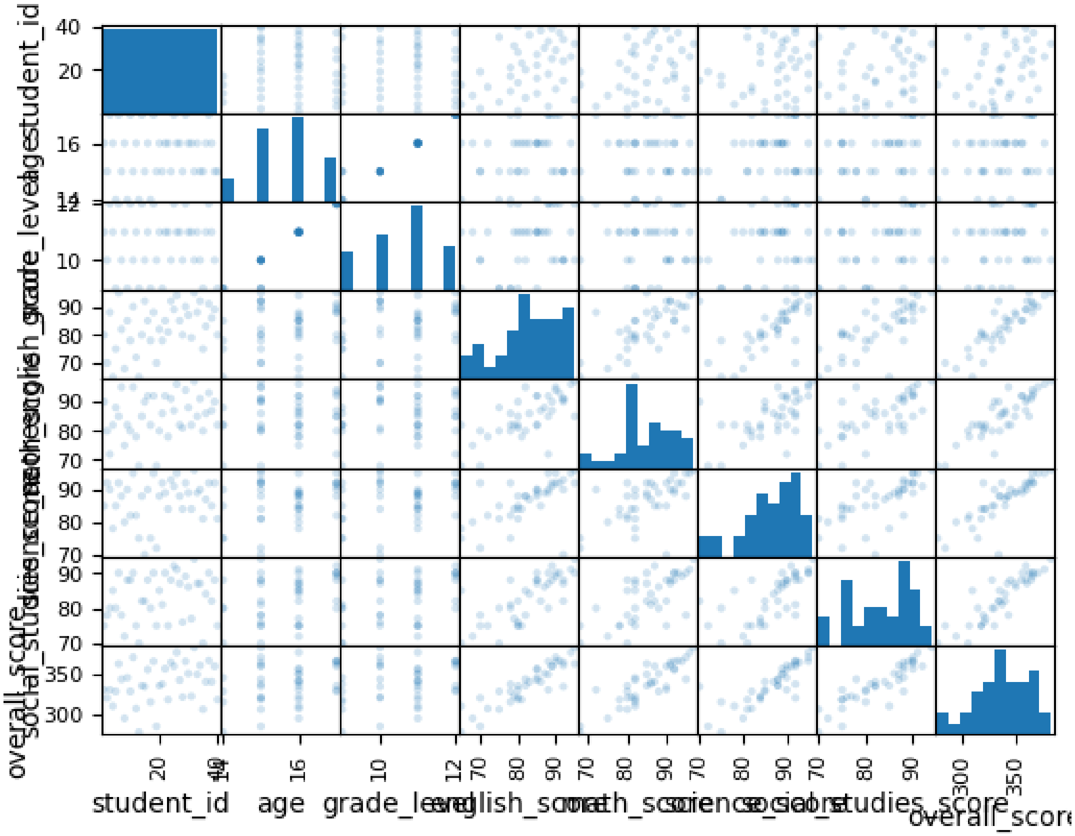
```
grade_level            -0.045710  0.965963     1.000000       0.305335
english_score           0.387646  0.284062     0.305335       1.000000
math_score              0.250597  0.113057     0.129292       0.701187
science_score           0.159167  0.314896     0.310005       0.629384
social_studies_score    0.191478  0.348830     0.406362       0.746895
overall_score           0.293684  0.301629     0.327436       0.897919

                      math_score  science_score  social_studies_score  \
student_id              0.250597       0.159167              0.191478
age                     0.113057       0.314896              0.348830
grade_level             0.129292       0.310005              0.406362
english_score           0.701187       0.629384              0.746895
math_score              1.000000       0.615301              0.673596
science_score           0.615301       1.000000              0.671446
social_studies_score    0.673596       0.671446              1.000000
overall_score           0.863773       0.826952              0.884650

                      overall_score
student_id                 0.293684
age                        0.301629
grade_level                0.327436
english_score              0.897919
math_score                 0.863773
science_score              0.826952
social_studies_score       0.884650
overall_score              1.000000
```

[30]:

```
#Question 18
pd.plotting.scatter_matrix(df, alpha=0.2)
```

[30]:

```
array([[<Axes: xlabel='student_id', ylabel='student_id'>,
        <Axes: xlabel='age', ylabel='student_id'>,
        <Axes: xlabel='grade_level', ylabel='student_id'>,
        <Axes: xlabel='english_score', ylabel='student_id'>,
        <Axes: xlabel='math_score', ylabel='student_id'>,
        <Axes: xlabel='science_score', ylabel='student_id'>,
        <Axes: xlabel='social_studies_score', ylabel='student_id'>,
        <Axes: xlabel='overall_score', ylabel='student_id'>],
       [<Axes: xlabel='student_id', ylabel='age'>,
        <Axes: xlabel='age', ylabel='age'>,
        <Axes: xlabel='grade_level', ylabel='age'>,
        <Axes: xlabel='english_score', ylabel='age'>,
        <Axes: xlabel='math_score', ylabel='age'>,
        <Axes: xlabel='science_score', ylabel='age'>,
        <Axes: xlabel='social_studies_score', ylabel='age'>,
        <Axes: xlabel='overall_score', ylabel='age'>],
       [<Axes: xlabel='student_id', ylabel='grade_level'>,
        <Axes: xlabel='age', ylabel='grade_level'>,
        <Axes: xlabel='grade_level', ylabel='grade_level'>,
        <Axes: xlabel='english_score', ylabel='grade_level'>,
        <Axes: xlabel='math_score', ylabel='grade_level'>,
        <Axes: xlabel='science_score', ylabel='grade_level'>,
        <Axes: xlabel='social_studies_score', ylabel='grade_level'>,
        <Axes: xlabel='overall_score', ylabel='grade_level'>],
       [<Axes: xlabel='student_id', ylabel='english_score'>,
        <Axes: xlabel='age', ylabel='english_score'>,
        <Axes: xlabel='grade_level', ylabel='english_score'>,
        <Axes: xlabel='english_score', ylabel='english_score'>,
        <Axes: xlabel='math_score', ylabel='english_score'>,
        <Axes: xlabel='science_score', ylabel='english_score'>,
        <Axes: xlabel='social_studies_score', ylabel='english_score'>,
        <Axes: xlabel='overall_score', ylabel='english_score'>],
       [<Axes: xlabel='student_id', ylabel='math_score'>,
        <Axes: xlabel='age', ylabel='math_score'>,
        <Axes: xlabel='grade_level', ylabel='math_score'>,
        <Axes: xlabel='english_score', ylabel='math_score'>,
        <Axes: xlabel='math_score', ylabel='math_score'>,
        <Axes: xlabel='science_score', ylabel='math_score'>,
        <Axes: xlabel='social_studies_score', ylabel='math_score'>,
        <Axes: xlabel='overall_score', ylabel='math_score'>],
       [<Axes: xlabel='student_id', ylabel='science_score'>,
        <Axes: xlabel='age', ylabel='science_score'>,
        <Axes: xlabel='grade_level', ylabel='science_score'>,
        <Axes: xlabel='english_score', ylabel='science_score'>,
        <Axes: xlabel='math_score', ylabel='science_score'>,
        <Axes: xlabel='science_score', ylabel='science_score'>,
        <Axes: xlabel='social_studies_score', ylabel='science_score'>,
        <Axes: xlabel='overall_score', ylabel='science_score'>],
       [<Axes: xlabel='student_id', ylabel='social_studies_score'>,
        <Axes: xlabel='age', ylabel='social_studies_score'>,
        <Axes: xlabel='grade_level', ylabel='social_studies_score'>,
        <Axes: xlabel='english_score', ylabel='social_studies_score'>,
        <Axes: xlabel='math_score', ylabel='social_studies_score'>,
        <Axes: xlabel='science_score', ylabel='social_studies_score'>,
        <Axes: xlabel='social_studies_score', ylabel='social_studies_score'>,
        <Axes: xlabel='overall_score', ylabel='social_studies_score'>],
       [<Axes: xlabel='student_id', ylabel='overall_score'>,
        <Axes: xlabel='age', ylabel='overall_score'>,
        <Axes: xlabel='grade_level', ylabel='overall_score'>,
        <Axes: xlabel='english_score', ylabel='overall_score'>,
        <Axes: xlabel='math_score', ylabel='overall_score'>,
        <Axes: xlabel='science_score', ylabel='overall_score'>,
        <Axes: xlabel='social_studies_score', ylabel='overall_score'>,
        <Axes: xlabel='overall_score', ylabel='overall_score'>]],
       dtype=object)
```



[31]:
```
#Question 19
agesArray = np.array(document.age)
rangeOfAges = np.max(agesArray) - np.min(agesArray)
print(rangeOfAges)
```

```
3
```

[32]:
```
#Question 20
minOverallScore = np.min(df.overall_score)
lowestScoringStudent = df.loc[df['overall_score'] == minOverallScore]
print(lowestScoringStudent)
```

```
    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
```

[33]:

```
#Question 21
meanMathScore = np.mean(document.math_score)
medianMathScore = np.median(document.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:  -0.32500000000000284
```

[34]:

```
#Question 22
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
```

[ ]:

[ ]:

[ ]: