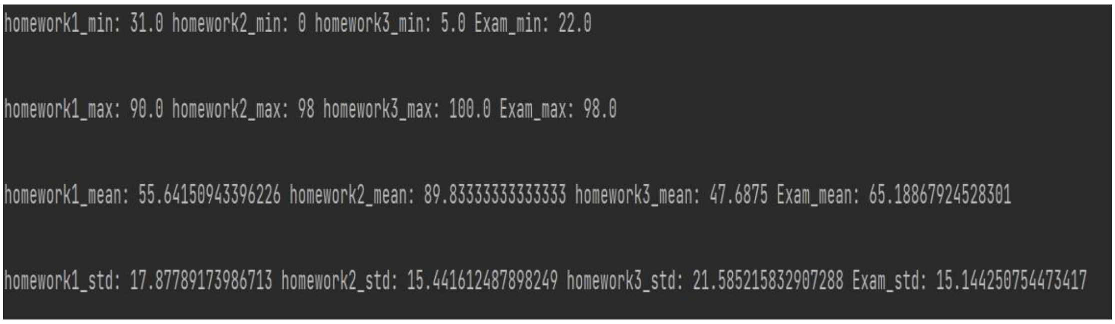
Question 1

1. min, max, mean and standard deviation for each column

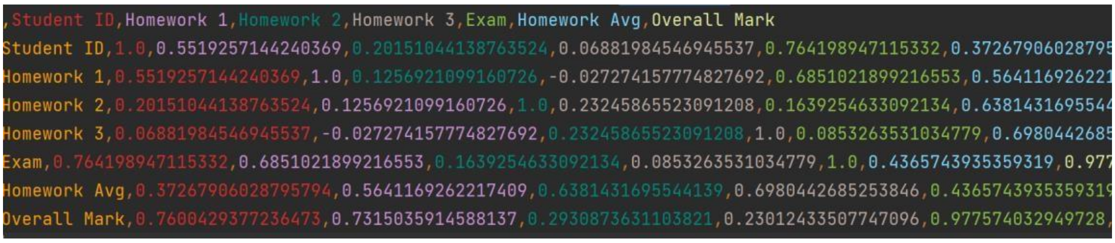


```
homework1_min: 31.0 homework2_min: 0 homework3_min: 5.0 Exam_min: 22.0

homework1_max: 90.0 homework2_max: 98 homework3_max: 100.0 Exam_max: 98.0

homework1_mean: 55.64150943396226 homework2_mean: 89.83333333333333 homework3_mean: 47.6875 Exam_mean: 65.18867924528301

homework1_std: 17.87789173986713 homework2_std: 15.441612487898249 homework3_std: 21.585215832907288 Exam_std: 15.144250754473417
```

2. #Add an additional named as ‘Homework Avg’ for the average homework mark for each student.Keep one decimal place

```
   grade['Homework Avg'] = grade[['Homework 1','Homework 2','Homework
3']].mean(1).round(1)
  # add an additional column named 'Overall Mark' for the overall folded mark.
grade['Overall Mark'] = grade['Homework Avg']*0.25+grade['Exam']*0.75
```

3.



```
,Student ID,Homework 1,Homework 2,Homework 3,Exam,Homework Avg,Overall Mark
Student ID,1.0,0.5519257144240369,0.20151044138763524,0.06881984546945537,0.764198947115332,0.37267906028795
Homework 1,0.5519257144240369,1.0,0.1256921099160726,-0.027274157774827692,0.6851021899216553,0.564116926221
Homework 2,0.20151044138763524,0.1256921099160726,1.0,0.23245865523091208,0.1639254633092134,0.638143169554
Homework 3,0.06881984546945537,-0.027274157774827692,0.23245865523091208,1.0,0.0853263531034779,0.6980442685
Exam,0.764198947115332,0.6851021899216553,0.1639254633092134,0.0853263531034779,1.0,0.4365743935359319,0.977
Homework Avg,0.37267906028795794,0.5641169262217409,0.6381431695544139,0.6980442685253846,0.436574393535931
Overall Mark,0.7600429377236473,0.7315035914588137,0.2930873631103821,0.23012433507747096,0.977574032949728,
```

As can been seen from the correlation matrix, Exam and Homework Avg are positively correlated. So, the higher the student's usual homework score, the more likely it is to get a higher score in the exam.

4.

Missing values in the data can be replaced by 0. Because most of the missing data is due to the unsubmitted of homework. For some special circumstances, such as illness, etc., the score manager can manually enter the score (Supplementary homework).

5.

Use apply() method of pandas package. Writing a function to grade according to Overall Mark.

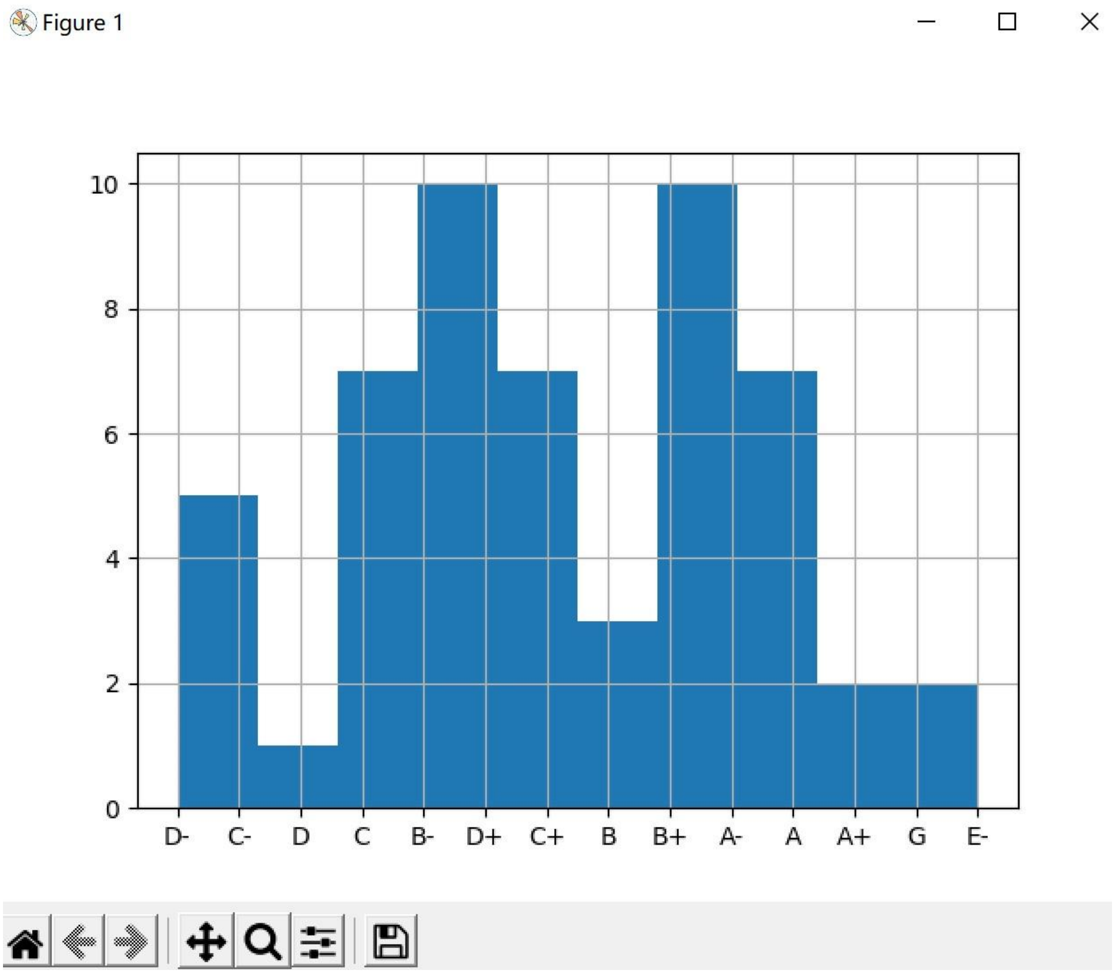


Figure 1
10
8
6
4
2
0
D-
C-
D
C
B-
D+
C+
B
B+
A-
A
A+
G
E-

Question 2

# 1 copy the attribute 3 and 12 data['Original Input3'] = data['Input3'] data['Original Input12'] = data['Input12']

# 2 z-score transformation,z = (x – μ) / σ data['Input3'] = (data['Original Input3']-data['Original Input3'].mean()) / data['Original Input3'].std()

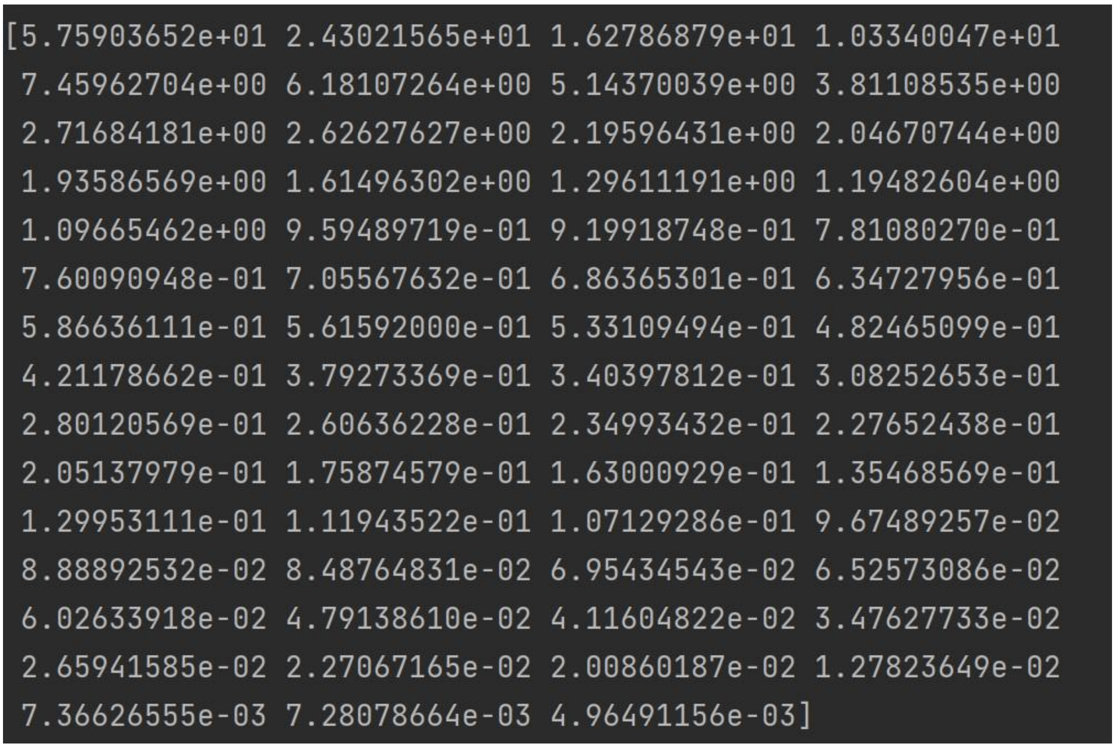
# 3 0-1 transformation (z-min)/(max-min) data['Input12'] = (data['Original Input12']-data['Original Input12'].min()) / (data['Original Input12'].max() - data['Original Input12'].min())

# 4 Calculate the average of all the attributes from “Input1” to “Input12” as new attributes

Data['AverageInput'] = data[['Input1','Input2','Input3','Input4','Input5','Input6','Input7','Input8','Input9' ,'Input10','Input11','Input12']].mean(1)

Question 3

1. **explained_variance_**



```
[5.75903652e+01 2.43021565e+01 1.62786879e+01 1.03340047e+01
 7.45962704e+00 6.18107264e+00 5.14370039e+00 3.81108535e+00
 2.71684181e+00 2.62627627e+00 2.19596431e+00 2.04670744e+00
 1.93586569e+00 1.61496302e+00 1.29611191e+00 1.19482604e+00
 1.09665462e+00 9.59489719e-01 9.19918748e-01 7.81080270e-01
 7.60090948e-01 7.05567632e-01 6.86365301e-01 6.34727956e-01
 5.86636111e-01 5.61592000e-01 5.33109494e-01 4.82465099e-01
 4.21178662e-01 3.79273369e-01 3.40397812e-01 3.08252653e-01
 2.80120569e-01 2.60636228e-01 2.34993432e-01 2.27652438e-01
 2.05137979e-01 1.75874579e-01 1.63000929e-01 1.35468569e-01
 1.29953111e-01 1.11943522e-01 1.07129286e-01 9.67489257e-02
 8.88892532e-02 8.48764831e-02 6.95434543e-02 6.52573086e-02
 6.02633918e-02 4.79138610e-02 4.11604822e-02 3.47627733e-02
 2.65941585e-02 2.27067165e-02 2.00860187e-02 1.27823649e-02
 7.36626555e-03 7.28078664e-03 4.96491156e-03]
```

2.3

```
# The spacing of each bin is the same
a = pd.cut(pca.explained_variance_, 10)
# The frequencies of the numbers contained are the same
b = pd.qcut(pca.explained_variance_, 10)
```