



## Project Cover Page

Assignment Title:	Project On Grade Sheet Generation		
Assignment No:	01	Date of Submission:	14 August 2021
Course Title:	Programming In Python		
Course Code:	CSC4162	Section:	A
Semester:	Summer	2020-21	Course Teacher: Akinul Islam Jony

**Declaration and Statement of Authorship:**

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them material used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

\* Student(s) must complete all details except the faculty use part.

\*\* Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.:

No	Name	ID	Program	Signature
1	MD. SHIHAB UDDIN AHMED	18-38960-3	BSc [CSE]	
2	MD. NAFISUR RAHMAN	18-39022-3	BSc [CSE]	
3	MARIUM SANA MIM	18-39059-3	BSc [CSE]	
4	A.A.M. FAIYAZ RAHMAN	18-38938-3	BSc [CSE]	
5			Choose an item.	
6			Choose an item.	
7			Choose an item.	
8			Choose an item.	
9			Choose an item.	
10			Choose an item.	

**Faculty use only**

FACULTY COMMENTS	Marks Obtained	
	Total Marks	

# Grade Sheet Generation Using Python

## NumPy, Pandas, Matplotlib & Seaborn

---

### Group Info:

1. Md. Shihab Uddin Ahmed-----18-38960-3
2. Md. Nafisur Rahman-----18-39022-3
3. A.A.M. Faiyaz Rahman-----18-38938-3
4. Marium Sana Mim-----18-39059-3

### Import Required Libraries:

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

### Load Data From Datasets:

1. Extract Quiz Marks
2. Extract Lab Exam Marks
3. Extract Assignment Marks
4. Extract Attendance Marks
5. Store All as Separate NumPy Array

```
In [2]: q1df = pd.read_excel(r'Quizes/Quiz 1.xlsx', sheet_name='Sheet1')
sid1=np.array(len(q1df.index), dtype='str')
sid1=pd.DataFrame(q1df, columns=['Email']).to_numpy()
for i in range(0,len(q1df.index)):
    sid1[i][0]=sid1[i][0].split('@')
    sid1[i]=sid1[i][0][0]
score1=np.array(len(q1df.index), dtype=np.int32)
score1=pd.DataFrame(q1df, columns=['Total points']).to_numpy()
```

```

q2df = pd.read_excel (r'Quizes/Quiz 2.xlsx', sheet_name='Sheet1')
sid2=np.array(len(q2df.index), dtype='str')
sid2=pd.DataFrame(q2df, columns=['Email']).to_numpy()
for i in range(0,len(q2df.index)):
    sid2[i][0]=sid2[i][0].split('@')
    sid2[i]=sid2[i][0][0]
score2=np.array(len(q2df.index), dtype=np.int32)
score2=pd.DataFrame(q2df, columns=['Total points']).to_numpy()

q3df = pd.read_excel (r'Quizes/Quiz 3.xlsx', sheet_name='Sheet1')
sid3=np.array(len(q3df.index), dtype='str')
sid3=pd.DataFrame(q3df, columns=['Email']).to_numpy()
for i in range(0,len(q3df.index)):
    sid3[i][0]=sid3[i][0].split('@')
    sid3[i]=sid3[i][0][0]
score3=np.array(len(q3df.index), dtype=np.int32)
score3=pd.DataFrame(q3df, columns=['Total points']).to_numpy()

ldf = pd.read_excel (r'Lab Exam.xlsx', sheet_name='Sheet1')
sid4=np.array(len(ldf.index), dtype='str')
sid4=pd.DataFrame(ldf, columns=['Email']).to_numpy()
for i in range(0,len(ldf.index)):
    sid4[i][0]=sid4[i][0].split('@')
    sid4[i]=sid4[i][0][0]
score4=np.array(len(ldf.index), dtype=np.int32)
score4=pd.DataFrame(ldf, columns=['Total points']).to_numpy()

adf = pd.read_csv ('Assignment.csv')
sid5=np.array(len(adf.index), dtype='str')
sid5=pd.DataFrame(adf, columns=['Student ID']).to_numpy()
score5=np.array(len(adf.index), dtype=np.int32)
score5=pd.DataFrame(adf, columns=['Ass.']).to_numpy()
name=np.array(len(adf.index), dtype='str')
name=pd.DataFrame(adf, columns=['Name']).to_numpy()
space=" "
sname=" "
for i in range(0,len(adf.index)):
    name[i][0]=name[i][0].split(', ')
    if len(name[i][0])==2:
        sname=np.char.add(name[i][0][1],space)
        name[i]=np.char.add(sname,name[i][0][0])
    else:
        name[i]=name[i][0][0]

```

```

file_data = open('Attendance_files/Week 1 Lab .csv')
name0=""
rows=[100]
i=0
for row in file_data:
    row=row.split('\x00')
    rows=row
    for x in rows:
        name0+=x
    i+=1
name0=name0.split('\t')
i=0
for x in name0:
    name0[i]=x.split('\n')
    if len(name0[i])==2:
        name0[i]=name0[i][1]
    else:
        name0[i]="Empty"
    i+=1

file_data = open('Attendance_files/Week 1 Theory.csv')
name1=""
rows=[100]
i=0
for row in file_data:
    row=row.split('\x00')
    rows=row
    for x in rows:
        name1+=x
    i+=1
name1=name1.split('\t')
i=0
for x in name1:
    name1[i]=x.split('\n')
    if len(name1[i])==2:
        name1[i]=name1[i][1]
    else:
        name1[i]="Empty"
    i+=1

file_data = open('Attendance_files/Week 2 Theory.csv')
name2=""
rows=[100]
i=0
for row in file_data:
    row=row.split('\x00')

```

```

        rows=row
        for x in rows:
            name2+=x
        i+=1
name2=name2.split('\t')
i=0
for x in name2:
    name2[i]=x.split('\n')
    if len(name2[i])==2:
        name2[i]=name2[i][1]
    else:
        name2[i]="Empty"
    i+=1

file_data = open('Attendance_files/Week 4 Lab (Makeup).csv')
name3=""
rows=[100]
i=0
for row in file_data:
    row=row.split('\x00')
    rows=row
    for x in rows:
        name3+=x
    i+=1
name3=name3.split('\t')
i=0
for x in name3:
    name3[i]=x.split('\n')
    if len(name3[i])==2:
        name3[i]=name3[i][1]
    else:
        name3[i]="Empty"
    i+=1

file_data = open('Attendance_files/Week 5 Lab.csv')
name4=""
rows=[100]
i=0
for row in file_data:
    row=row.split('\x00')
    rows=row
    for x in rows:
        name4+=x
    i+=1
name4=name4.split('\t')
i=0

```

```

for x in name4:
    name4[i]=x.split('\n')
    if len(name4[i])==2:
        name4[i]=name4[i][1]
    else:
        name4[i]="Empty"
    i+=1

```

## Processing Loaded Data into a Table of Data:

1. Initializing a 2D-Array
2. Compare ID to find out Quiz marks, Lab Exam & Assignment
3. Compare Name to find out Attendance
4. Calculation of Attendance & Best 2 Quizes
5. Calculation of Total Marks
6. Assigning Grades According to the Total Marks

In [3]:

```

rows, cols = (len(sid5), 16)
result=[]
for i in range(rows):
    col = []
    for j in range(cols):
        col.append(0)
    result.append(col)

for r in range(0,len(sid5)):
    result[r][0]=sid5[r][0]
    result[r][1]=name[r][0]
    result[r][13]=score5[r][0]
    for q in range(0,len(sid1)):
        if(sid1[q][0]==result[r][0]):
            result[r][8]=score1[q][0]
        else:
            continue
    for q in range(0,len(sid2)):
        if(sid2[q][0]==result[r][0]):
            result[r][9]=score2[q][0]
        else:
            continue
    for q in range(0,len(sid3)):
        if(sid3[q][0]==result[r][0]):
            result[r][10]=score3[q][0]

```

```

        else:
            continue
    for q in range(0, len(sid4)):
        if sid4[q][0] == result[r][0]:
            result[r][12] = score4[q][0]
        else:
            continue
    for nam in range(0, len(name0)):
        if name0[nam] == result[r][1]:
            result[r][2] = 1
        else:
            continue
    for nam in range(0, len(name1)):
        if name1[nam] == result[r][1]:
            result[r][3] = 1
        else:
            continue
    for nam in range(0, len(name2)):
        if name2[nam] == result[r][1]:
            result[r][4] = 1
        else:
            continue
    for nam in range(0, len(name3)):
        if name3[nam] == result[r][1]:
            result[r][5] = 1
        else:
            continue
    for nam in range(0, len(name4)):
        if name4[nam] == result[r][1]:
            result[r][6] = 1
        else:
            continue
    for r in range(0, len(sid5)):
        result[r][7] = (result[r][2] + result[r][3] + result[r][4] + result[r][5] + result[r][6]) * 2
        list1 = [result[r][8], result[r][9], result[r][10]]
        list1.sort()
        result[r][11] = list1[-1] + list1[-2]
        result[r][14] = result[r][13] + result[r][12] + result[r][11] + result[r][7]
        if result[r][14] >= 90:
            result[r][15] = 'A+'
        elif result[r][14] >= 85:
            result[r][15] = 'A'
        elif result[r][14] >= 80:
            result[r][15] = 'B+'
        elif result[r][14] >= 75:
            result[r][15] = 'B'

```

```

elif result[r][14]>=70:
    result[r][15]='C+'
elif result[r][14]>=65:
    result[r][15]='C'
elif result[r][14]>=60:
    result[r][15]='D+'
elif result[r][14]>=50:
    result[r][15]='D'
else:
    result[r][15]='F'

```

## Generating Grade Sheet

1. Converting 2D-Array into DataFrame Object
2. Write to An Excel File as Grade Sheet

```

In [4]: df = pd.DataFrame(data=result, columns=["Student ID", "Name", "Week 1 LAB", "Week 1 Theory", "Week 2 Theory",
                                                "Week 4 LAB(Makeup)", "Week 5 LAB", "Attendance Marks", "Quiz1", "Quiz2",
                                                "Quiz3", "Quiz Total", "Lab Exam", "Assignment", "Total Marks", "Grade"])
df.to_excel (r'FinalGradeSheet.xlsx', index = False, header = True)

```

# Grade Data Representation

## Using Seaborn & Matplotlib

---

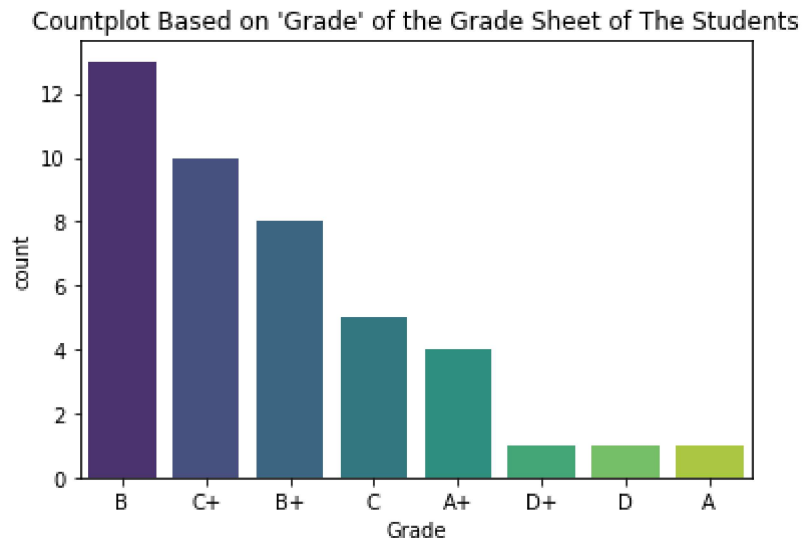
### Counting The Population of Each Grade:

```

In [5]: sns.countplot(x="Grade", data=df, palette="viridis", order = df['Grade'].value_counts().index)
plt.title("Countplot Based on 'Grade' of the Grade Sheet of The Students");

```



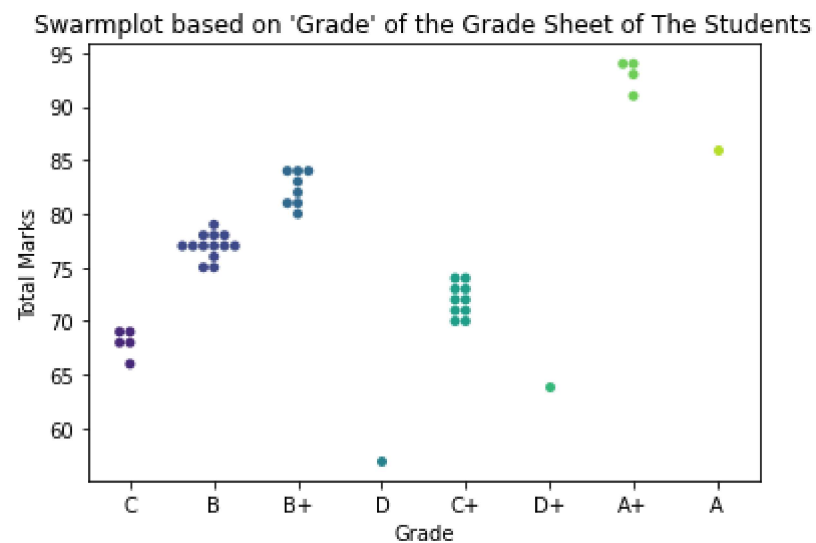


## Showing The Representation of Marks Secured in Each Grade:

(This analysis shows whether the bonus mark can change grades or not)

In [6]:

```
sns.swarmplot(x='Grade', y='Total Marks', data=df, palette="viridis")  
plt.title("Swarmplot based on 'Grade' of the Grade Sheet of The Students");
```



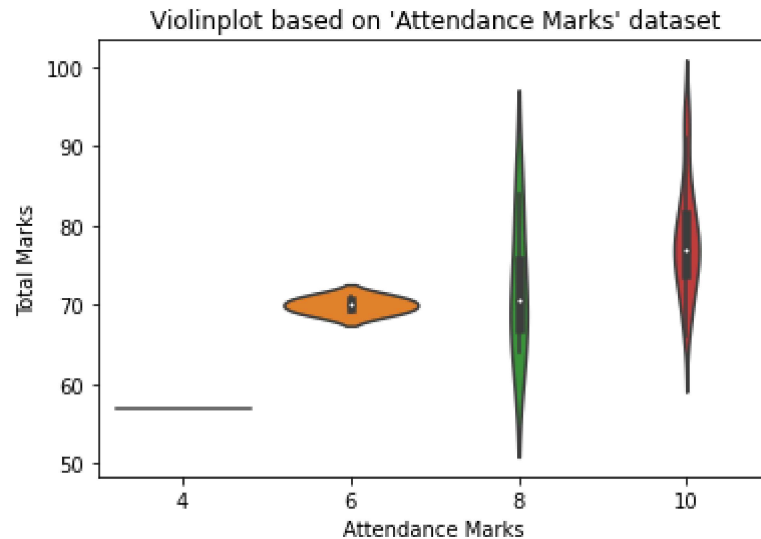
In [ ]:

## Showing Total Marks for Based on Attendance:

(This represents that the density of population in total marks secured based on the attendance.)

In [7]:

```
sns.violinplot(x='Attendance Marks', y='Total Marks', data=df)
plt.title("Violinplot based on 'Attendance Marks' dataset");
```



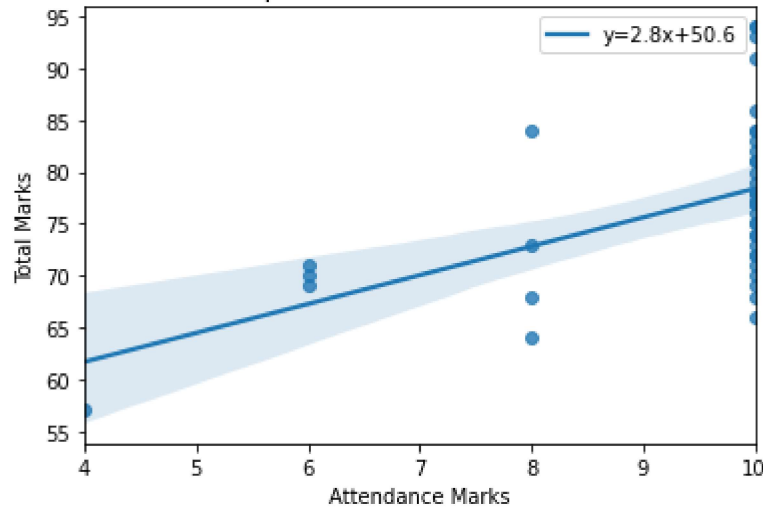
## Linear Regression of Total Marks Based on Attendance:

(This shows how likely students securing good marks based on attendance.)

In [8]:

```
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Attendance Marks'], df['Total Marks'])
sns.regplot(x="Attendance Marks", y="Total Marks", data=df,
            line_kws={'label': "y={0:.1f}x+{1:.1f}".format(slope, intercept)})
plt.legend()
plt.title("Regplot based on 'Total Marks' in Respect of 'Attendance Marks' of the Grade Sheet of The Students");
```

Regplot based on 'Total Marks' in Respect of 'Attendance Marks' of the Grade Sheet of The Students



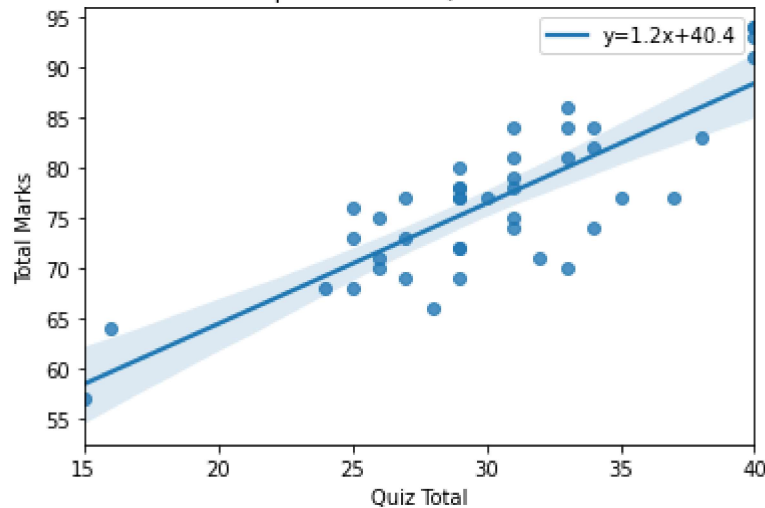
## Linear Regression of Total Marks Based on Quiz Marks:

(This shows how likely students securing good marks based on Quiz Marks.)

In [9]:

```
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Quiz Total'],df['Total Marks'])
sns.regplot(x="Quiz Total", y="Total Marks", data=df, line_kws={'label':"y={0:.1f}x+{1:.1f}".format(slope,intercept)})
plt.legend()
plt.title("Regplot based on 'Total Marks' in Respect of 'Total Quiz Marks' of the Grade Sheet of The Students");
```

Regplot based on 'Total Marks' in Respect of 'Total Quiz Marks' of the Grade Sheet of The Students

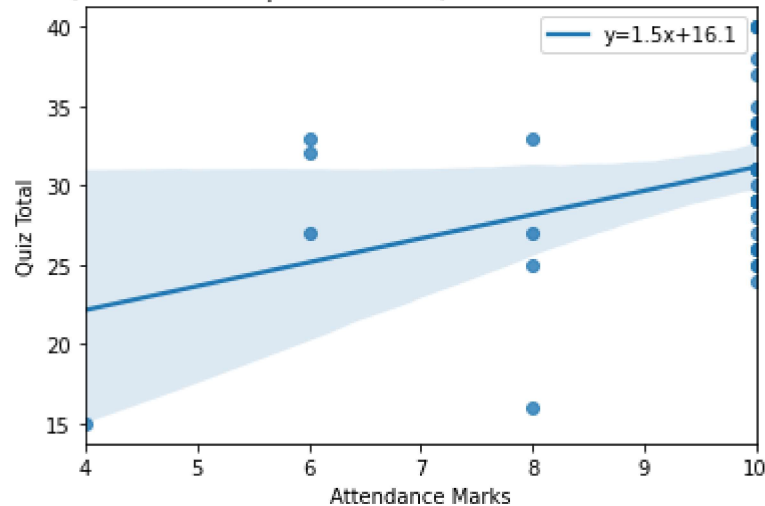


## Linear Regression of Quiz Marks Based on Attendance:

(This shows how likely students securing good marks in Quiz based on attendance.)

```
In [10]: slope, intercept, r_value, p_value, std_err = stats.linregress(df['Attendance Marks'],df['Quiz Total'])
sns.regplot(x="Attendance Marks", y="Quiz Total", data=df,
            line_kws={'label':"y={0:.1f}x+{1:.1f}".format(slope,intercept)})
plt.legend()
plt.title("Regplot based on 'Quiz Total' in Respect of 'Total Quiz Marks' of the Grade Sheet of The Students");
```

Regplot based on 'Quiz Total' in Respect of 'Total Quiz Marks' of the Grade Sheet of The Students



In [ ]: