

## Assessment

I am going to provide two .csv files , you are supposed to work on them and have to provide solutions to the following problems

import necessary libraries

```
In [1]: import pandas as pd
```

merge those two csv files (after getting as dataframes, get them as a single dataframe)

```
In [38]: import pandas as pd
#df_college1 = pd.read_csv("C:/Program Files/Python310/Guvi_Assignments/college_1.csv")
#df_college2 = pd.read_csv("C:/Program Files/Python310/Guvi_Assignments/college_2.csv")

#print(df_college1.head)
#print(df_college2.head)

# merging two csv files , get them as single dataframe
df_college12 = pd.concat(map(pd.read_csv, ['C:/Program Files/Python310/Guvi_Assignments
#debug file
#df_college12.to_csv('debug_merged.csv', index=False)
print(df_college12)
```

	Name	python	mysql	Previous	Geekions	CodeKata	Score	\
0	A.Dharani	82.0	20.0		24500		24500	
1	V.JEEVITHA	82.0	20.0		21740		21740	
2	HEMAVATHI.R	100.0	100.0		19680		19680	
3	Mugunthan S	100.0	47.0		10610		10610	
4	Sathammai.S	100.0	8.0		8980		8980	
..	...	...	...		...		...	
114	praveen raj j	24.0	0.0		2380		2380	
115	AMARNATH D	-1.0	12.0		1890		1890	
116	bala	32.0	0.0		1720		1720	
117	XY Z	-1.0	-1.0		0		0	
118	Hariharan	-1.0	-1.0		0		0	

	Department	Rising	python_en	\
0	Computer Science and Engineering	0	NaN	
1	Computer Science and Engineering	0	NaN	
2	Computer Science and Engineering	0	NaN	
3	Computer Science and Engineering	0	NaN	
4	Computer Science and Engineering	0	NaN	
..	...	...	...	
114	Computer Science and Engineering	0	-1.0	
115	Electronics and Communication Engineering	0	52.0	
116	Electronics and Communication Engineering	0	49.0	
117	Computer Science and Engineering	0	20.0	
118	Computer Science and Engineering	0	-1.0	

	computational_thinking
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN
..	...
114	0.0
115	-1.0

```

116                 -1.0
117                 -1.0
118                 0.0

```

[119 rows x 9 columns]

Take each csv file , split that csv file into multiple categories (example csv files are added in the repo)

consider if the codekata score exceeds 15000 points(present week) then make a csv on those observations as Exceeded\_expectations.csv

if 10000<codekata score<15000 (Reached\_expectations.csv)

if 7000<codekata score<10000 (Needs\_Improvement.csv)

if codekata score < 7000 (Unsatisfactory.csv)

```

In [27]: Exceeded_expectations = df_college12[df_college12['CodeKata Score'] > 15000]
Exceeded_expectations.to_csv('Exceeded_expectations.csv', index=False)

```

```

In [37]: Reached_expectations = df_college12[(df_college12['CodeKata Score'] > 10000) & (df_college12['CodeKata Score'] < 15000)]
Reached_expectations.to_csv('Reached_expectations.csv', index=False)

Needs_Improvement = df_college12[(df_college12['CodeKata Score'] > 7000) & (df_college12['CodeKata Score'] < 10000)]
Needs_Improvement.to_csv('Needs_Improvement.csv', index=False)

Unsatisfactory = df_college12[df_college12['CodeKata Score'] < 7000]
Unsatisfactory.to_csv('Unsatisfactory.csv', index=False)

```

Average of previous week geekions vs this week geekions (i.e Previous Geekions vs CodeKata Score)

```

In [41]: print(df_college12['Previous Geekions'].mean())
print(df_college12['CodeKata Score'].mean())

```

```

3842.3529411764707
4051.764705882353

```

No of students participated

```

In [42]: print(df_college12['Name'].count())

```

```

119

```

#Average completion of python course or my\_sql or python english or computational thinking

```

In [48]: #df_college12.fillna(0, inplace=True)
#df_college12.to_csv('debug_merged.csv', index=False)

print(df_college12['python'].mean())
print(df_college12['mysql'].mean())
print(df_college12['computational_thinking'].mean())

```

```

54.35294117647059
23.5
0.6470588235294118

```

rising star of the week (top 3 candidate who performed well in that particular week)

```

In [68]: df_college12['Rising'].sort_values(ascending=False)
df_college12.nlargest(3, 'Rising')

```

Out[68]:

	Name	python	mysql	Previous Geekions	CodeKata Score	Department	Rising	python_en	computational_
92	shifak N	58.0	0.0	5180	8320	Electronics and Electrical Engineering	3140	0.0	
86	Ganesh Ramkumar R	-1.0	24.0	8790	10790	Computer Science and Engineering	2000	55.0	
102	Narasimhan Y L	-1.0	0.0	4800	6800	Computer Science and Engineering	2000	-1.0	

Shining stars of the week (top 3 candidates who has highest geekions)

In [67]:

```
df_college12['Previous Geekions'].sort_values(ascending=False)
df_college12.nlargest(3, 'Previous Geekions')
```

Out[67]:

	Name	python	mysql	Previous Geekions	CodeKata Score	Department	Rising	python_en	computational_t
0	A.Dharani	82.0	20.0	24500	24500	Computer Science and Engineering	0	0.0	
1	V.JEEVITHA	82.0	20.0	21740	21740	Computer Science and Engineering	0	0.0	
2	HEMAVATHI.R	100.0	100.0	19680	19680	Computer Science and Engineering	0	0.0	

Department wise codekata performance (pie chart)

In [97]:

```
import matplotlib.pyplot as plot

cs = df_college12[(df_college12['Department'] == 'Computer Science and Engineering')]
ece = df_college12[(df_college12['Department'] == 'Electronics and Communication Engine
ee = df_college12[(df_college12['Department'] == 'Electronics and Electrical Engineerin

cs_mean = cs['CodeKata Score'].mean()
ece_mean = ece['CodeKata Score'].mean()
ee_mean = ee['CodeKata Score'].mean()

print(cs_mean)
print(ece_mean)
print(ee_mean)

data = [cs_mean,ece_mean,ee_mean]

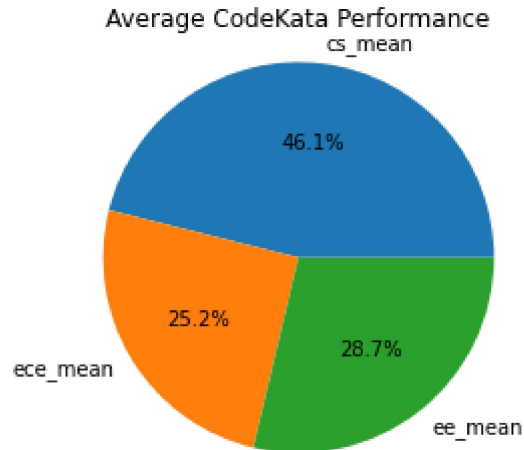
# df = DataFrame(total)
```

```
my_labels = 'cs_mean', 'ece_mean', 'ee_mean'
plot.pie(data, labels=my_labels, autopct='%1.1f%%')
plot.title('Average CodeKata Performance')
plot.axis('equal')
plot.show()
```

5079.761904761905

2777.8205128205127

3164.705882352941



Department wise toppers (horizontal bar graph or any visual representations of your choice)

In [15]:

```
import matplotlib.pyplot as plot
import numpy as np
import pandas as pd
# Considering highest codekata score for selecting topper(s)
df_college12 = pd.concat(map(pd.read_csv, ['C:/Program Files/Python310/Guvi_Assignments

cs = df_college12[(df_college12['Department'] == 'Computer Science and Engineering')]
ece = df_college12[(df_college12['Department'] == 'Electronics and Communication Engine
ee = df_college12[(df_college12['Department'] == 'Electronics and Electrical Engineerin

cs_sorted = cs.sort_values(['CodeKata Score'], ascending=[False])
ece_sorted = ece.sort_values(['CodeKata Score'], ascending=[False])
ee_sorted = ee.sort_values(['CodeKata Score'], ascending=[False])

# cs_sorted_topper = cs_sorted.head(1)

data_y = [cs_sorted['Name'].values[0] + " ( " + cs_sorted['Department'].values[0] + "

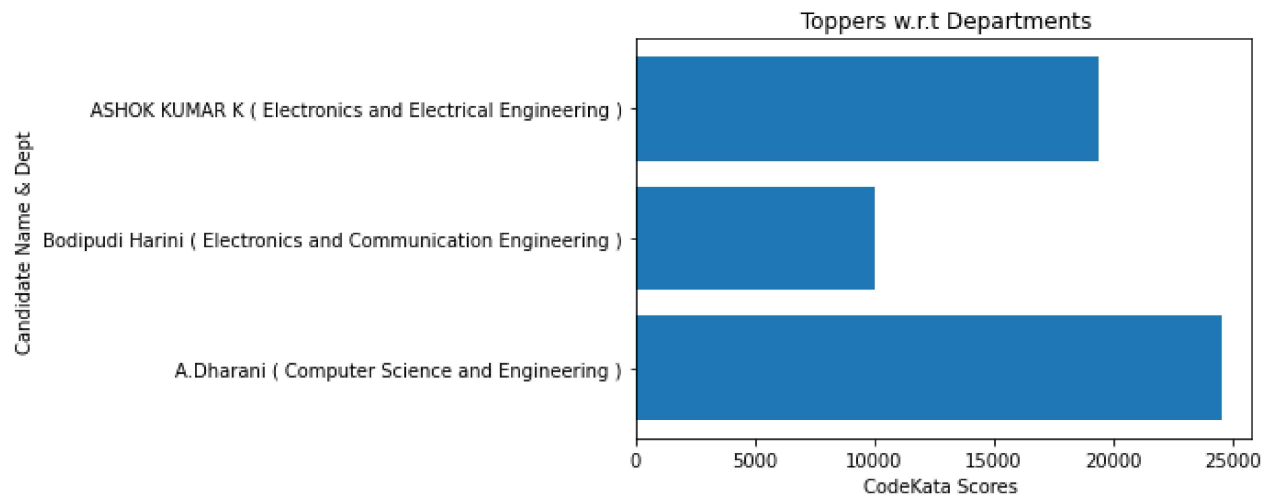
#debug
#first_value = cs_sorted['Name'].values[0]
#print(first_value)

# getting values against each value of y
data_x=[cs_sorted['CodeKata Score'].values[0],ece_sorted['CodeKata Score'].values[0],ee
plot.barh(data_y, data_x)

# setting label of y-axis
plot.ylabel("Candidate Name & Dept ")

# setting label of x-axis
```

```
plot.xlabel("CodeKata Scores ")  
plot.title(" Toppers w.r.t Departments ")  
plot.show()
```



In [ ]:

In [ ]:

In [ ]: