# Homework Assignment #3

## 1 Problem

### 1.1 problem definition

1. opens a provided file (CEE_220_Scores.txt)
2. reads and analyzes the header line to identify what columns contain "Assignment"s, "Lab"s, "Midterm"s, and the "Final" so you can organize the subsequent data lines into the respective four groups.
3. reads the second line and used information from 2. to compute target scores for each group
4. loops through the remaining lines (list of students) and computes total points assigned for each of the four groups.

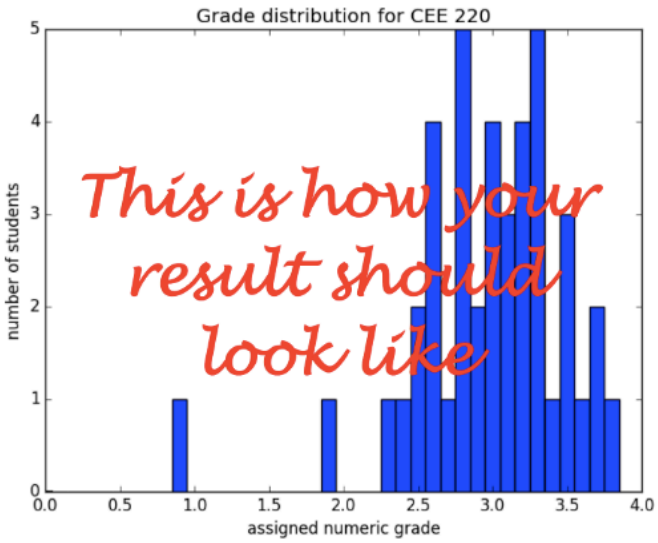
- compute a weighted score as

  score = $\sum (weight\ for\ the\ group)\ \frac{(student\ score\ for\ the\ group)}{(target\ score\ for\ the\ group)} \leq 1.0$

  with weights = {'HW':0.25, 'Labs':0.05, "Midterms":0.40, 'Final':0.30}
- compute the grade as

  $$grade\ = \frac{(score\ -\ 0.20)}{0.20}$$

  where the grade needs to be rounded to one decimal. Grades below 0.7 should be corrected to 0.0 (UW rules)
- store name, group point sums (the four discussed above), weighted score, and grade in a dictionary with sensible keys (name for the entry) and collect these in a growing list of student results.
- Perform a statistical analysis of all assigned grades using pyplot's hist function. Use bins in 0.1 intervals (one by grade; limits at 0.05, 0.15, ...) and do not normalize the plot but rather show the actual number of students who received a particular grade. Check the online documentation for additional plot control features to add axis labels and title as shown in the following example figure.




Use the pyplot.savefig() method to automatically generate a pdf (mac) or png (windows) file for your report.

- Include sensible error handling for missing data, IO-errors, or other likely problems.
- After testing with the dataset in CEE_220_Scores.txt and comparing results to the above image, run your code on the alternative file CEE_220_AlternativeList.txt and present the statistics (figure) in your report. WARNING: the column order in this file differs from the initial file. This is to test your ability to identify columns as requested under point 2.

## 1.2 Algorithm

1. Open the file and read lines of the file and consider the IOError.
2. Identify the first line and split the column by "\t". Create the four lists for each section. Find the keywords that contain the section name and append that column to the respective lists. Use a dictionary to show four sections.
3. Read the second line and regard it as the target line. Split that line by "\t" in the same way and use the for loop to sum the scores in the same section and return it to a new dictionary.
4. Create a dictionary of weights. Use for loop to read the file line by line. Conduct the process in the similar way as mentioned in part 3. Convert the value into float type number. Indicate the ValueError to show that there is a missing value in which line and which column. Calculate the grade by using the weighted score and round the grades according to the requirement. Build a dictionary to store the student's name and several sections of his scores and append it to the student list that contains all the students' name and groups of scores. At last, create a grade list to get each student's grade from the list of dictionary and plot the histogram.
5. Change the filename to test the "alternativelist.txt".

## 1.3 Implementation

```
import numpy as np
import matplotlib.pyplot as plt
from astropy.io.fits.util import first
from dask.array.chunk import arange


## Q1
try:
    #f=open("CEE_220_Scores.txt","r")
    f=open("CEE_220_AlternativeList.txt","r")
    lines=f.readlines()
except IOError:
    print "Could not open the file!"
except:
    print "Other error"


###Q2
firstline=lines[0].split('\t')
assignment=[]
lab=[]
midterm=[]
final=[]
for i in range(1,len(firstline)):
    if firstline[i].find('Assignment')>=0:
        assignment.append(i)
    elif firstline[i].find('Lab')>=0:
        lab.append(i)
    elif firstline[i].find('Midterm')>=0:
        midterm.append(i)
    elif firstline[i].find('Final')>=0:
        final.append(i)

print assignment,lab,midterm,final
dictionary=dict(AS=assignment,LB=lab,MID=midterm,FN=final)
print dictionary
tar=dict()
```

```
## Q3
targetline=lines[1].split('\t')
for j in dictionary.keys():
    s=0
    for m in dictionary.get(j):
        try:
            s=s+float(targetline[m])
        except ValueError:
            print "Data error"
        print "Score for group ", j,"is ",s
        tar[j]=s
print tar

for j in range(1,len(targetline)):
    if firstline[j].find('Bonus Assignment') >= 0:
        tar['AS'] -= float(targetline[j])

##Q4
weights1=dict(AS=0.25,LB=0.05,MID=0.4,FIN=0.3)
student=[]
result=dict()
for i in np.arange(2,len(lines)):
    l=lines[i].split('\t')
    target_as=0
    target_mt=0
    target_lab=0
    target_fin=0
    for j in assignment:
        try:
            target_as=target_as+float(l[j])
        except ValueError:
            print "missing data for column", j,'of',l[0]
    for j in lab:
        try:
            target_lab=target_lab+float(l[j])
        except ValueError:
            print "missing data for column", j,'of',l[0]
    for j in midterm:
        try:
            target_mt=target_mt+float(l[j])
        except ValueError:
            print "missing data for column", j,'of',l[0]
```

```
      for j in final:
        try:
            target_fin=target_fin+float(l[j])
        except ValueError:
            print "missing data for column", j,'of',l[0]


      weighted_score=target_as/tar['AS']*0.25+target_lab/tar['LB']*0.
      05+target_mt/tar['MID']*0.40+target_fin/tar['FN']*0.30
    grade=(weighted_score-0.2)/0.20
    if grade<0.7:
        grade=0
    else:
        grade=round(grade,1)
    result = dict(student=l[0], AS_result=target_as,
LB_result=target_lab, MID_result=target_mt, FN_result=target_fin,
Weightedscore=weighted_score, Grade=grade)
    student.append(result)

print student

grade_list = []
for s in student:
    grade_list.append(s.get('Grade'))
print grade_list

plt.hist(grade_list, bins=np.arange(0.05, 4.1, 0.1))
plt.xlabel('assigned numberic grade')
plt.ylabel('number of students')
plt.title('Grade Distribution for CEE200')
plt.savefig('plot.pdf')
plt.show()
```

## 1.4 Validation

**Procedure**:

1. Test IOError
2. Print the index that belongs to same section and print the components of each main section.
3. Print the total score of four sections for the second line of the file and check the sum number with the calculator.
4. Print the missing value of the data set to see whether the code finds the NAs in the right way and check it in the original txt file.
5. Print groups of dictionaries in a list and extract the value of grade in each dictionary.
6. The plot for CEE_200_Scores.txt
7. The plot for CEE_200_Alternativelist.txt

### Validation code& Output

1.Test IOError

Code:

```
## Q1
try:
    #f=open("CEE_220_Scores.txt","r")
    f=open("CEE_22_AlternativeList.txt","r")
    lines=f.readlines()
except IOError:
    print "Could not open the file!"
except:
    print "Other error"
```

Result:

```
Console
<terminated> hw3.py [/Users/helloRC/anaconda2/anaconda/bin/pythonw]
Traceback (most recent call last):
  File "/Users/helloRC/eclipse-workspace/cee505_hw1/src/hw3.py", line 24, in <module>
Could not open the file!
    firstline=lines[0].split('\t')
NameError: name 'lines' is not defined
```

2.Print the index that belongs to same section and print the components of each main section.

Code:

```
print assignment,lab,midterm,final
dictionary=dict(AS=assignment,LB=lab,MID=midterm,FN=final)
print dictionary
```

Result:

```
<terminated> hw3.py [/Users/helloRC/anaconda2/anaconda/bin/pythonw]
[1, 2, 3, 4, 5, 6, 7, 8, 9] [10, 11, 12, 13, 14, 15, 16, 17, 18, 19] [20, 21] [22]
{'AS': [1, 2, 3, 4, 5, 6, 7, 8, 9], 'LB': [10, 11, 12, 13, 14, 15, 16, 17, 18, 19], 'MID': [20, 21], 'FN': [22]}
```

3. Print the total score of four sections for the second line of the file and check the sum number with the calculator.

```
print tar
print tar['AS'],tar['LB']
{'AS': 530.0, 'LB': 100.0, 'MID': 200.0, 'FN': 100.0}
530.0 100.0
```

AS=sum(assignment1-8)+bonus assignment=60+60+70+80+50+70+60+80+0=535

LB=Sum(Lab1-10)= 10+10+10+10+10+10+10+10+10+10=100

MID=Midterm1+Midterm2=100+100=200

FN=100

4. Print the missing value of the data set to see whether the code finds the NAs in the right way and check it in the original txt file.

Code:

```
for j in assignment:
    try:
        target_as=target_as+float(l[j])
    except ValueError:
        print "missing data for column", j,'of',l[0]
for j in lab:
    try:
        target_lab=target_lab+float(l[j])
    except ValueError:
        print "missing data for column", j,'of',l[0]
for j in midterm:
    try:
        target_mt=target_mt+float(l[j])
    except ValueError:
        print "missing data for column", j,'of',l[0]
for j in final:
    try:
        target_fin=target_fin+float(l[j])
    except ValueError:
        print "missing data for column", j,'of',l[0]
```

Result:

```
missing data for column 3 of "Queen, Elizabeth"
missing data for column 5 of "Queen, Elizabeth"
missing data for column 9 of "Queen, Elizabeth"
missing data for column 22 of "Queen, Elizabeth"
missing data for column 12 of "Paul, Newman"
missing data for column 10 of "Hans, Schmidt"
missing data for column 11 of "Carolle, King"
missing data for column 16 of "Carolle, King"
missing data for column 18 of "Carolle, King"
missing data for column 18 of "Ginghis, Khan"
missing data for column 10 of "Lt, Ohura"
```

Check it in txt. file:

| Student | Homework A | Homework A | Homework A | Homework A | Homework A | Homework A | Homework A | Homework A | Bonus Assi | Lab #1 | Lab #2 | Lab #3 | La |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Points Pos | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 10 | |
| Sam, Cook | 60 | 60 | 67.5 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 10 | |
| Charlie, B | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 50 | 10 | 10 | 10 | |
| Peter, Pan | 60 | 60 | 70 | 77.5 | 50 | 61 | 60 | 80 | 25 | 10 | 10 | 10 | |
| Robbie, Wi | 57 | 47 | 64 | 80 | 37.5 | 57 | 60 | 70 | 55 | 10 | 10 | 10 | |
| Sammy, Dav | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 22 | 10 | 10 | 10 | |
| Frank, Sina | 54 | 46 | 65 | 77.5 | 50 | 70 | 60 | 80 | 23.5 | 10 | 10 | 10 | |
| Billy, Joe | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 11.5 | |
| Queen, Eli | 60 | 60 | | 49 | | 0 | 49 | 0 | | 10 | 10 | 10 | |
| Jimmy, Car | 60 | 60 | 70 | 80 | 40 | 66 | 55 | 60 | 15 | 10 | 10 | 10 | |
| Paul, Simo | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 31 | 10 | 10 | 10 | |
| Jane, Doe | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 32 | 10 | 10 | 10 | |
| Billy, The | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 12 | |
| Don, DeLil | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 10 | |
| Ronald, Re | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 49 | 10 | 11 | 9 | |
| Franklyn, I | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 10 | |
| Bill, Clin | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 10 | |
| Paul, Newm | 60 | 52.5 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | | |
| Robert, Re | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 10 | |
| Hans, Schm | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 53 | | 10 | 10 | |
| Michael, S | 60 | 53 | 70 | 80 | 50 | 70 | 50 | 70 | 21.5 | 10 | 10 | 11 | |
| Randy, Smi | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 10 | |
| Don, Jones | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 10 | |
| Justin, Bi | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 12.5 | 9.5 | 10 | 10 | |
| Michael, J | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 10 | |
| Hans, Muir | 60 | 60 | 70 | 80 | 47.5 | 70 | 60 | 80 | 38 | 10 | 10 | 10 | |
| Arnold, Te | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 9.5 | 10 | 10 | 12 | |
| Hillary, C | 60 | 60 | 70 | 80 | 45 | 70 | 60 | 80 | 16.5 | 10 | 10 | 10 | |
| Jannet, Ja | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 55 | 10 | 10 | 10 | |
| Carolle, K | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 44 | 10 | | 10 | |
| Nancy, Pol | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 28 | 10 | 10 | 10 | |
| Elton, Joh | 58 | 51 | 62 | 34 | 35 | 60 | 0 | 50 | 5 | 10 | 10 | 11 | |
| Sylvester, | 53 | 53.5 | 70 | 80 | 50 | 62 | 60 | 80 | 16 | 10 | 10 | 10 | |
| Al, Pacino | 60 | 60 | 70 | 80 | 50 | 70 | 60 | 80 | 39 | 10 | 12 | 10 | |

5. Print groups of dictionaries in a list and extract the value of grade in each dictionary.

Code:

```
print student
grade_list = []
for s in student:
    grade_list.append(s.get('Grade'))
print grade_list
```

Result: (Part)

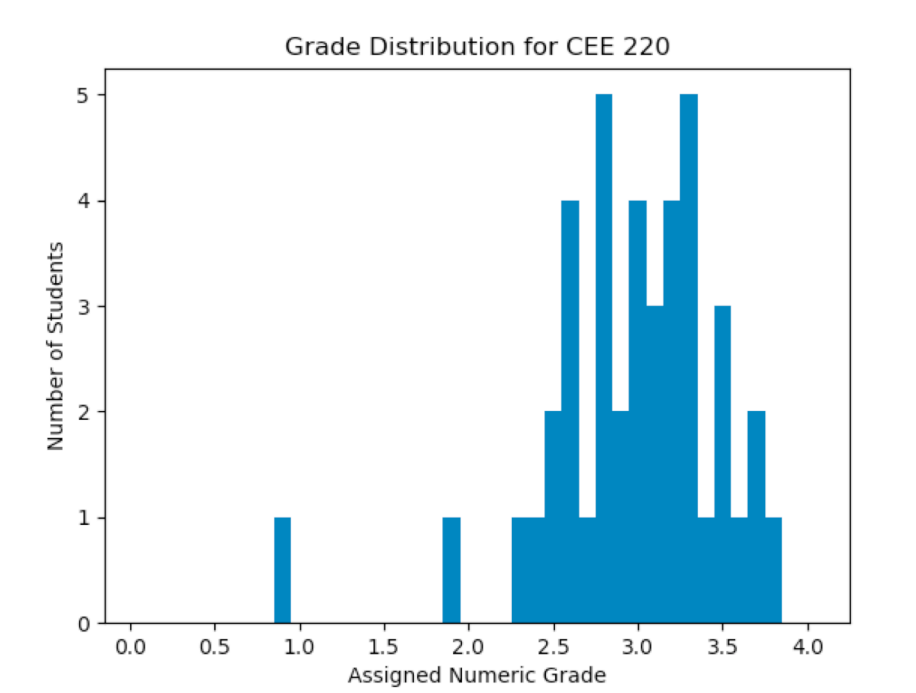

[{'MID_result': 131.0, 'LB_result': 100.0, 'student': '"Sam, Cook"', 'Grade': 2.4, 'FN_result': 33.5, 'AS_result': 582.5, 'Weightedscore': 0.6872641509433963}, {'MID_result': 120.5, 'LB_result': 103.0, 'student': '"Charlie, Brown"', 'Grade': 2.8, 'FN_result': 62.5, 'AS_result': 580.0, 'Weightedscore': 0.7535849056603774}, {'MID_result': 188.5, 'LB_result': 102.0, 'student': '"Peter, Pan"', 'Grade': 3.8, 'FN_result': 89.0, 'AS_result': 543.5, 'Weightedscore': 0.9513679245283019}, {'MID_result': 134.0, 'LB_result': 102.0, 'student': '"Robbie, Williams"', 'Grade': 2.8, 'FN_result': 65.0, 'AS_result': 527.5, 'Weightedscore': 0.7628207547169812}, {'MID_result': 128.5, 'LB_result': 103.0, 'student': '"Sammy, Davis"', 'Grade': 2.5, 'FN_result': 47.0, 'AS_result': 552.0, 'Weightedscore': 0.709877358490566},…{'MID_result': 119.0, 'LB_result': 102.5, 'student': '"Jim, Kirk"', 'Grade': 2.8, 'FN_result': 71.0, 'AS_result': 559.0, 'Weightedscore': 0.7659292452830189}, {'MID_result': 160.25, 'LB_result': 104.5, 'student': 'Spock', 'Grade': 3.3, 'FN_result': 70.0, 'AS_result': 585.0, 'Weightedscore': 0.8586933962264152}, {'MID_result': 143.5, 'LB_result': 86.0, 'student': '"Lt, Ohura"', 'Grade': 3.1, 'FN_result': 83.5, 'AS_result': 521.5, 'Weightedscore': 0.8264905660377357}]

[2.4, 2.8, 3.8, 2.8, 2.5, 2.8, 3.2, 0.9, 3.5, 3.7, 2.6, 3.3, 3.1, 3.4, 2.6, 3.7, 3.2, 3.0, 2.9, 2.3, 3.2, 3.6, 1.9, 3.1, 3.3, 2.6, 2.8, 3.5, 3.5, 3.2, 2.5, 2.6, 3.3, 3.0, 2.9, 3.0, 3.3, 2.7, 3.0, 2.8, 3.3, 3.1]

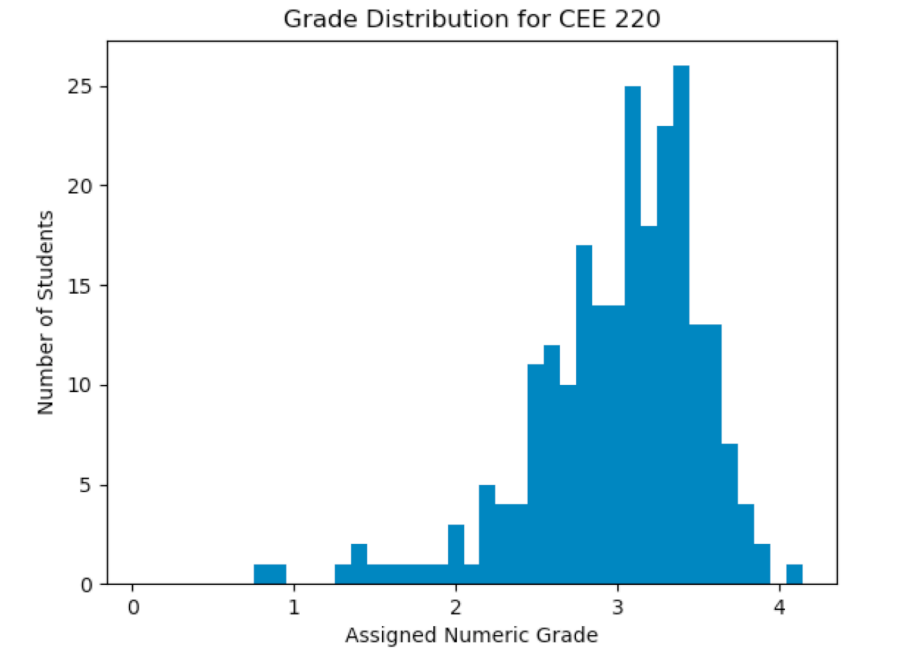

Console:

```
[{'MID_result': 131.0, 'LB_result': 100.0, 'student': '"Sam, Cook"', 'Grade': 2.4, 'FN_result'
[2.4, 2.8, 3.8, 2.8, 2.5, 2.8, 3.2, 0.9, 3.5, 3.7, 2.6, 3.3, 3.1, 3.4, 2.6, 3.7, 3.2, 3.0, 2.9
```

6. The plot for CEE_200_Scores.txt




7. The plot for CEE_200_Alternativelist.txt




**Result:**
Validation shows that all required input options yield correct results.