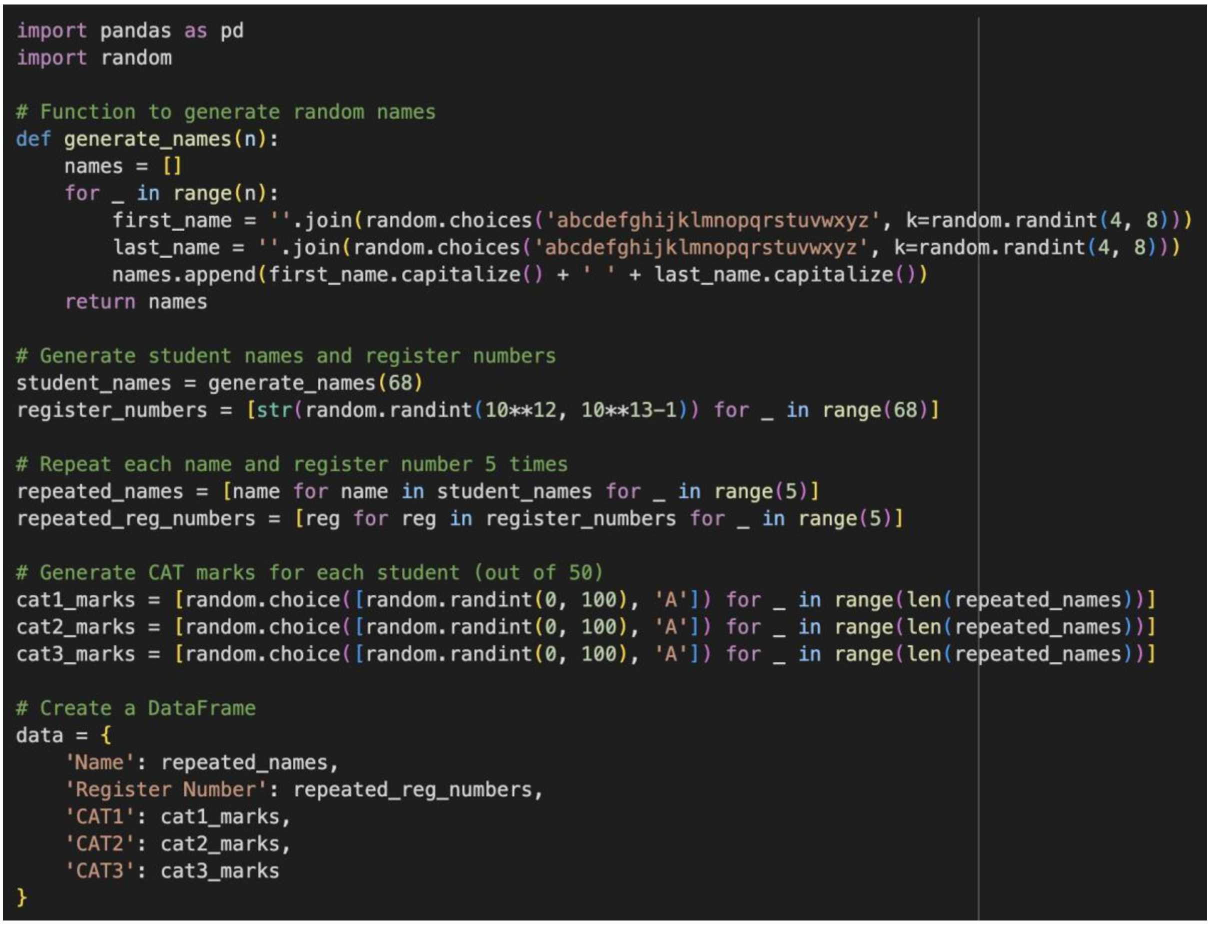
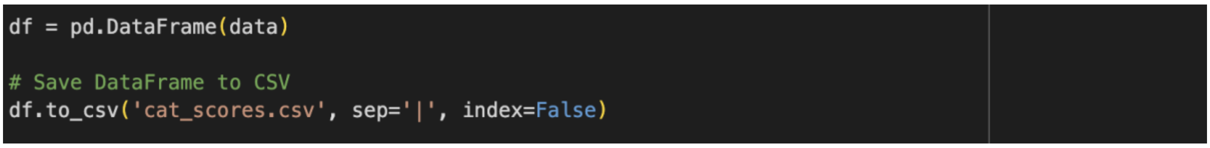
Name: SK Shivaanee
Register number: 3122235001123
Section: CSE B
Academic year: **2023-2024**

Assignment 1
UCS2202 – Foundations of Data Science

Creating the dataframe with the student's name, register ID, and the three cat scores.

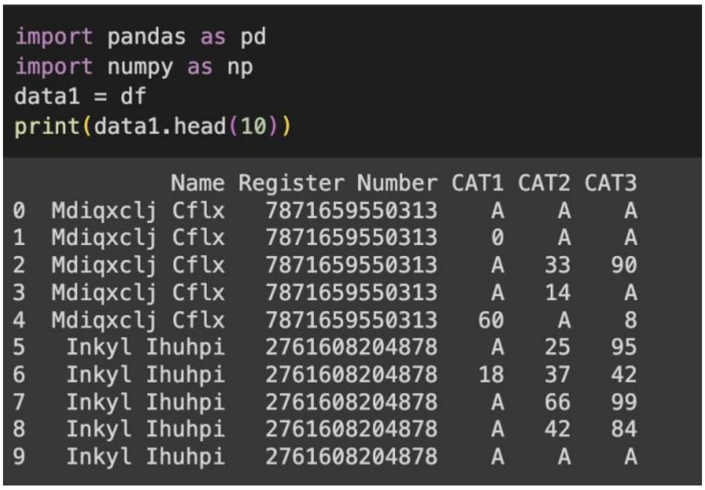


```
import pandas as pd
import random

# Function to generate random names
def generate_names(n):
    names = []
    for _ in range(n):
        first_name = ''.join(random.choices('abcdefghijklmnopqrstuvwxyz', k=random.randint(4, 8)))
        last_name = ''.join(random.choices('abcdefghijklmnopqrstuvwxyz', k=random.randint(4, 8)))
        names.append(first_name.capitalize() + ' ' + last_name.capitalize())
    return names

# Generate student names and register numbers
student_names = generate_names(68)
register_numbers = [str(random.randint(10**12, 10**13-1)) for _ in range(68)]

# Repeat each name and register number 5 times
repeated_names = [name for name in student_names for _ in range(5)]
repeated_reg_numbers = [reg for reg in register_numbers for _ in range(5)]

# Generate CAT marks for each student (out of 50)
cat1_marks = [random.choice([random.randint(0, 100), 'A']) for _ in range(len(repeated_names))]
cat2_marks = [random.choice([random.randint(0, 100), 'A']) for _ in range(len(repeated_names))]
cat3_marks = [random.choice([random.randint(0, 100), 'A']) for _ in range(len(repeated_names))]

# Create a DataFrame
data = {
    'Name': repeated_names,
    'Register Number': repeated_reg_numbers,
    'CAT1': cat1_marks,
    'CAT2': cat2_marks,
    'CAT3': cat3_marks
}
```



```
df = pd.DataFrame(data)

# Save DataFrame to CSV
df.to_csv('cat_scores.csv', sep='|', index=False)
```

The data set would look like:

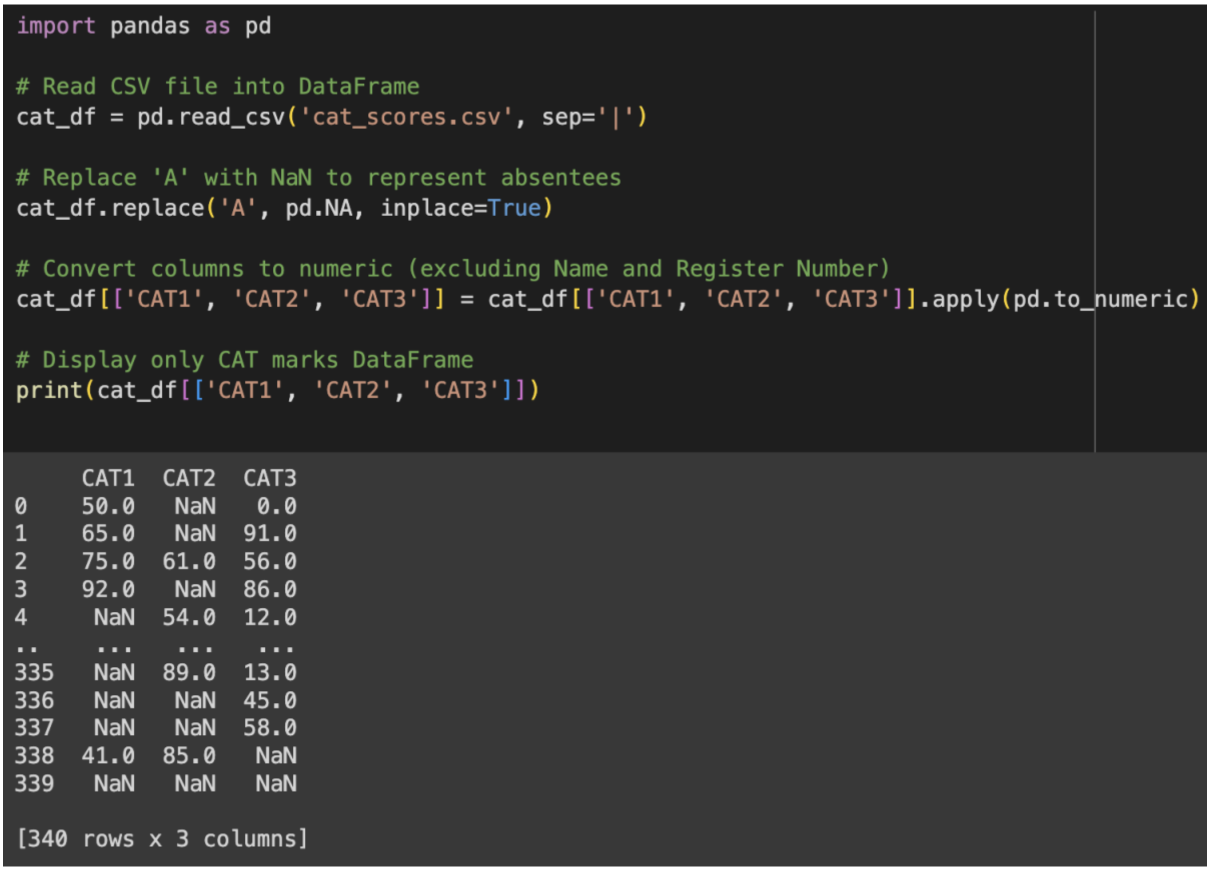


```python
import pandas as pd
import numpy as np
data1 = df
print(data1.head(10))
```

```
             Name Register Number CAT1 CAT2 CAT3
0  Mdiqxclj Cflx   7871659550313    A    A    A
1  Mdiqxclj Cflx   7871659550313    0    A    A
2  Mdiqxclj Cflx   7871659550313    A   33   90
3  Mdiqxclj Cflx   7871659550313    A   14    A
4  Mdiqxclj Cflx   7871659550313   60    A    8
5   Inkyl Ihuhpi   2761608204878    A   25   95
6   Inkyl Ihuhpi   2761608204878   18   37   42
7   Inkyl Ihuhpi   2761608204878    A   66   99
8   Inkyl Ihuhpi   2761608204878    A   42   84
9   Inkyl Ihuhpi   2761608204878    A    A    A
```

CREATE DATAFRAMES:

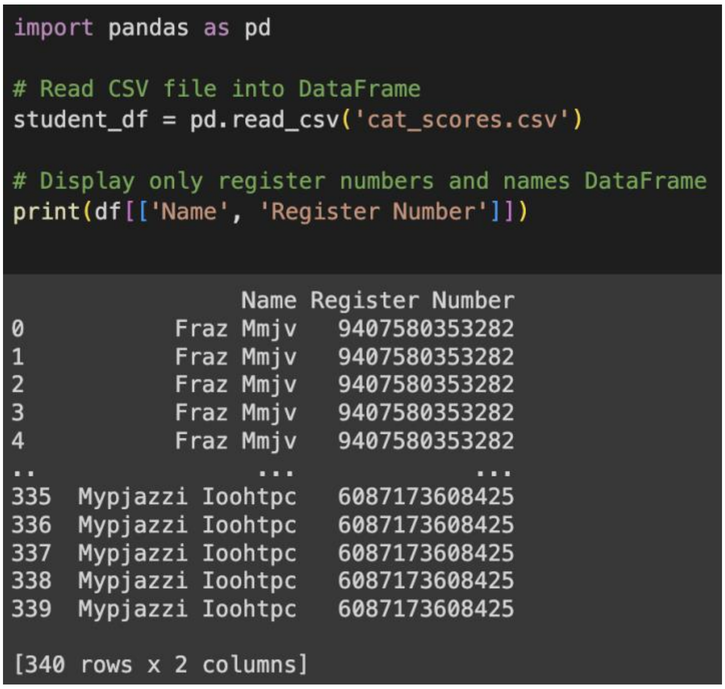
1(1) Create a dataframe containing the three CAT marks of the students from the CSV file. The CSV file has three columns: CAT1, CAT2 and CAT3. There may be a few absentees in the CATs. The instructor indicates an absentee's mark in a CAT as "A" in the CSV file.



```python
import pandas as pd

# Read CSV file into DataFrame
cat_df = pd.read_csv('cat_scores.csv', sep='|')

# Replace 'A' with NaN to represent absentees
cat_df.replace('A', pd.NA, inplace=True)

# Convert columns to numeric (excluding Name and Register Number)
cat_df[['CAT1', 'CAT2', 'CAT3']] = cat_df[['CAT1', 'CAT2', 'CAT3']].apply(pd.to_numeric)

# Display only CAT marks DataFrame
print(cat_df[['CAT1', 'CAT2', 'CAT3']])
```

```
     CAT1  CAT2  CAT3
0    50.0   NaN   0.0
1    65.0   NaN  91.0
2    75.0  61.0  56.0
3    92.0   NaN  86.0
4     NaN  54.0  12.0
..    ...   ...   ...
335   NaN  89.0  13.0
336   NaN   NaN  45.0
337   NaN   NaN  58.0
338  41.0  85.0   NaN
339   NaN   NaN   NaN

[340 rows x 3 columns]
```

1(2) Create another dataframe containing the register numbers and the names of the students. The CSV file has two columns: Reg No and Name.

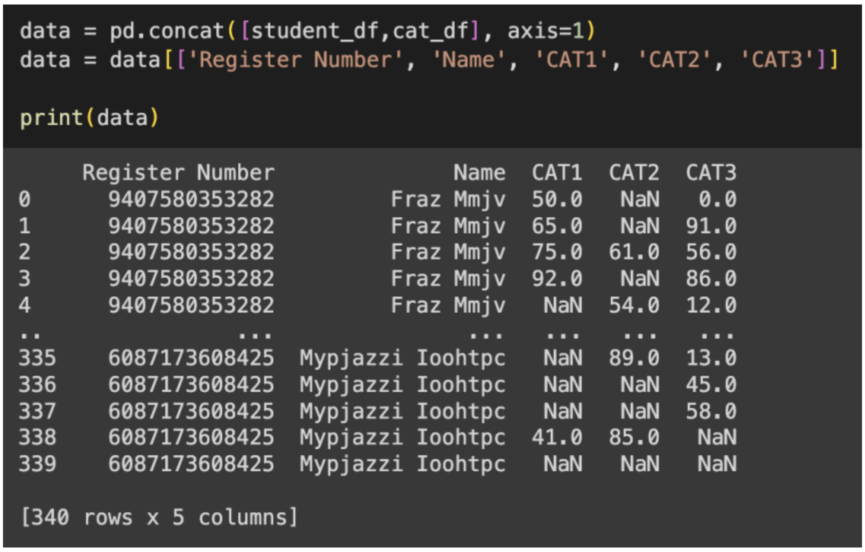


```python
import pandas as pd

# Read CSV file into DataFrame
student_df = pd.read_csv('cat_scores.csv')

# Display only register numbers and names DataFrame
print(df[['Name', 'Register Number']])
```

```
                 Name Register Number
0           Fraz Mmjv   9407580353282
1           Fraz Mmjv   9407580353282
2           Fraz Mmjv   9407580353282
3           Fraz Mmjv   9407580353282
4           Fraz Mmjv   9407580353282
..                ...             ...
335  Mypjazzi Ioohtpc   6087173608425
336  Mypjazzi Ioohtpc   6087173608425
337  Mypjazzi Ioohtpc   6087173608425
338  Mypjazzi Ioohtpc   6087173608425
339  Mypjazzi Ioohtpc   6087173608425

[340 rows x 2 columns]
```

COMBINE DATAFRAMES:

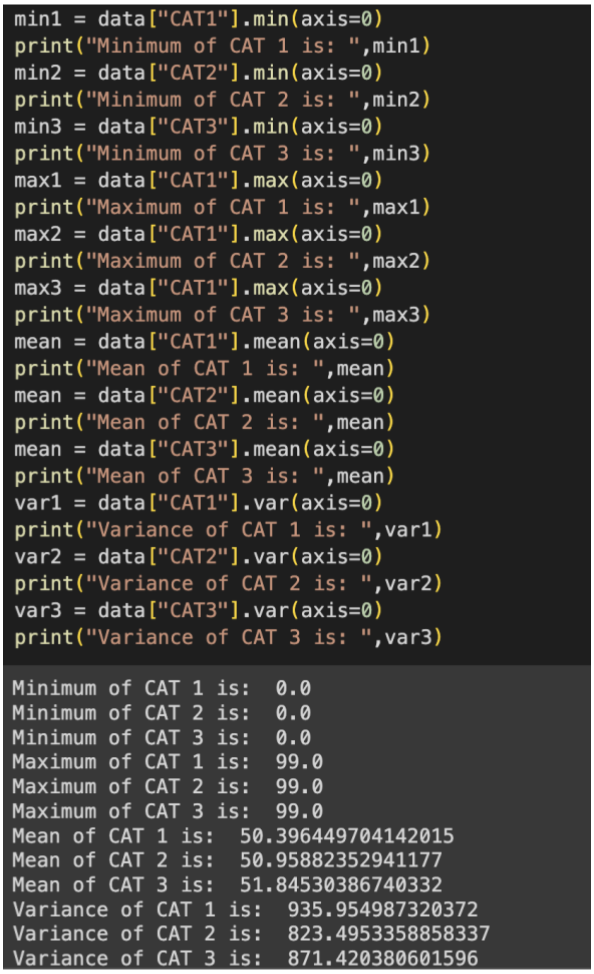
2(1) Combine the two DataFrames to create a new DataFrame containing Reg No, Name, CAT1, CAT2 and CAT3 columns.



```
data = pd.concat([student_df,cat_df], axis=1)
data = data[['Register Number', 'Name', 'CAT1', 'CAT2', 'CAT3']]

print(data)
```

```
     Register Number              Name  CAT1  CAT2  CAT3
0      9407580353282         Fraz Mmjv  50.0   NaN   0.0
1      9407580353282         Fraz Mmjv  65.0   NaN  91.0
2      9407580353282         Fraz Mmjv  75.0  61.0  56.0
3      9407580353282         Fraz Mmjv  92.0   NaN  86.0
4      9407580353282         Fraz Mmjv   NaN  54.0  12.0
..               ...               ...   ...   ...   ...
335    6087173608425  Mypjazzi Ioohtpc   NaN  89.0  13.0
336    6087173608425  Mypjazzi Ioohtpc   NaN   NaN  45.0
337    6087173608425  Mypjazzi Ioohtpc   NaN   NaN  58.0
338    6087173608425  Mypjazzi Ioohtpc  41.0  85.0   NaN
339    6087173608425  Mypjazzi Ioohtpc   NaN   NaN   NaN

[340 rows x 5 columns]
```

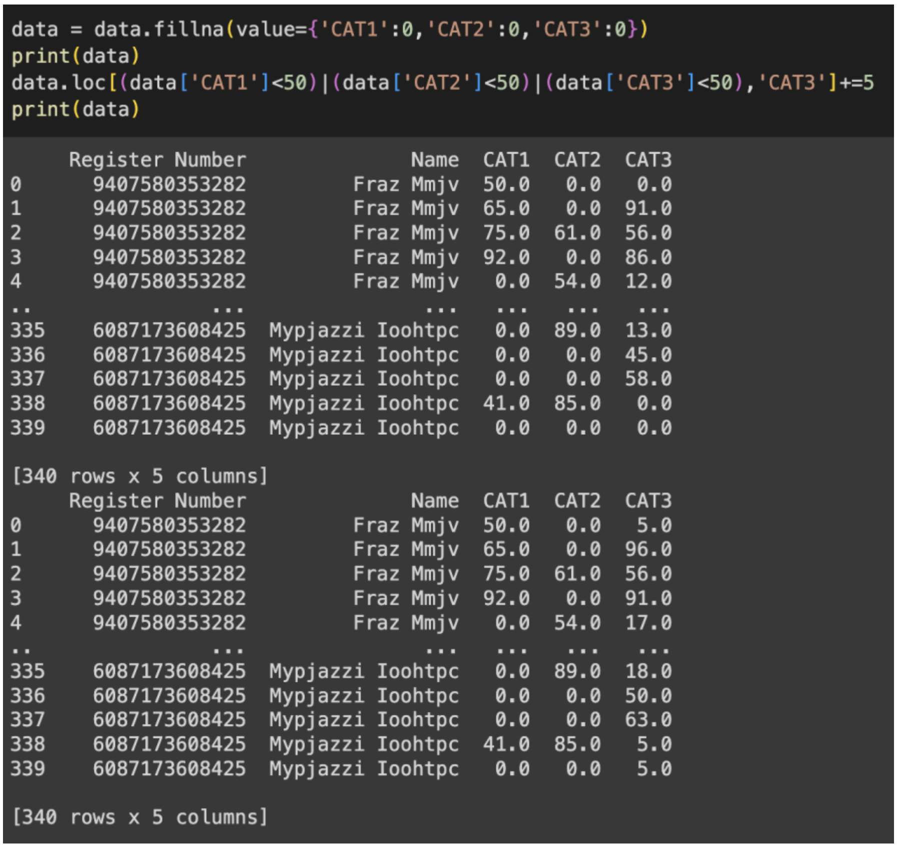
2(2) Print statistics such as the minimum, maximum, mean, and variance for the three CATs.



```
min1 = data["CAT1"].min(axis=0)
print("Minimum of CAT 1 is: ",min1)
min2 = data["CAT2"].min(axis=0)
print("Minimum of CAT 2 is: ",min2)
min3 = data["CAT3"].min(axis=0)
print("Minimum of CAT 3 is: ",min3)
max1 = data["CAT1"].max(axis=0)
print("Maximum of CAT 1 is: ",max1)
max2 = data["CAT1"].max(axis=0)
print("Maximum of CAT 2 is: ",max2)
max3 = data["CAT1"].max(axis=0)
print("Maximum of CAT 3 is: ",max3)
mean = data["CAT1"].mean(axis=0)
print("Mean of CAT 1 is: ",mean)
mean = data["CAT2"].mean(axis=0)
print("Mean of CAT 2 is: ",mean)
mean = data["CAT3"].mean(axis=0)
print("Mean of CAT 3 is: ",mean)
var1 = data["CAT1"].var(axis=0)
print("Variance of CAT 1 is: ",var1)
var2 = data["CAT2"].var(axis=0)
print("Variance of CAT 2 is: ",var2)
var3 = data["CAT3"].var(axis=0)
print("Variance of CAT 3 is: ",var3)
```

```
Minimum of CAT 1 is:  0.0
Minimum of CAT 2 is:  0.0
Minimum of CAT 3 is:  0.0
Maximum of CAT 1 is:  99.0
Maximum of CAT 2 is:  99.0
Maximum of CAT 3 is:  99.0
Mean of CAT 1 is:  50.396449704142015
Mean of CAT 2 is:  50.95882352941177
Mean of CAT 3 is:  51.84530386740332
Variance of CAT 1 is:  935.954987320372
Variance of CAT 2 is:  823.4953358858337
Variance of CAT 3 is:  871.420380601596
```

2(3) The instructor decides to add a grace mark of 5 to those who have scored less than 50 in CAT. Update the column CAT3 with the grace mark added.

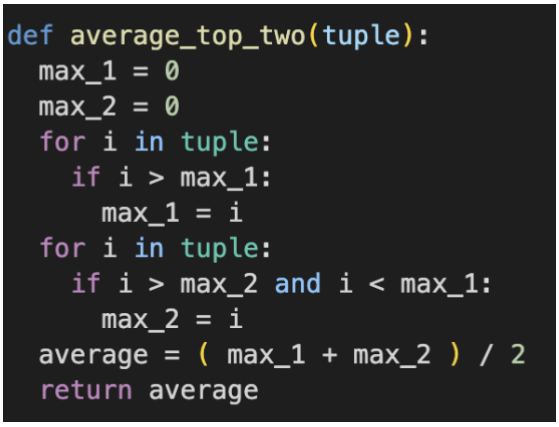


```
data = data.fillna(value={'CAT1':0,'CAT2':0,'CAT3':0})
print(data)
data.loc[(data['CAT1']<50)|(data['CAT2']<50)|(data['CAT3']<50),'CAT3']+=5
print(data)
```

```
     Register Number              Name  CAT1  CAT2  CAT3
0      9407580353282         Fraz Mmjv  50.0   0.0   0.0
1      9407580353282         Fraz Mmjv  65.0   0.0  91.0
2      9407580353282         Fraz Mmjv  75.0  61.0  56.0
3      9407580353282         Fraz Mmjv  92.0   0.0  86.0
4      9407580353282         Fraz Mmjv   0.0  54.0  12.0
..               ...               ...   ...   ...   ...
335    6087173608425  Mypjazzi Ioohtpc   0.0  89.0  13.0
336    6087173608425  Mypjazzi Ioohtpc   0.0   0.0  45.0
337    6087173608425  Mypjazzi Ioohtpc   0.0   0.0  58.0
338    6087173608425  Mypjazzi Ioohtpc  41.0  85.0   0.0
339    6087173608425  Mypjazzi Ioohtpc   0.0   0.0   0.0

[340 rows x 5 columns]
     Register Number              Name  CAT1  CAT2  CAT3
0      9407580353282         Fraz Mmjv  50.0   0.0   5.0
1      9407580353282         Fraz Mmjv  65.0   0.0  96.0
2      9407580353282         Fraz Mmjv  75.0  61.0  56.0
3      9407580353282         Fraz Mmjv  92.0   0.0  91.0
4      9407580353282         Fraz Mmjv   0.0  54.0  17.0
..               ...               ...   ...   ...   ...
335    6087173608425  Mypjazzi Ioohtpc   0.0  89.0  18.0
336    6087173608425  Mypjazzi Ioohtpc   0.0   0.0  50.0
337    6087173608425  Mypjazzi Ioohtpc   0.0   0.0  63.0
338    6087173608425  Mypjazzi Ioohtpc  41.0  85.0   5.0
339    6087173608425  Mypjazzi Ioohtpc   0.0   0.0   5.0

[340 rows x 5 columns]
```

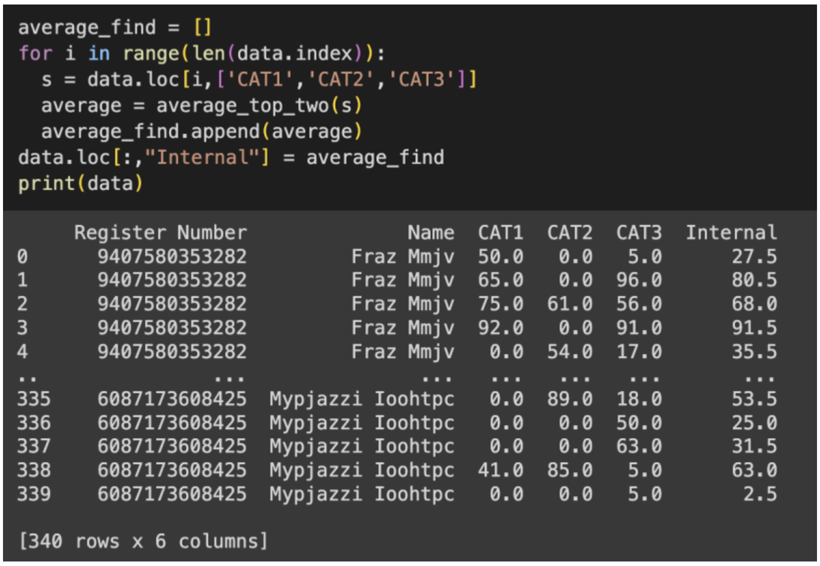
CALCULATE THE INTERNAL MARKS:

3(1) Write a function average_top_two() to find the average of the maximum and the second maximum of a list of numbers. This function takes a 3-tuple of marks.



```
def average_top_two(tuple):
  max_1 = 0
  max_2 = 0
  for i in tuple:
    if i > max_1:
      max_1 = i
  for i in tuple:
    if i > max_2 and i < max_1:
      max_2 = i
  average = ( max_1 + max_2 ) / 2
  return average
```

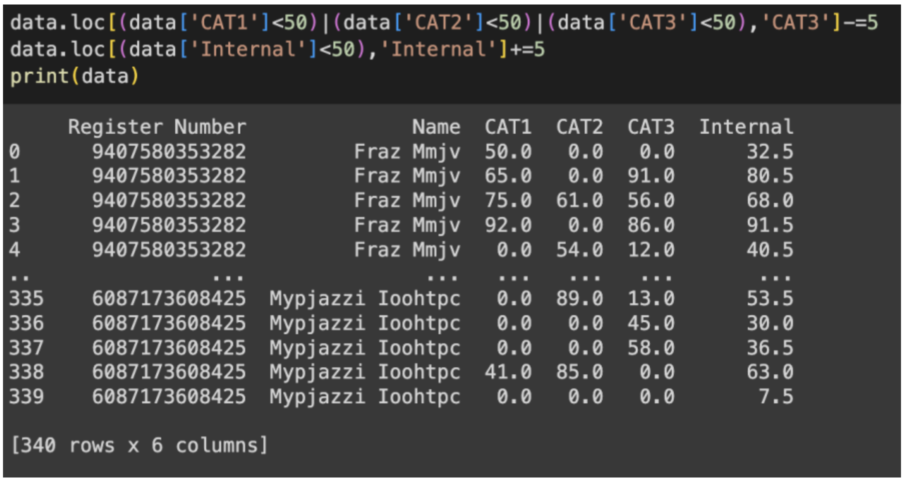
3(2) Apply the function average_top_two() on columns CAT1, CAT2, and CAT3. Create a new column Internal which equals the average of the best two of the three CAT marks for each student.



```
average_find = []
for i in range(len(data.index)):
  s = data.loc[i,['CAT1','CAT2','CAT3']]
  average = average_top_two(s)
  average_find.append(average)
data.loc[:,"Internal"] = average_find
print(data)
```

```
     Register Number             Name  CAT1  CAT2  CAT3  Internal
0      9407580353282        Fraz Mmjv  50.0   0.0   5.0      27.5
1      9407580353282        Fraz Mmjv  65.0   0.0  96.0      80.5
2      9407580353282        Fraz Mmjv  75.0  61.0  56.0      68.0
3      9407580353282        Fraz Mmjv  92.0   0.0  91.0      91.5
4      9407580353282        Fraz Mmjv   0.0  54.0  17.0      35.5
..               ...              ...   ...   ...   ...       ...
335    6087173608425  Mypjazzi Ioohtpc   0.0  89.0  18.0      53.5
336    6087173608425  Mypjazzi Ioohtpc   0.0   0.0  50.0      25.0
337    6087173608425  Mypjazzi Ioohtpc   0.0   0.0  63.0      31.5
338    6087173608425  Mypjazzi Ioohtpc  41.0  85.0   5.0      63.0
339    6087173608425  Mypjazzi Ioohtpc   0.0   0.0   5.0       2.5

[340 rows x 6 columns]
```

3(3) The instructor changes her mind about having added grace marks. She wants to undo the addition of grace mark in the earlier step. Instead, she now wants to add a grace mark only to those whose internal mark is below 50.



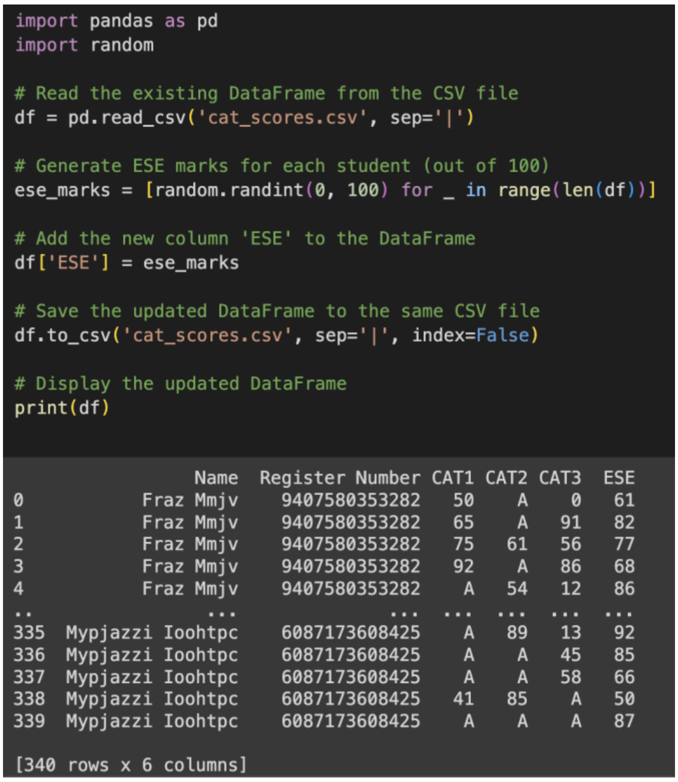
```
data.loc[(data['CAT1']<50)|(data['CAT2']<50)|(data['CAT3']<50),'CAT3']-=5
data.loc[(data['Internal']<50),'Internal']+=5
print(data)
```

```
     Register Number              Name  CAT1  CAT2  CAT3  Internal
0      9407580353282         Fraz Mmjv  50.0   0.0   0.0      32.5
1      9407580353282         Fraz Mmjv  65.0   0.0  91.0      80.5
2      9407580353282         Fraz Mmjv  75.0  61.0  56.0      68.0
3      9407580353282         Fraz Mmjv  92.0   0.0  86.0      91.5
4      9407580353282         Fraz Mmjv   0.0  54.0  12.0      40.5
..               ...               ...   ...   ...   ...       ...
335    6087173608425  Mypjazzi Ioohtpc   0.0  89.0  13.0      53.5
336    6087173608425  Mypjazzi Ioohtpc   0.0   0.0  45.0      30.0
337    6087173608425  Mypjazzi Ioohtpc   0.0   0.0  58.0      36.5
338    6087173608425  Mypjazzi Ioohtpc  41.0  85.0   0.0      63.0
339    6087173608425  Mypjazzi Ioohtpc   0.0   0.0   0.0       7.5

[340 rows x 6 columns]
```

TOTAL MARKS:

The exam office adds the internal mark of each student to her end-semester exam (ESE) mark as the final mark.

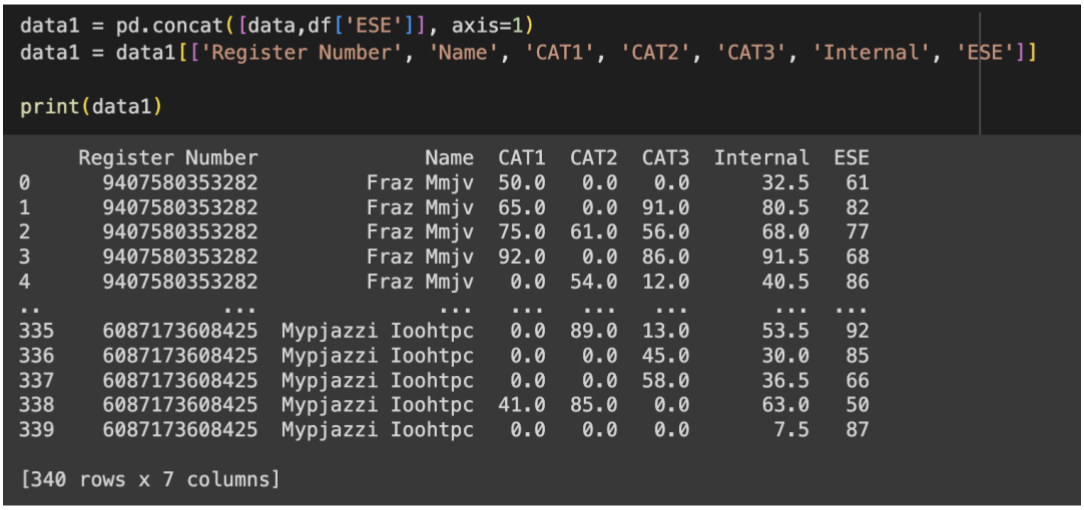
4(1) Add one more column for the end-semester exam marks in your CSV file.



```
import pandas as pd
import random

# Read the existing DataFrame from the CSV file
df = pd.read_csv('cat_scores.csv', sep='|')

# Generate ESE marks for each student (out of 100)
ese_marks = [random.randint(0, 100) for _ in range(len(df))]

# Add the new column 'ESE' to the DataFrame
df['ESE'] = ese_marks

# Save the updated DataFrame to the same CSV file
df.to_csv('cat_scores.csv', sep='|', index=False)

# Display the updated DataFrame
print(df)
```

```
                 Name  Register Number CAT1 CAT2 CAT3  ESE
0           Fraz Mmjv    9407580353282   50    A    0   61
1           Fraz Mmjv    9407580353282   65    A   91   82
2           Fraz Mmjv    9407580353282   75   61   56   77
3           Fraz Mmjv    9407580353282   92    A   86   68
4           Fraz Mmjv    9407580353282    A   54   12   86
..                ...              ...  ...  ...  ...  ...
335  Mypjazzi Ioohtpc    6087173608425    A   89   13   92
336  Mypjazzi Ioohtpc    6087173608425    A    A   45   85
337  Mypjazzi Ioohtpc    6087173608425    A    A   58   66
338  Mypjazzi Ioohtpc    6087173608425   41   85    A   50
339  Mypjazzi Ioohtpc    6087173608425    A    A    A   87

[340 rows x 6 columns]
```

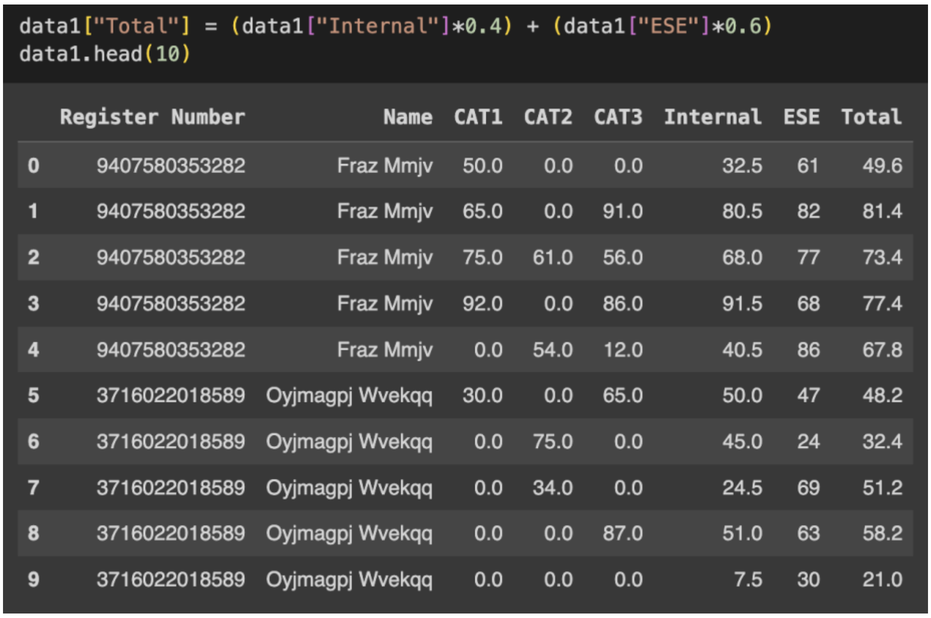
4(2) Do the necessary changes in your program to create the DataFrame accordingly.



```
data1 = pd.concat([data,df['ESE']], axis=1)
data1 = data1[['Register Number', 'Name', 'CAT1', 'CAT2', 'CAT3', 'Internal', 'ESE']]

print(data1)
```

```
     Register Number              Name  CAT1  CAT2  CAT3  Internal  ESE
0      9407580353282         Fraz Mmjv  50.0   0.0   0.0      32.5   61
1      9407580353282         Fraz Mmjv  65.0   0.0  91.0      80.5   82
2      9407580353282         Fraz Mmjv  75.0  61.0  56.0      68.0   77
3      9407580353282         Fraz Mmjv  92.0   0.0  86.0      91.5   68
4      9407580353282         Fraz Mmjv   0.0  54.0  12.0      40.5   86
..               ...               ...   ...   ...   ...       ...  ...
335    6087173608425  Mypjazzi Ioohtpc   0.0  89.0  13.0      53.5   92
336    6087173608425  Mypjazzi Ioohtpc   0.0   0.0  45.0      30.0   85
337    6087173608425  Mypjazzi Ioohtpc   0.0   0.0  58.0      36.5   66
338    6087173608425  Mypjazzi Ioohtpc  41.0  85.0   0.0      63.0   50
339    6087173608425  Mypjazzi Ioohtpc   0.0   0.0   0.0       7.5   87

[340 rows x 7 columns]
```

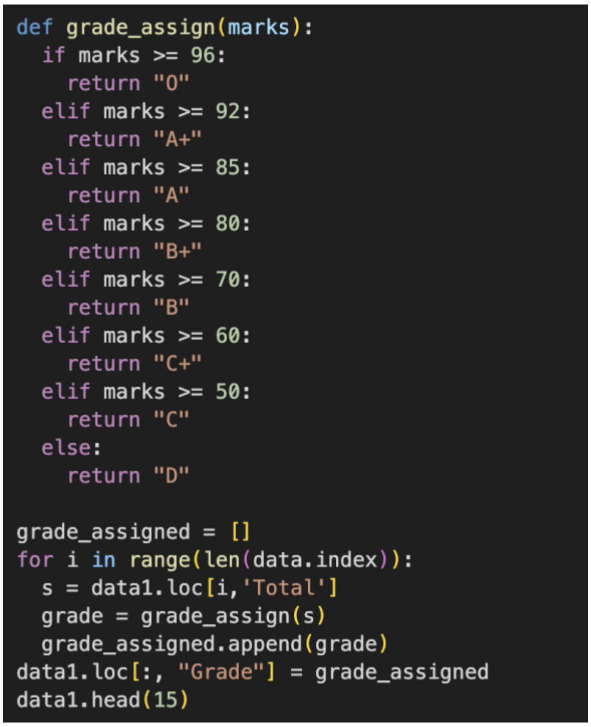
4(3) Add the average of the best of the two CAT marks and the ESE mark to get the final mark. Add a new column to the DataFrame, Total, with the final mark.



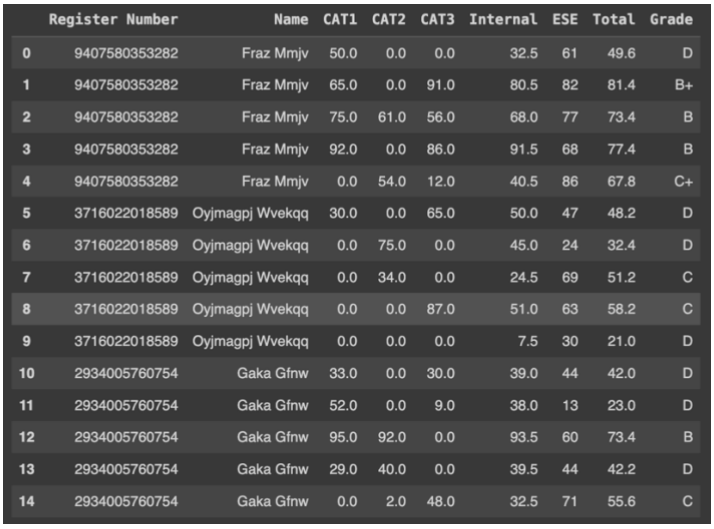
```
data1["Total"] = (data1["Internal"]*0.4) + (data1["ESE"]*0.6)
data1.head(10)
```

	Register Number	Name	CAT1	CAT2	CAT3	Internal	ESE	Total
0	9407580353282	Fraz Mmjv	50.0	0.0	0.0	32.5	61	49.6
1	9407580353282	Fraz Mmjv	65.0	0.0	91.0	80.5	82	81.4
2	9407580353282	Fraz Mmjv	75.0	61.0	56.0	68.0	77	73.4
3	9407580353282	Fraz Mmjv	92.0	0.0	86.0	91.5	68	77.4
4	9407580353282	Fraz Mmjv	0.0	54.0	12.0	40.5	86	67.8
5	3716022018589	Oyjmagpj Wvekqq	30.0	0.0	65.0	50.0	47	48.2
6	3716022018589	Oyjmagpj Wvekqq	0.0	75.0	0.0	45.0	24	32.4
7	3716022018589	Oyjmagpj Wvekqq	0.0	34.0	0.0	24.5	69	51.2
8	3716022018589	Oyjmagpj Wvekqq	0.0	0.0	87.0	51.0	63	58.2
9	3716022018589	Oyjmagpj Wvekqq	0.0	0.0	0.0	7.5	30	21.0

4(4) Add yet another column to the DataFrame with the grade of the student. Grade is a string.



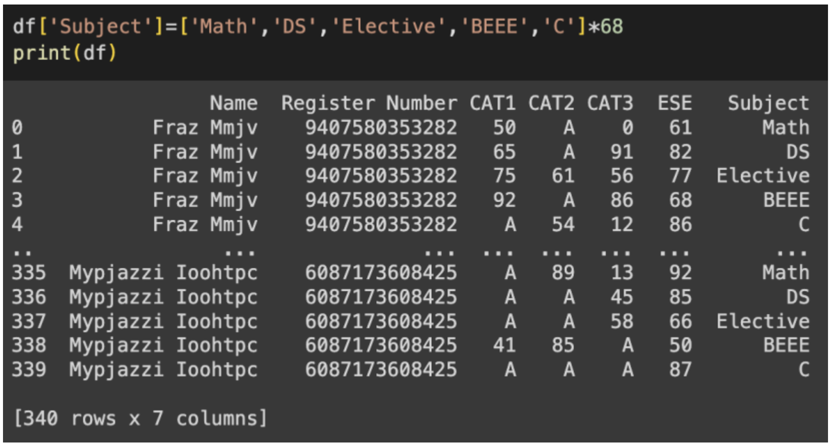
```
def grade_assign(marks):
  if marks >= 96:
    return "O"
  elif marks >= 92:
    return "A+"
  elif marks >= 85:
    return "A"
  elif marks >= 80:
    return "B+"
  elif marks >= 70:
    return "B"
  elif marks >= 60:
    return "C+"
  elif marks >= 50:
    return "C"
  else:
    return "D"

grade_assigned = []
for i in range(len(data.index)):
  s = data1.loc[i,'Total']
  grade = grade_assign(s)
  grade_assigned.append(grade)
data1.loc[:, "Grade"] = grade_assigned
data1.head(15)
```



	Register Number	Name	CAT1	CAT2	CAT3	Internal	ESE	Total	Grade
0	9407580353282	Fraz Mmjv	50.0	0.0	0.0	32.5	61	49.6	D
1	9407580353282	Fraz Mmjv	65.0	0.0	91.0	80.5	82	81.4	B+
2	9407580353282	Fraz Mmjv	75.0	61.0	56.0	68.0	77	73.4	B
3	9407580353282	Fraz Mmjv	92.0	0.0	86.0	91.5	68	77.4	B
4	9407580353282	Fraz Mmjv	0.0	54.0	12.0	40.5	86	67.8	C+
5	3716022018589	Oyjmagpj Wvekqq	30.0	0.0	65.0	50.0	47	48.2	D
6	3716022018589	Oyjmagpj Wvekqq	0.0	75.0	0.0	45.0	24	32.4	D
7	3716022018589	Oyjmagpj Wvekqq	0.0	34.0	0.0	24.5	69	51.2	C
8	3716022018589	Oyjmagpj Wvekqq	0.0	0.0	87.0	51.0	63	58.2	C
9	3716022018589	Oyjmagpj Wvekqq	0.0	0.0	0.0	7.5	30	21.0	D
10	2934005760754	Gaka Gfnw	33.0	0.0	30.0	39.0	44	42.0	D
11	2934005760754	Gaka Gfnw	52.0	0.0	9.0	38.0	13	23.0	D
12	2934005760754	Gaka Gfnw	95.0	92.0	0.0	93.5	60	73.4	B
13	2934005760754	Gaka Gfnw	29.0	40.0	0.0	39.5	44	42.2	D
14	2934005760754	Gaka Gfnw	0.0	2.0	48.0	32.5	71	55.6	C

CONSOLIDATED MARKS OF A CLASS:

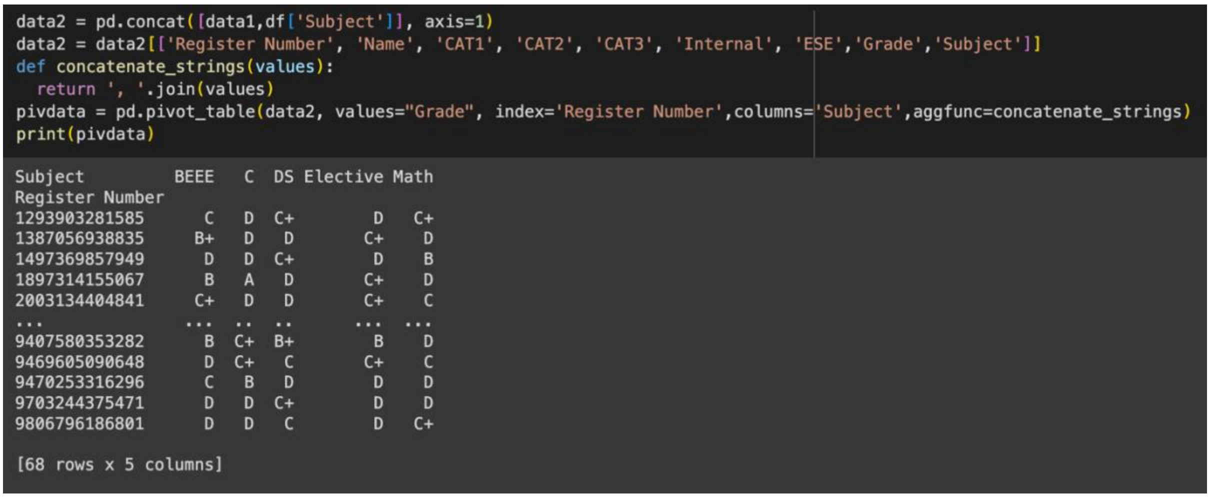
The exam office plan to generate the consolidated marks of a branch.
5(1) Add one more column for the subject names in your CSV file.



```
df['Subject']=['Math','DS','Elective','BEEE','C']*68
print(df)
```

```
                 Name  Register Number CAT1 CAT2 CAT3  ESE   Subject
0           Fraz Mmjv    9407580353282   50    A    0   61      Math
1           Fraz Mmjv    9407580353282   65    A   91   82        DS
2           Fraz Mmjv    9407580353282   75   61   56   77  Elective
3           Fraz Mmjv    9407580353282   92    A   86   68      BEEE
4           Fraz Mmjv    9407580353282    A   54   12   86         C
..                ...              ...  ...  ...  ...  ...       ...
335  Mypjazzi Ioohtpc    6087173608425    A   89   13   92      Math
336  Mypjazzi Ioohtpc    6087173608425    A    A   45   85        DS
337  Mypjazzi Ioohtpc    6087173608425    A    A   58   66  Elective
338  Mypjazzi Ioohtpc    6087173608425   41   85    A   50      BEEE
339  Mypjazzi Ioohtpc    6087173608425    A    A    A   87         C

[340 rows x 7 columns]
```

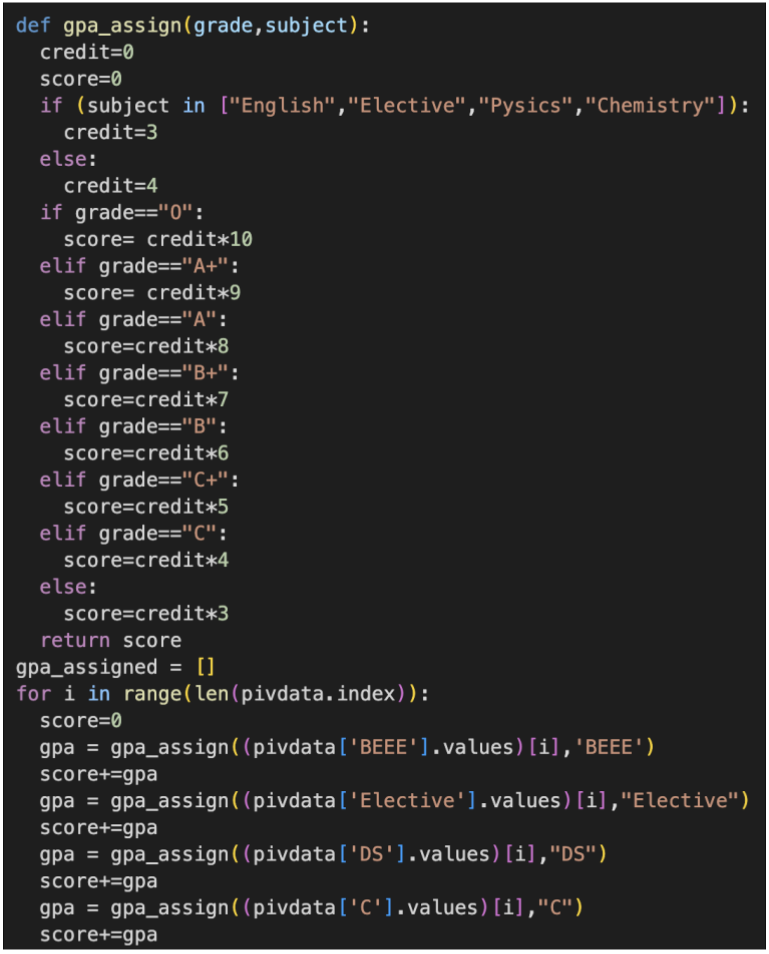
5(2) Create a PivotTable which takes the register number as index and subject names as columns and grades as value.



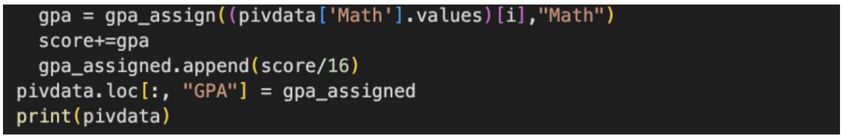
```
data2 = pd.concat([data1,df['Subject']], axis=1)
data2 = data2[['Register Number', 'Name', 'CAT1', 'CAT2', 'CAT3', 'Internal', 'ESE','Grade','Subject']]
def concatenate_strings(values):
  return ', '.join(values)
pivdata = pd.pivot_table(data2, values="Grade", index='Register Number',columns='Subject',aggfunc=concatenate_strings)
print(pivdata)
```

```
Subject         BEEE   C  DS Elective Math
Register Number
1293903281585      C   D  C+        D   C+
1387056938835     B+   D   D       C+    D
1497369857949      D   D  C+        D    B
1897314155067      B   A   D       C+    D
2003134404841     C+   D   D       C+    C
...              ...  ..  ..      ...  ...
9407580353282      B  C+  B+        B    D
9469605090648      D  C+   C       C+    C
9470253316296      C   B   D        D    D
9703244375471      D   D  C+        D    D
9806796186801      D   D   C        D   C+

[68 rows x 5 columns]
```

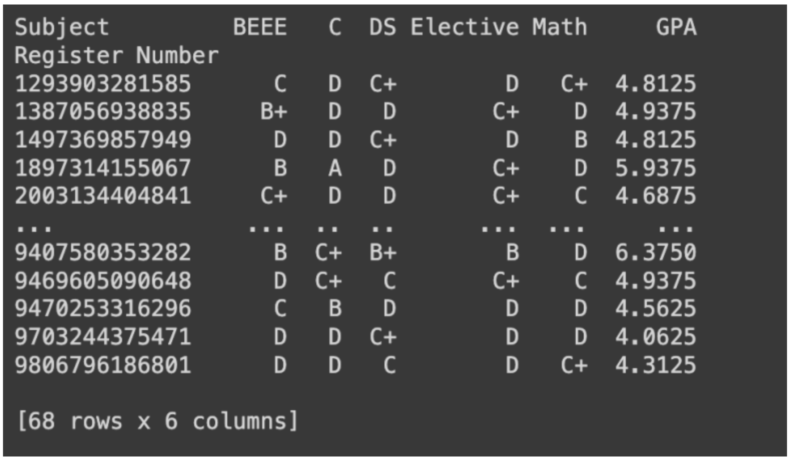
5(3) Add a column named GPA to the new DataFrame created using Pivot Table



```
def gpa_assign(grade,subject):
  credit=0
  score=0
  if (subject in ["English","Elective","Pysics","Chemistry"]):
    credit=3
  else:
    credit=4
  if grade=="O":
    score= credit*10
  elif grade=="A+":
    score= credit*9
  elif grade=="A":
    score=credit*8
  elif grade=="B+":
    score=credit*7
  elif grade=="B":
    score=credit*6
  elif grade=="C+":
    score=credit*5
  elif grade=="C":
    score=credit*4
  else:
    score=credit*3
  return score
gpa_assigned = []
for i in range(len(pivdata.index)):
  score=0
  gpa = gpa_assign((pivdata['BEEE'].values)[i],'BEEE')
  score+=gpa
  gpa = gpa_assign((pivdata['Elective'].values)[i],"Elective")
  score+=gpa
  gpa = gpa_assign((pivdata['DS'].values)[i],"DS")
  score+=gpa
  gpa = gpa_assign((pivdata['C'].values)[i],"C")
  score+=gpa
```

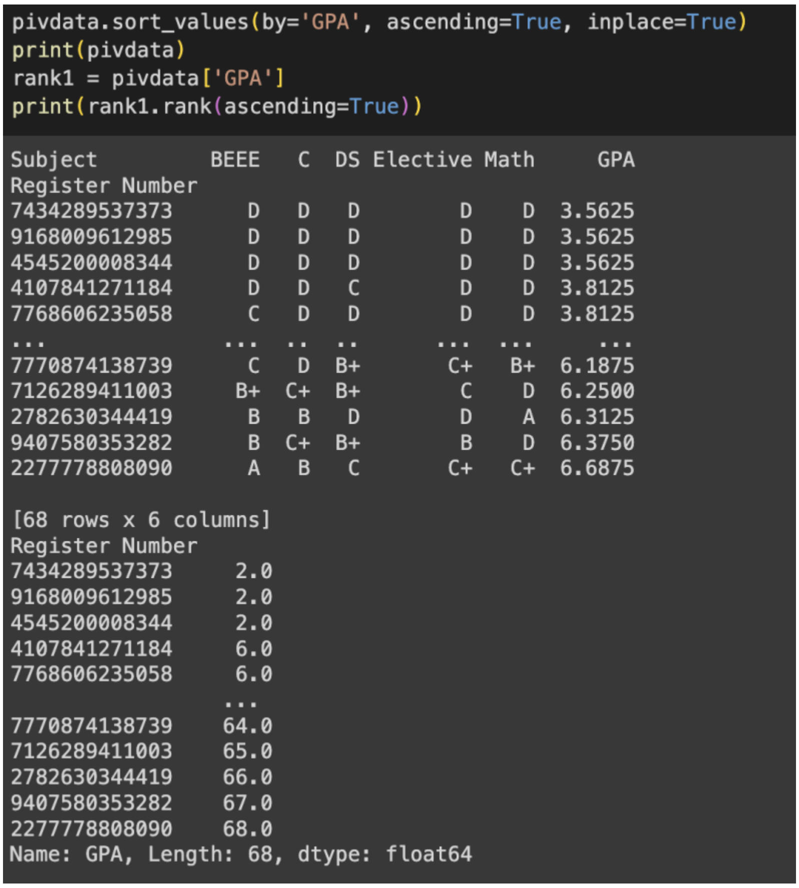


```
  gpa = gpa_assign((pivdata['Math'].values)[i],"Math")
  score+=gpa
  gpa_assigned.append(score/16)
pivdata.loc[:, "GPA"] = gpa_assigned
print(pivdata)
```



```
Subject          BEEE   C  DS Elective Math     GPA
Register Number
1293903281585       C   D  C+        D   C+  4.8125
1387056938835      B+   D   D       C+    D  4.9375
1497369857949       D   D  C+        D    B  4.8125
1897314155067       B   A   D       C+    D  5.9375
2003134404841      C+   D   D       C+    C  4.6875
...               ...  ..  ..      ...  ...     ...
9407580353282       B  C+  B+        B    D  6.3750
9469605090648       D  C+   C       C+    C  4.9375
9470253316296       C   B   D        D    D  4.5625
9703244375471       D   D  C+        D    D  4.0625
9806796186801       D   D   C        D   C+  4.3125

[68 rows x 6 columns]
```

5(4) Sort and rank the table according to GPA obtained by the students.

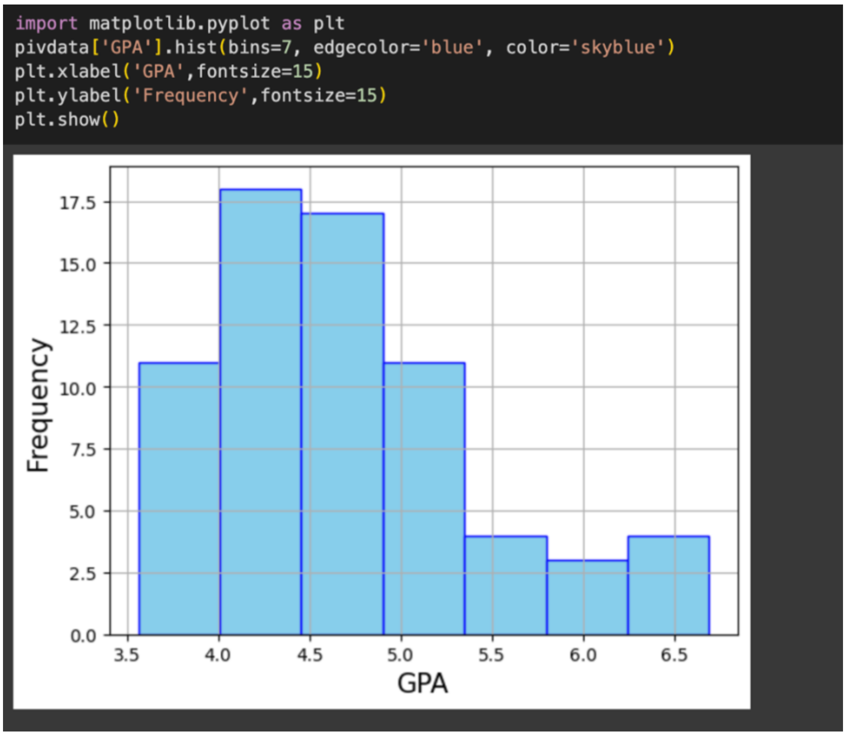


```
pivdata.sort_values(by='GPA', ascending=True, inplace=True)
print(pivdata)
rank1 = pivdata['GPA']
print(rank1.rank(ascending=True))
```

```
Subject          BEEE   C  DS Elective Math     GPA
Register Number
7434289537373       D   D   D        D    D  3.5625
9168009612985       D   D   D        D    D  3.5625
4545200008344       D   D   D        D    D  3.5625
4107841271184       D   D   C        D    D  3.8125
7768606235058       C   D   D        D    D  3.8125
...               ...  ..  ..      ...  ...     ...
7770874138739       C   D  B+       C+   B+  6.1875
7126289411003      B+  C+  B+        C    D  6.2500
2782630344419       B   B   D        D    A  6.3125
9407580353282       B  C+  B+        B    D  6.3750
2277778808090       A   B   C       C+   C+  6.6875

[68 rows x 6 columns]
Register Number
7434289537373      2.0
9168009612985      2.0
4545200008344      2.0
4107841271184      6.0
7768606235058      6.0
                 ...
7770874138739     64.0
7126289411003     65.0
2782630344419     66.0
9407580353282     67.0
2277778808090     68.0
Name: GPA, Length: 68, dtype: float64
```

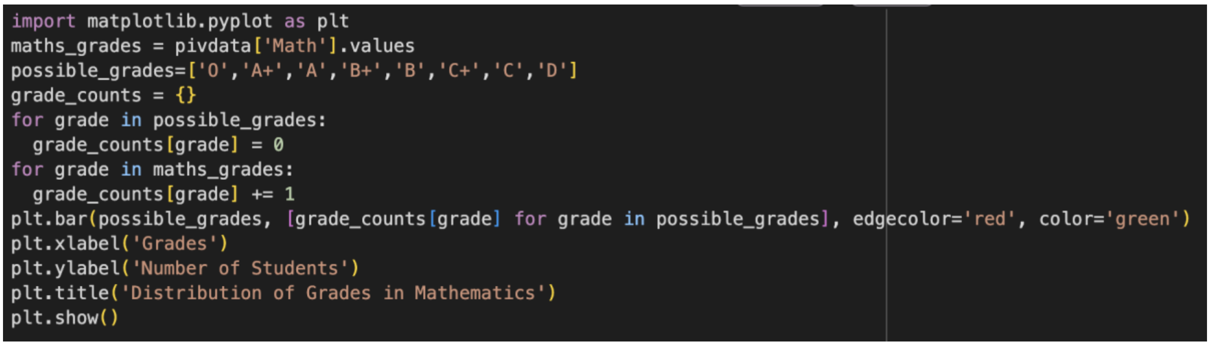
DATA VISUALIZATION:

6(1) Create a histogram of GPA of the students with 7 bins, “blue” edge color and “skyblue” fill color.

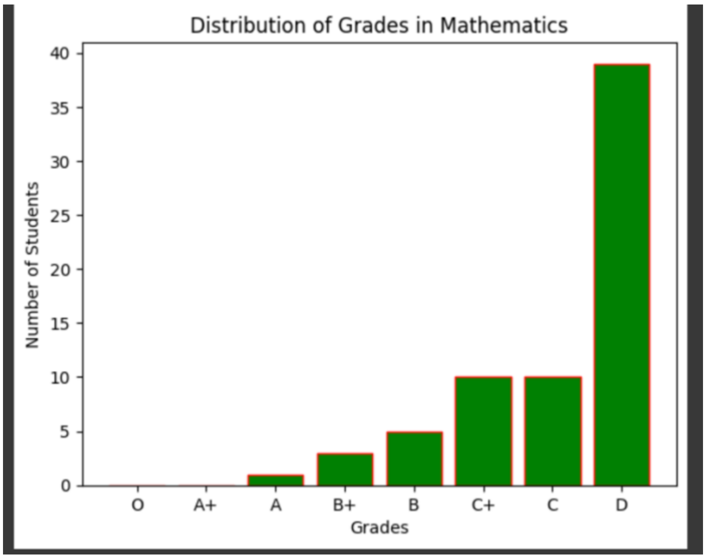


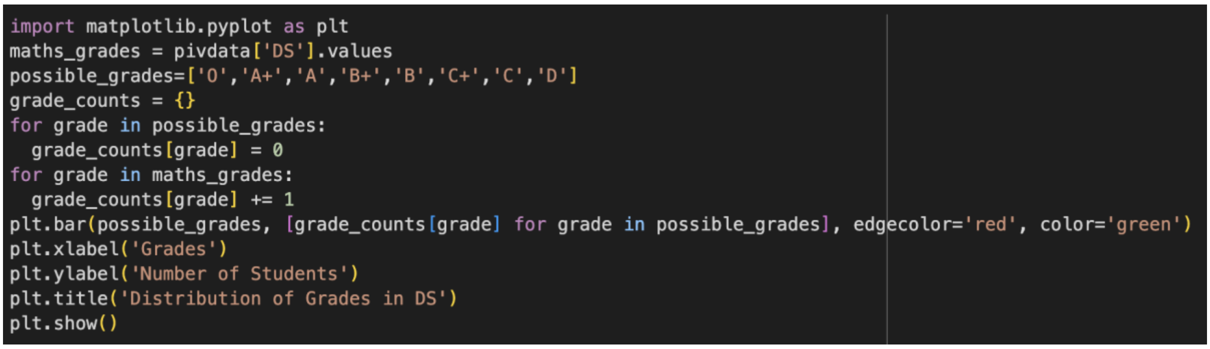
```
import matplotlib.pyplot as plt
pivdata['GPA'].hist(bins=7, edgecolor='blue', color='skyblue')
plt.xlabel('GPA',fontsize=15)
plt.ylabel('Frequency',fontsize=15)
plt.show()
```

6(2) Create a bar chart with “red” edge color and “green” fill color, showing grade distribution of each course/subject.

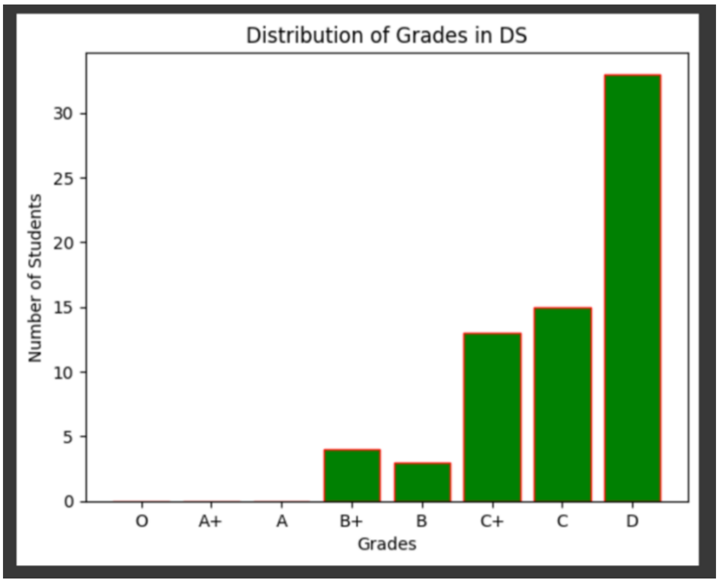


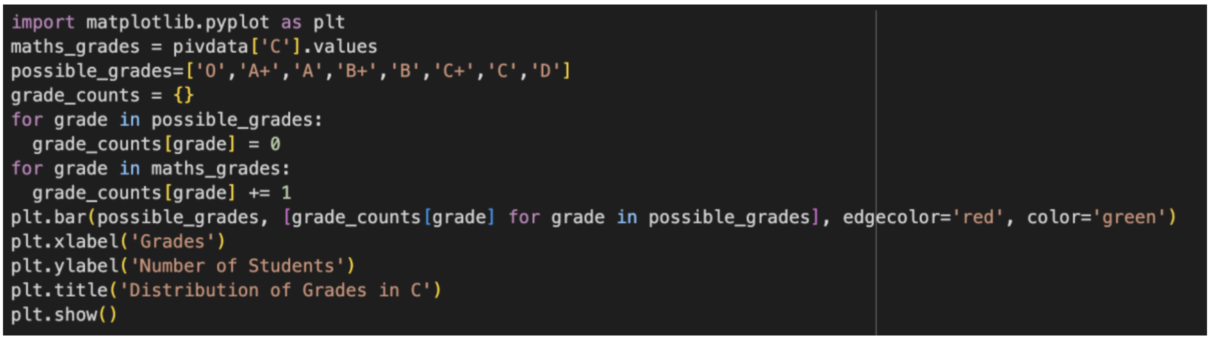
```
import matplotlib.pyplot as plt
maths_grades = pivdata['Math'].values
possible_grades=['O','A+','A','B+','B','C+','C','D']
grade_counts = {}
for grade in possible_grades:
  grade_counts[grade] = 0
for grade in maths_grades:
  grade_counts[grade] += 1
plt.bar(possible_grades, [grade_counts[grade] for grade in possible_grades], edgecolor='red', color='green')
plt.xlabel('Grades')
plt.ylabel('Number of Students')
plt.title('Distribution of Grades in Mathematics')
plt.show()
```



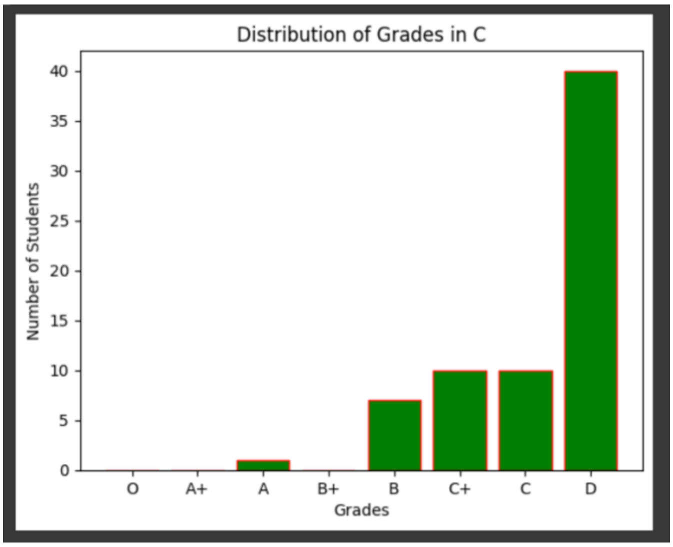


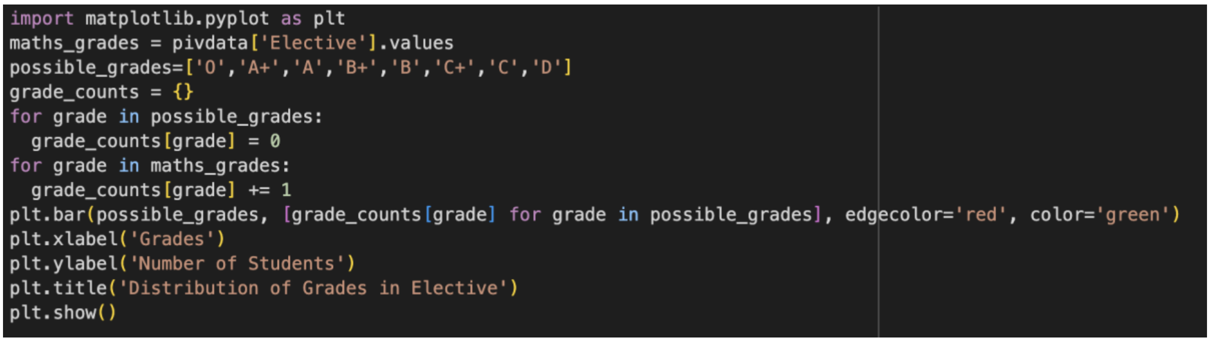
```
import matplotlib.pyplot as plt
maths_grades = pivdata['DS'].values
possible_grades=['O','A+','A','B+','B','C+','C','D']
grade_counts = {}
for grade in possible_grades:
  grade_counts[grade] = 0
for grade in maths_grades:
  grade_counts[grade] += 1
plt.bar(possible_grades, [grade_counts[grade] for grade in possible_grades], edgecolor='red', color='green')
plt.xlabel('Grades')
plt.ylabel('Number of Students')
plt.title('Distribution of Grades in DS')
plt.show()
```



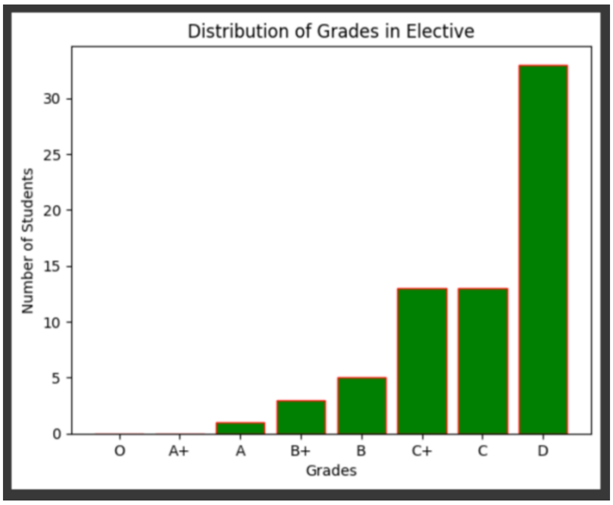


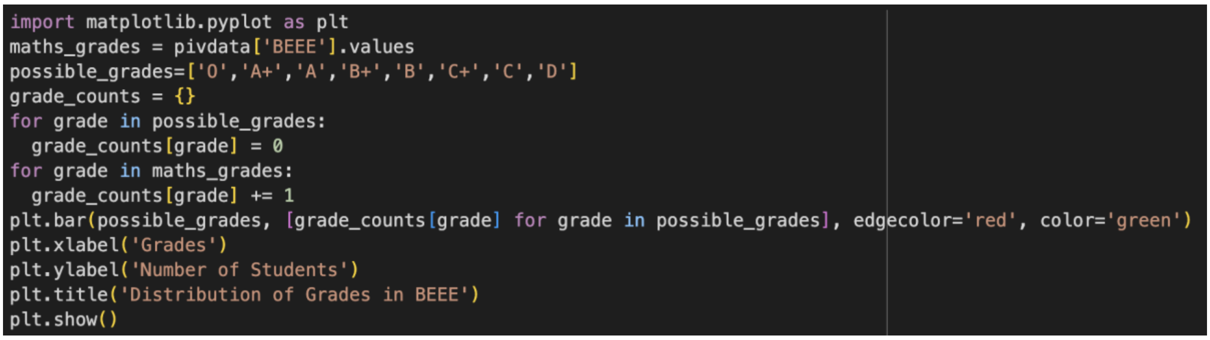
```
import matplotlib.pyplot as plt
maths_grades = pivdata['C'].values
possible_grades=['O','A+','A','B+','B','C+','C','D']
grade_counts = {}
for grade in possible_grades:
  grade_counts[grade] = 0
for grade in maths_grades:
  grade_counts[grade] += 1
plt.bar(possible_grades, [grade_counts[grade] for grade in possible_grades], edgecolor='red', color='green')
plt.xlabel('Grades')
plt.ylabel('Number of Students')
plt.title('Distribution of Grades in C')
plt.show()
```



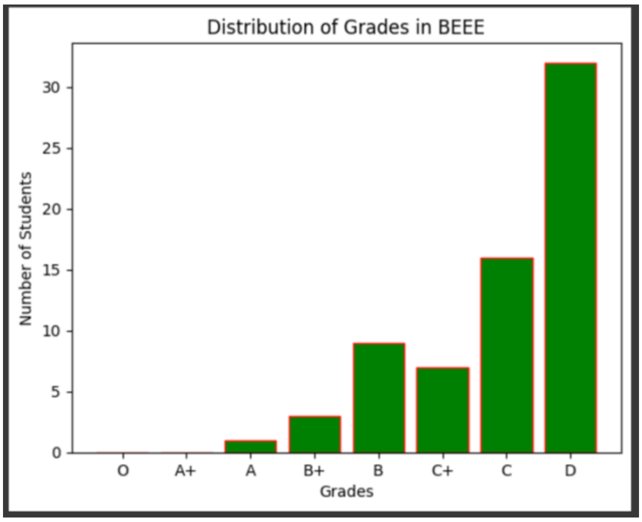


```
import matplotlib.pyplot as plt
maths_grades = pivdata['Elective'].values
possible_grades=['O','A+','A','B+','B','C+','C','D']
grade_counts = {}
for grade in possible_grades:
  grade_counts[grade] = 0
for grade in maths_grades:
  grade_counts[grade] += 1
plt.bar(possible_grades, [grade_counts[grade] for grade in possible_grades], edgecolor='red', color='green')
plt.xlabel('Grades')
plt.ylabel('Number of Students')
plt.title('Distribution of Grades in Elective')
plt.show()
```





```
import matplotlib.pyplot as plt
maths_grades = pivdata['BEEE'].values
possible_grades=['O','A+','A','B+','B','C+','C','D']
grade_counts = {}
for grade in possible_grades:
  grade_counts[grade] = 0
for grade in maths_grades:
  grade_counts[grade] += 1
plt.bar(possible_grades, [grade_counts[grade] for grade in possible_grades], edgecolor='red', color='green')
plt.xlabel('Grades')
plt.ylabel('Number of Students')
plt.title('Distribution of Grades in BEEE')
plt.show()
```


Distribution of Grades in BEEE
Number of Students
O
A+
A
B+
B
C+
C
D
Grades