Case study DA

K.Vinay balaji reddy

2211cs010276 group-4

In [88]:
```
import pandas as pd
df = pd.read_excel("MIDMARKS.xlsx")
df
```

Out[88]:

	S.NO	SECTION	DV	M-II	PP	BEEE	FL	FIMS
0	1.0	ALPHA	12	0	17	9	19	15
1	2.0	ALPHA	19	12	16	16	18	3
2	3.0	ALPHA	18	14	18	18	18	16
3	4.0	ALPHA	15	9	19	17	19	15
4	5.0	ALPHA	18	17	19	19	20	18
...	...	...	...	...	...	...	...	...
713	NaN	ZETA	19	8	8	19	17	18
714	NaN	ZETA	12	1	7	10	20	8
715	NaN	ZETA	17	6	14	14	17	18
716	NaN	ZETA	12	1	6	7	15	12
717	NaN	ZETA	19	14	17	16	20	19

718 rows × 8 columns

Importing pandas and reading Excel file data

In [90]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 718 entries, 0 to 717
Data columns (total 8 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   S.NO     601 non-null    float64
 1   SECTION  691 non-null    object 
 2   DV       716 non-null    object 
 3   M-II     716 non-null    object 
 4   PP       716 non-null    object 
 5   BEEE     716 non-null    object 
 6   FL       715 non-null    object 
 7   FIMS     716 non-null    object 
dtypes: float64(1), object(7)
memory usage: 45.0+ KB
```

Shows Execl Data info

In [92]:
```
df[df['DV'].isnull()]
```

Out[92]:

	S.NO	SECTION	DV	M-II	PP	BEEE	FL	FIMS
564	565.0	SIGMA	NaN	NaN	NaN	NaN	NaN	NaN
601	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Filtering rows where 'DV' subject column is null

In [94]:
```
df.rename(columns={'M-II':'M2'}, inplace=True)
df
```

Out[94]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS
0	1.0	ALPHA	12	0	17	9	19	15
1	2.0	ALPHA	19	12	16	16	18	3
2	3.0	ALPHA	18	14	18	18	18	16
3	4.0	ALPHA	15	9	19	17	19	15
4	5.0	ALPHA	18	17	19	19	20	18
...	...	...	...	...	...	...	...	...
713	NaN	ZETA	19	8	8	19	17	18
714	NaN	ZETA	12	1	7	10	20	8
715	NaN	ZETA	17	6	14	14	17	18
716	NaN	ZETA	12	1	6	7	15	12
717	NaN	ZETA	19	14	17	16	20	19

718 rows × 8 columns

Renaming column 'M-II' to 'M2' in dataframe

In [96]:
```
columns_to_convert = ['DV', 'M2', 'PP', 'BEEE', 'FL', 'FIMS']
for col in columns_to_convert:
 df[col] = pd.to_numeric(df[col], errors='coerce')

df['Total'] = df['DV'] + df['M2'] + df['PP'] + df['BEEE'] + df['FL'] + df['FIMS'
df
```

Out[96]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	Total
0	1.0	ALPHA	12.0	0.0	17.0	9.0	19.0	15.0	72.0
1	2.0	ALPHA	19.0	12.0	16.0	16.0	18.0	3.0	84.0
2	3.0	ALPHA	18.0	14.0	18.0	18.0	18.0	16.0	102.0
3	4.0	ALPHA	15.0	9.0	19.0	17.0	19.0	15.0	94.0
4	5.0	ALPHA	18.0	17.0	19.0	19.0	20.0	18.0	111.0
...	...	...	...	...	...	...	...	...	...
713	NaN	ZETA	19.0	8.0	8.0	19.0	17.0	18.0	89.0
714	NaN	ZETA	12.0	1.0	7.0	10.0	20.0	8.0	58.0
715	NaN	ZETA	17.0	6.0	14.0	14.0	17.0	18.0	86.0
716	NaN	ZETA	12.0	1.0	6.0	7.0	15.0	12.0	53.0
717	NaN	ZETA	19.0	14.0	17.0	16.0	20.0	19.0	105.0

718 rows × 9 columns

In [97]:
```
df['PERCENTAGE'] = (df['Total'] /120)*100
print(df)
```

```
     S.NO SECTION    DV    M2    PP  BEEE    FL  FIMS  Total  PERCENTAGE
0     1.0   ALPHA  12.0   0.0  17.0   9.0  19.0  15.0   72.0   60.000000
1     2.0   ALPHA  19.0  12.0  16.0  16.0  18.0   3.0   84.0   70.000000
2     3.0   ALPHA  18.0  14.0  18.0  18.0  18.0  16.0  102.0   85.000000
3     4.0   ALPHA  15.0   9.0  19.0  17.0  19.0  15.0   94.0   78.333333
4     5.0   ALPHA  18.0  17.0  19.0  19.0  20.0  18.0  111.0   92.500000
..    ...     ...   ...   ...   ...   ...   ...   ...    ...         ...
713   NaN    ZETA  19.0   8.0   8.0  19.0  17.0  18.0   89.0   74.166667
714   NaN    ZETA  12.0   1.0   7.0  10.0  20.0   8.0   58.0   48.333333
715   NaN    ZETA  17.0   6.0  14.0  14.0  17.0  18.0   86.0   71.666667
716   NaN    ZETA  12.0   1.0   6.0   7.0  15.0  12.0   53.0   44.166667
717   NaN    ZETA  19.0  14.0  17.0  16.0  20.0  19.0  105.0   87.500000

[718 rows x 10 columns]
```

Calculating percentage based on 'Total' column values

In [99]:
```
def assign_grade(percentage):
    if percentage >= 90:
        return 'A'
    elif percentage >= 80:
        return 'B'
    elif percentage >= 70:
        return 'C'
    elif percentage >= 60:
        return 'D'
    else:
        return 'F'
```

```
df['GRADE'] = df['PERCENTAGE'].apply(assign_grade)
df
```

Out[99]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	Total	PERCENTAGE	GRADE
0	1.0	ALPHA	12.0	0.0	17.0	9.0	19.0	15.0	72.0	60.000000	D
1	2.0	ALPHA	19.0	12.0	16.0	16.0	18.0	3.0	84.0	70.000000	C
2	3.0	ALPHA	18.0	14.0	18.0	18.0	18.0	16.0	102.0	85.000000	B
3	4.0	ALPHA	15.0	9.0	19.0	17.0	19.0	15.0	94.0	78.333333	C
4	5.0	ALPHA	18.0	17.0	19.0	19.0	20.0	18.0	111.0	92.500000	A
...	...	...	...	...	...	...	...	...	...	...	...
713	NaN	ZETA	19.0	8.0	8.0	19.0	17.0	18.0	89.0	74.166667	C
714	NaN	ZETA	12.0	1.0	7.0	10.0	20.0	8.0	58.0	48.333333	F
715	NaN	ZETA	17.0	6.0	14.0	14.0	17.0	18.0	86.0	71.666667	C
716	NaN	ZETA	12.0	1.0	6.0	7.0	15.0	12.0	53.0	44.166667	F
717	NaN	ZETA	19.0	14.0	17.0	16.0	20.0	19.0	105.0	87.500000	B

718 rows × 11 columns

Assigning grades based on percentage values in dataframe

In [101…

```
df[df['SECTION'] == 'BETA']['GRADE'].value_counts()
```

Out[101…

```
GRADE
F    32
C    17
D    16
B    15
A    10
Name: count, dtype: int64
```

Counting grade frequencies for 'BETA' section in dataframe

In [103…

```
df[df['SECTION'] == 'ALPHA']['GRADE'].value_counts()
```

Out[103…

```
GRADE
A    23
F    23
C    18
B    16
D    10
Name: count, dtype: int64
```

Counting grade frequencies for 'ALPHA' section in dataframe

In [105…
```
df[df['SECTION'] == 'EPSILON']['GRADE'].value_counts()
```

Out[105…
```
GRADE
F    43
C    15
D    14
B    11
A     5
Name: count, dtype: int64
```

Counting grade frequencies for 'EPSILON' section in dataframe

In [107…
```
df[df['SECTION'] == 'GAMMA']['GRADE'].value_counts()
```

Out[107…
```
GRADE
F    31
C    21
D    15
B    14
A     9
Name: count, dtype: int64
```

Counting grade frequencies for 'GAMMA' section in dataframe

In [109…
```
df[df['SECTION'] == 'OMEGA']['GRADE'].value_counts()
```

Out[109…
```
GRADE
C    21
F    21
A    20
B    15
D    13
Name: count, dtype: int64
```

Counting grade frequencies for 'OMEGA' section in dataframe

In [111…
```
df[df['SECTION'] == 'SIGMA']['GRADE'].value_counts()
```

Out[111…
```
GRADE
A    20
F    14
B    13
D    10
C     6
Name: count, dtype: int64
```

Counting grade frequencies for 'SIGMA' section in dataframe

In [113…
```
df[df['SECTION'] == 'ZETA']['GRADE'].value_counts()
```

Out[113…
```
GRADE
F    36
C    20
D    16
B    14
A     4
Name: count, dtype: int64
```

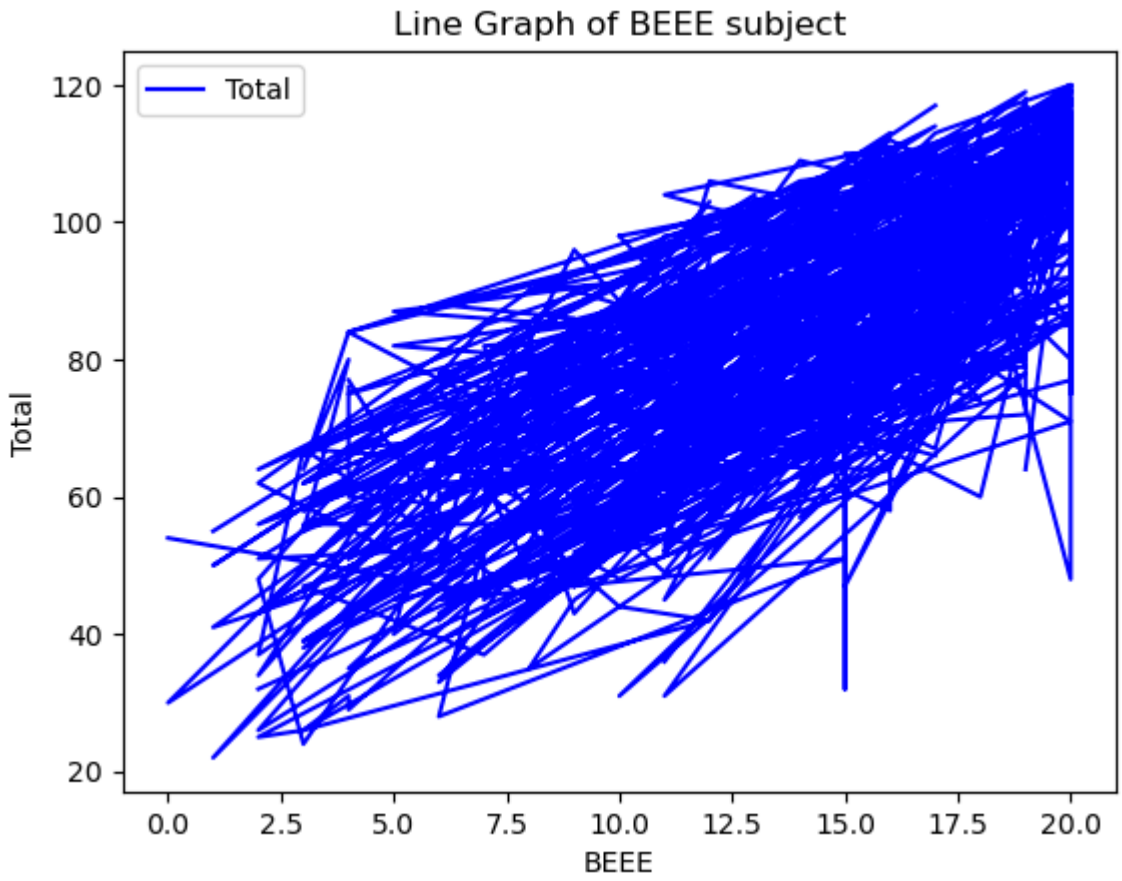
Counting grade frequencies for 'ZETA' section in dataframe

In [115…
```
a_grades = df[df['GRADE'] == 'A']
a_grades['SECTION'].value_counts()
```

Out[115…
```
SECTION
ALPHA      23
OMEGA      20
SIGMA      20
BETA       10
GAMMA       9
DELTA       8
EPSILON     5
ZETA        4
Name: count, dtype: int64
```

Counting sections with 'A' grades in dataframe

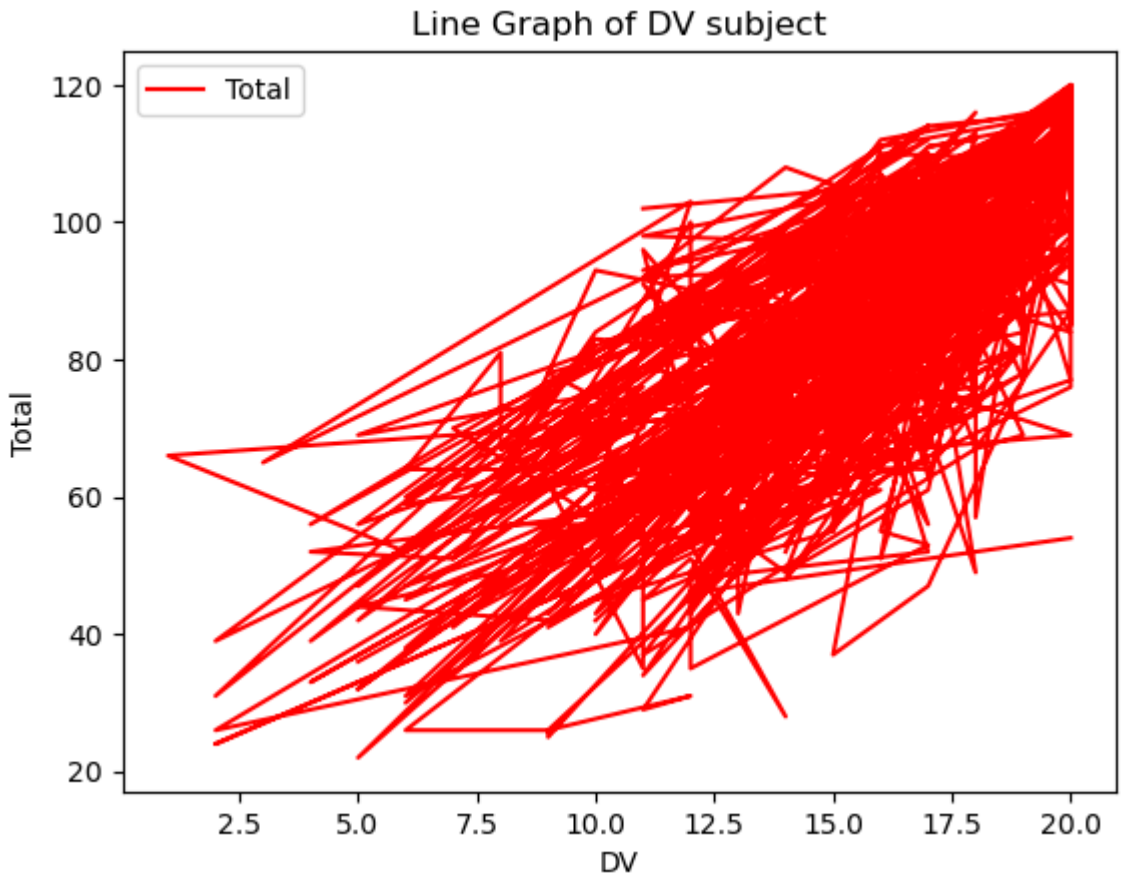
In [117…
```
df.plot.line(x='BEEE',y='Total',color='blue')
plt.title("Line Graph of BEEE subject")
plt.ylabel("Total")
plt.show()
```



Plotting line graph of 'BEEE' vs 'Total' values

In [119…

```
import matplotlib.pyplot as plt
df.plot.line(x='DV',y='Total',color='red')
plt.title("Line Graph of DV subject")
plt.ylabel("Total")
plt.show()
```



Plotting line graph of 'DV' vs 'Total' values

In [121…

```
DF=df.sort_values("Total",ascending=True)
DF
```

Out[121…

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	Total	PERCENTAGE	GRADE
190	191.0	DELTA	5.0	0.0	1.0	1.0	10.0	5.0	22.0	18.333333	F
611	NaN	NaN	2.0	0.0	0.0	3.0	10.0	9.0	24.0	20.000000	F
635	NaN	ZETA	9.0	0.0	2.0	2.0	3.0	9.0	25.0	20.833333	F
357	358.0	EPSILON	9.0	0.0	2.0	3.0	11.0	1.0	26.0	21.666667	F
636	NaN	ZETA	9.0	0.0	5.0	3.0	4.0	5.0	26.0	21.666667	F
...	...	...	...	...	...	...	...	...	...	...	...
628	NaN	ZETA	17.0	3.0	8.0	16.0	15.0	NaN	NaN	NaN	F
650	NaN	ZETA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	F
669	NaN	ZETA	16.0	6.0	NaN	NaN	8.0	9.0	NaN	NaN	F
673	NaN	ZETA	2.0	NaN	0.0	0.0	2.0	1.0	NaN	NaN	F
705	NaN	ZETA	16.0	0.0	11.0	16.0	20.0	NaN	NaN	NaN	F

718 rows × 11 columns

Sorting dataframe by 'Total' in ascending order

In [123…

```
d=df.loc[(df['Total']>=25) & (df['Total']<=75)]
d=d.reset_index()
d
```

Out[123...

	index	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	Total	PERCENTAGE	G
0	0	1.0	ALPHA	12.0	0.0	17.0	9.0	19.0	15.0	72.0	60.000000	
1	20	21.0	ALPHA	4.0	2.0	5.0	3.0	16.0	9.0	39.0	32.500000	
2	25	26.0	ALPHA	6.0	10.0	10.0	11.0	13.0	10.0	60.0	50.000000	
3	27	28.0	ALPHA	5.0	4.0	3.0	12.0	13.0	5.0	42.0	35.000000	
4	29	30.0	ALPHA	8.0	2.0	11.0	10.0	13.0	12.0	56.0	46.666667	
...	...	...	...	...	...	...	...	...	...	...	...	
227	700	NaN	ZETA	17.0	7.0	7.0	12.0	13.0	9.0	65.0	54.166667	
228	701	NaN	ZETA	15.0	4.0	3.0	13.0	14.0	13.0	62.0	51.666667	
229	710	NaN	ZETA	18.0	1.0	6.0	12.0	11.0	9.0	57.0	47.500000	
230	714	NaN	ZETA	12.0	1.0	7.0	10.0	20.0	8.0	58.0	48.333333	
231	716	NaN	ZETA	12.0	1.0	6.0	7.0	15.0	12.0	53.0	44.166667	

232 rows × 12 columns

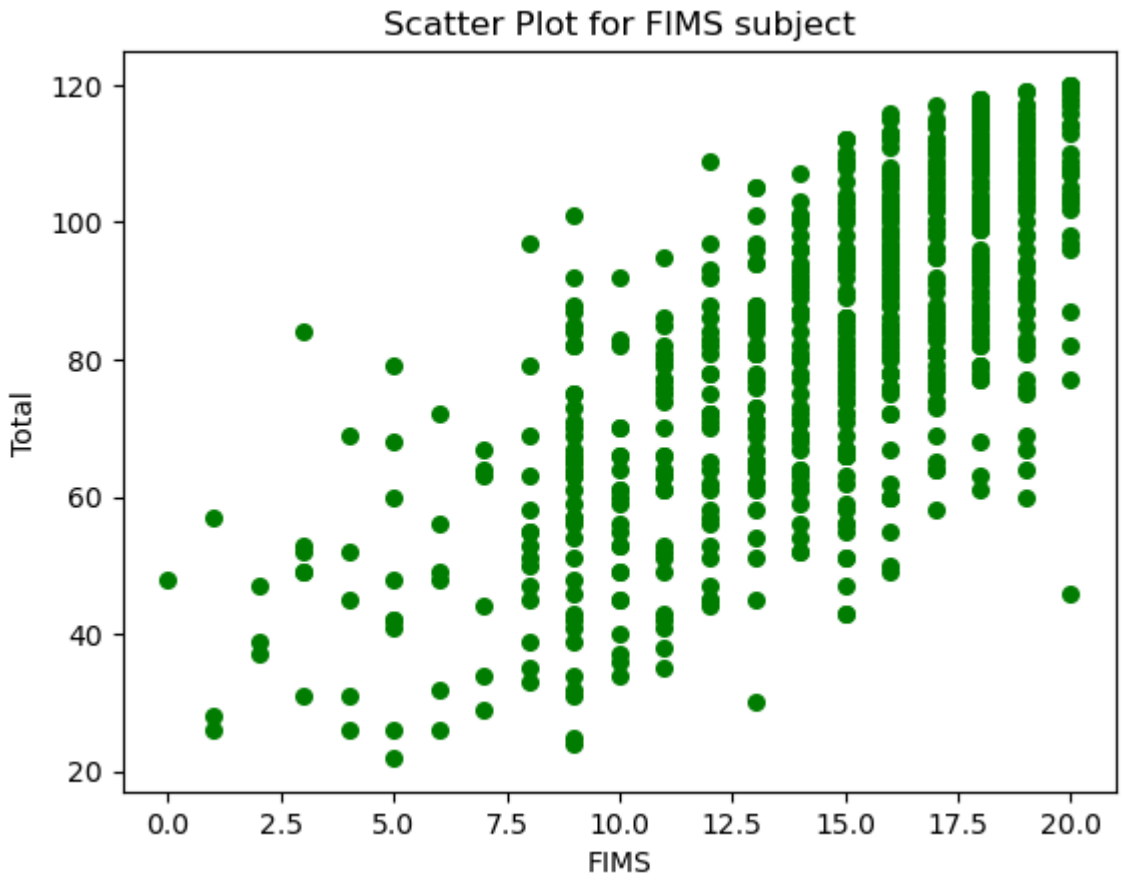
Filtering rows where 'Total' is between 25 and 75

In [125...

```
df.plot.scatter(x = 'FIMS', y = 'Total',color='green',s=30)
plt.title("Scatter Plot for FIMS subject")
```

Out[125...

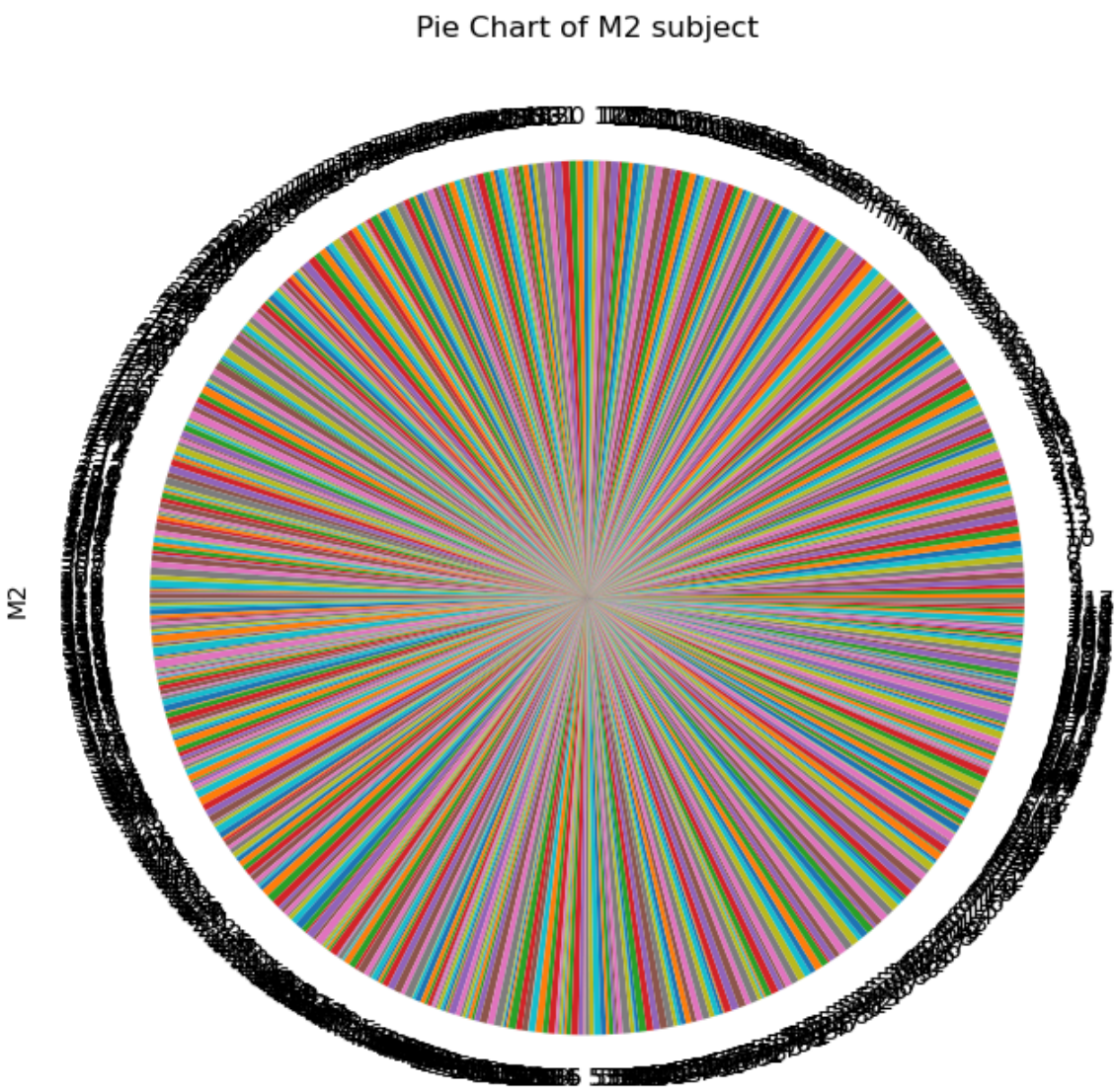
```
Text(0.5, 1.0, 'Scatter Plot for FIMS subject')
```



Creating scatter plot for 'FIMS' vs 'Total' values

In [127…
```
df['M2'].plot(kind='pie',subplots=True,figsize=(8,8))
plt.title("Pie Chart of M2 subject")
```

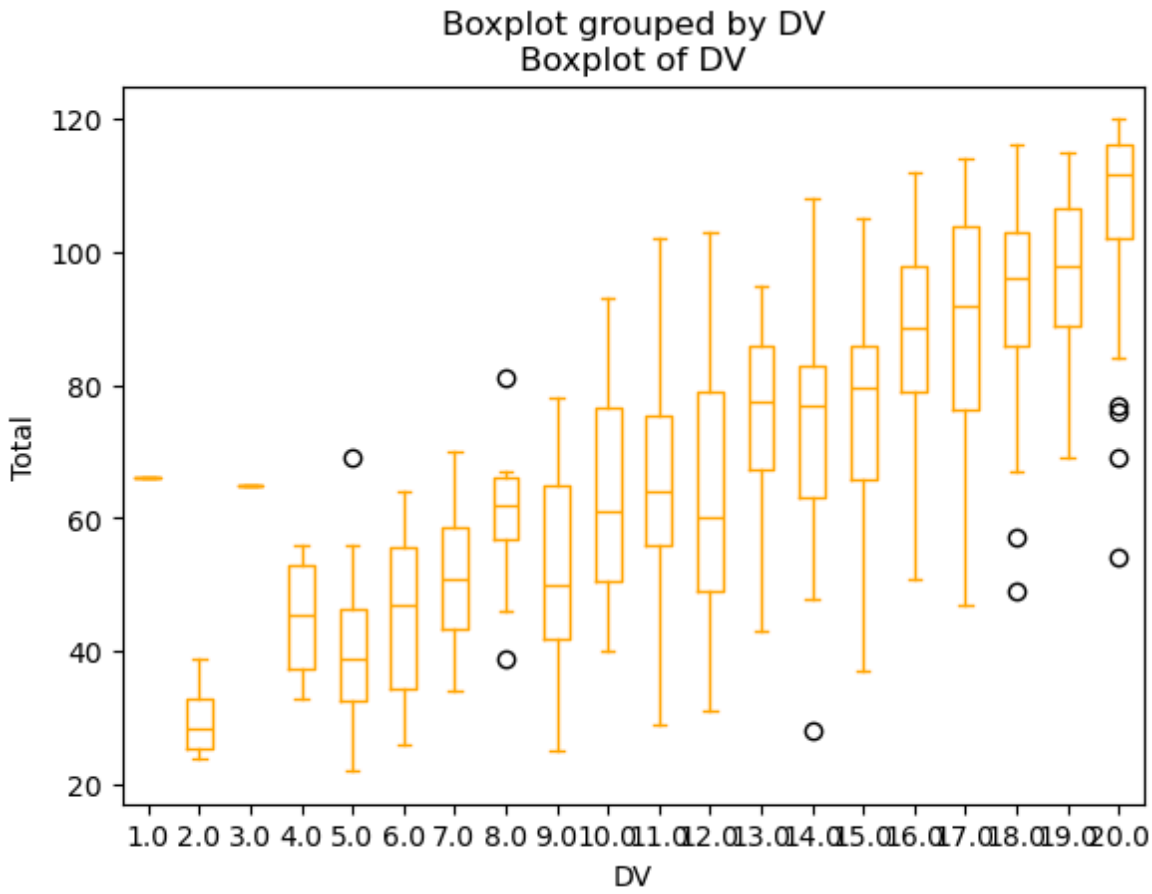
Out[127…
```
Text(0.5, 1.0, 'Pie Chart of M2 subject')
```



Creating pie chart for 'M2' subject data distribution

In [132…

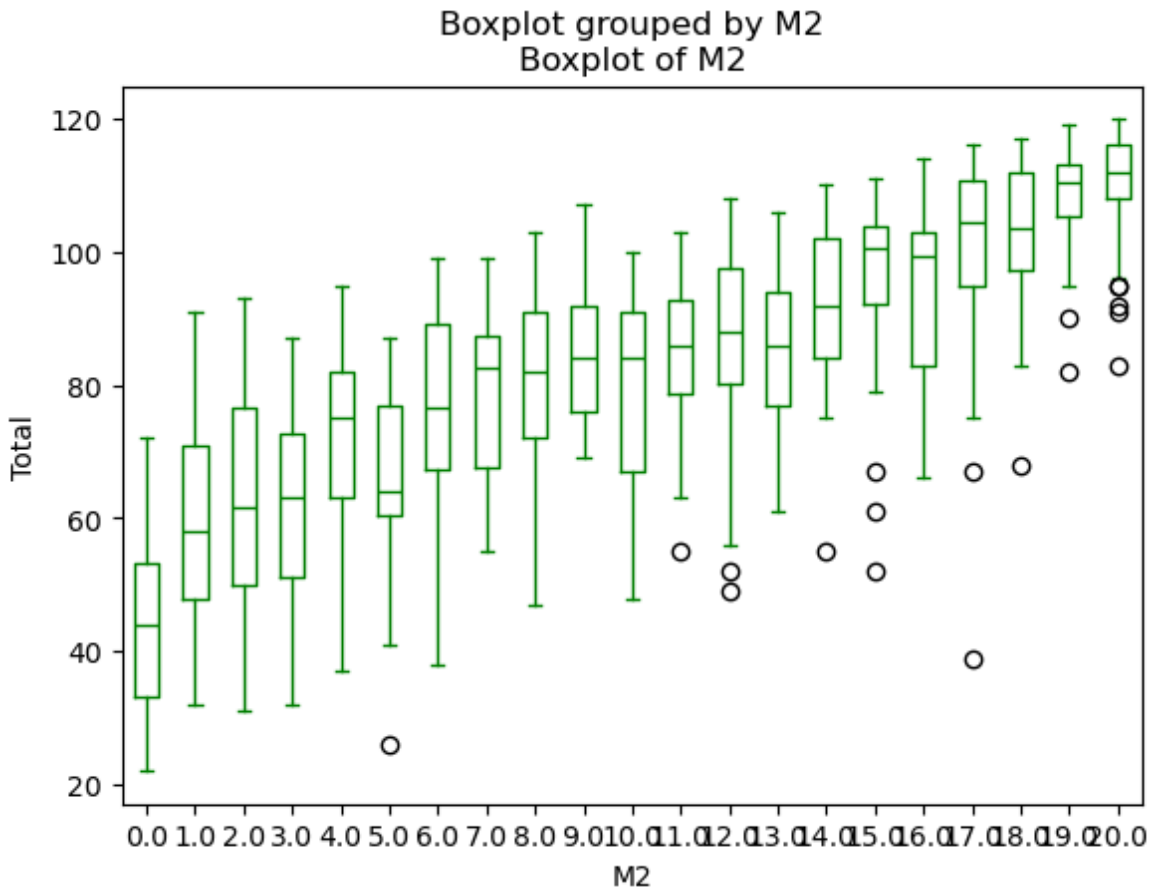
```
df.boxplot(by='DV', column =['Total'], grid = False,color='orange')
plt.title("Boxplot of DV")
plt.ylabel("Total")
plt.show()
```



Creating boxplot to visualize 'Total' distribution by 'DV'

In [134…

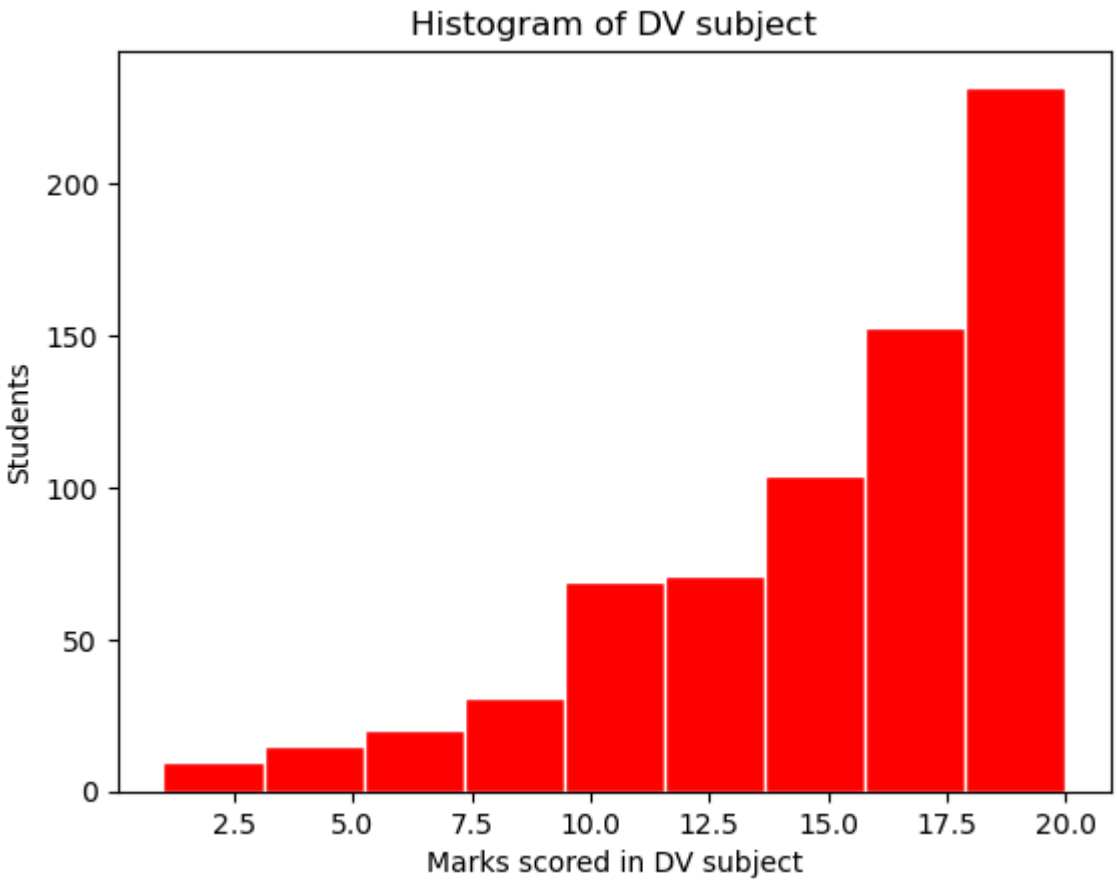
```
df.boxplot(by='M2', column =['Total'], grid = False,color='green')
plt.title("Boxplot of M2")
plt.ylabel("Total")
plt.show()
```



Creating boxplot to visualize 'Total' distribution by 'M2

In [136…
```
plt.hist(df['DV'],color='red',edgecolor='white',bins=9)
plt.xlabel("Marks scored in DV subject")
plt.ylabel("Students")
plt.title("Histogram of DV subject")
```

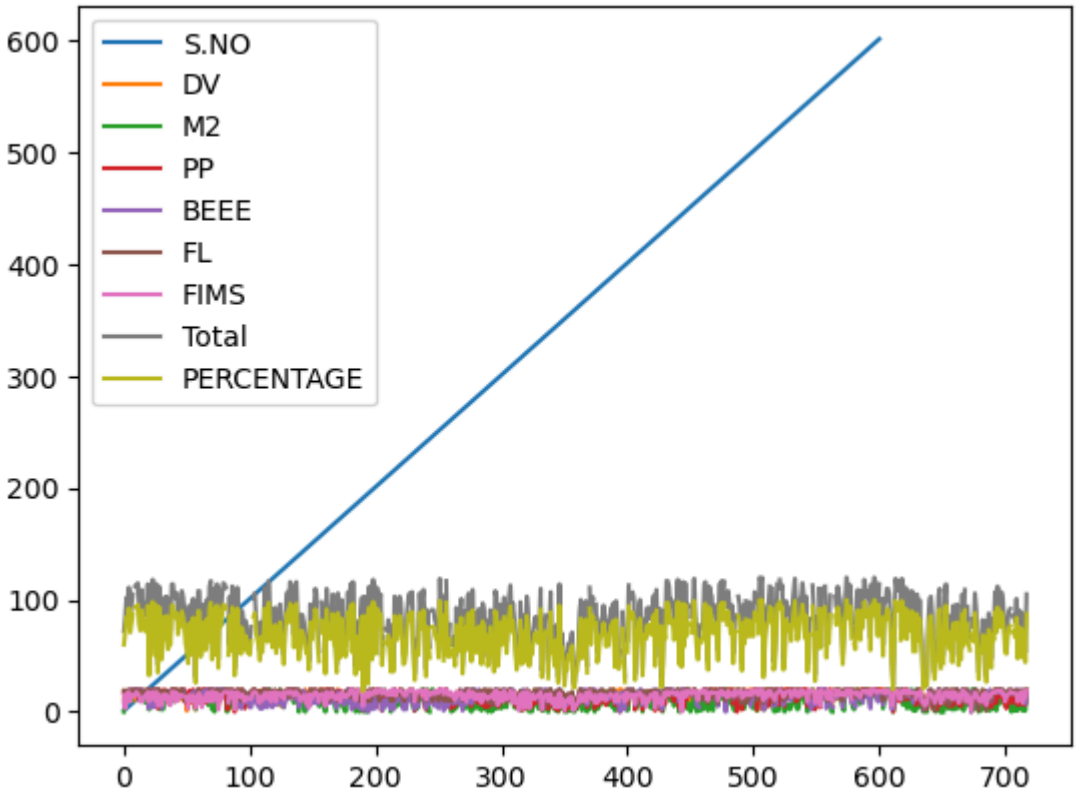
Out[136… Text(0.5, 1.0, 'Histogram of DV subject')



Creating histogram to visualize 'DV' subject marks distribution

In [138…

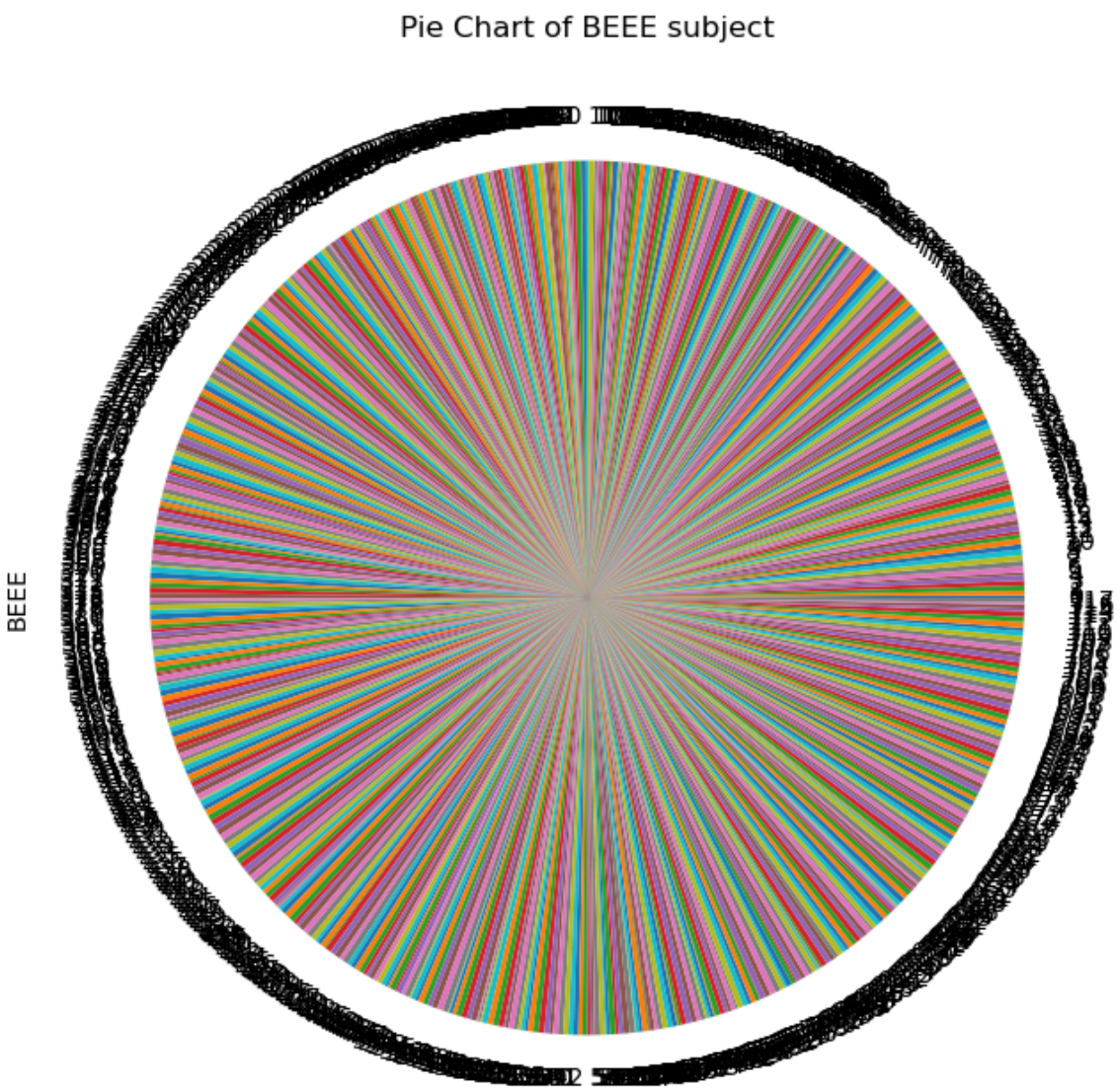
```
df.plot()
plt.show()
```



Plotting all columns in the dataframe for visualization

In [141…
```
df['BEEE'].plot(kind='pie',subplots=True,figsize=(8,8))
plt.title("Pie Chart of BEEE subject")
```

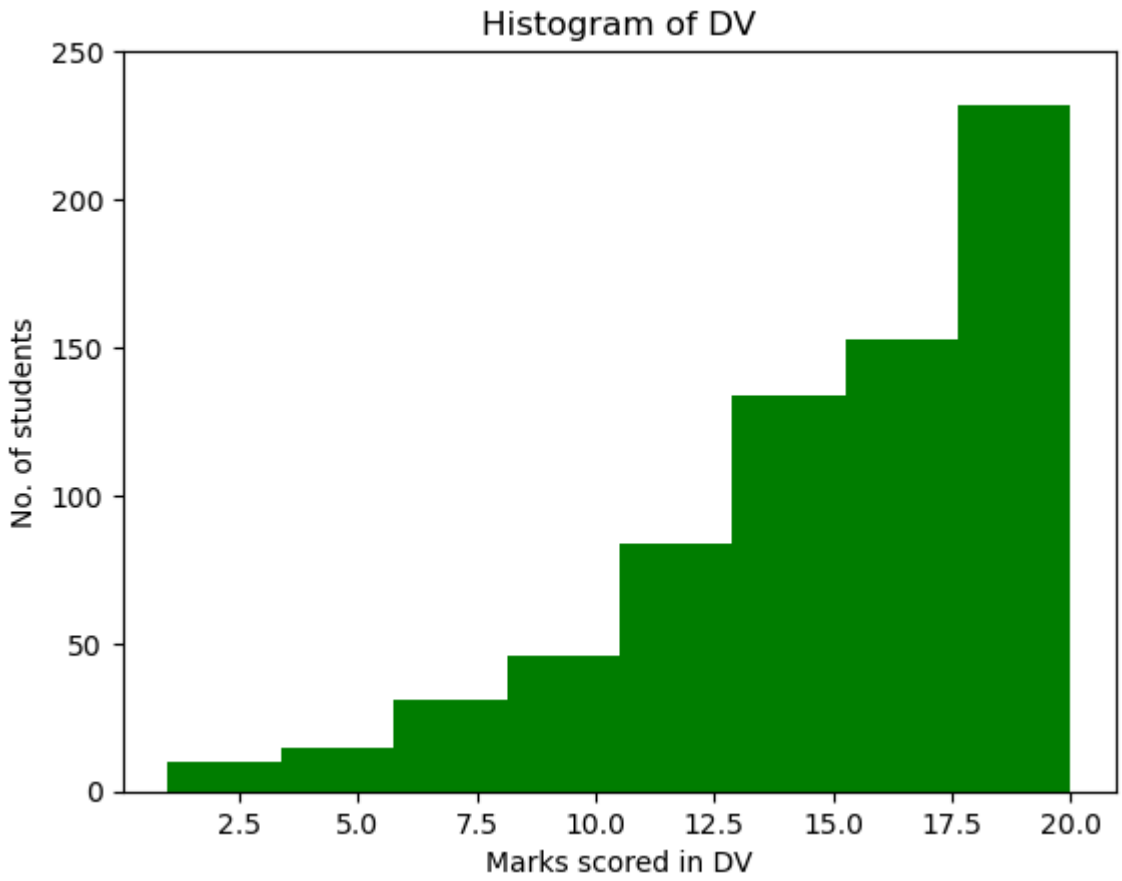
Out[141…
```
Text(0.5, 1.0, 'Pie Chart of BEEE subject')
```



Creating pie chart for 'BEEE' subject data distribution

In [145…

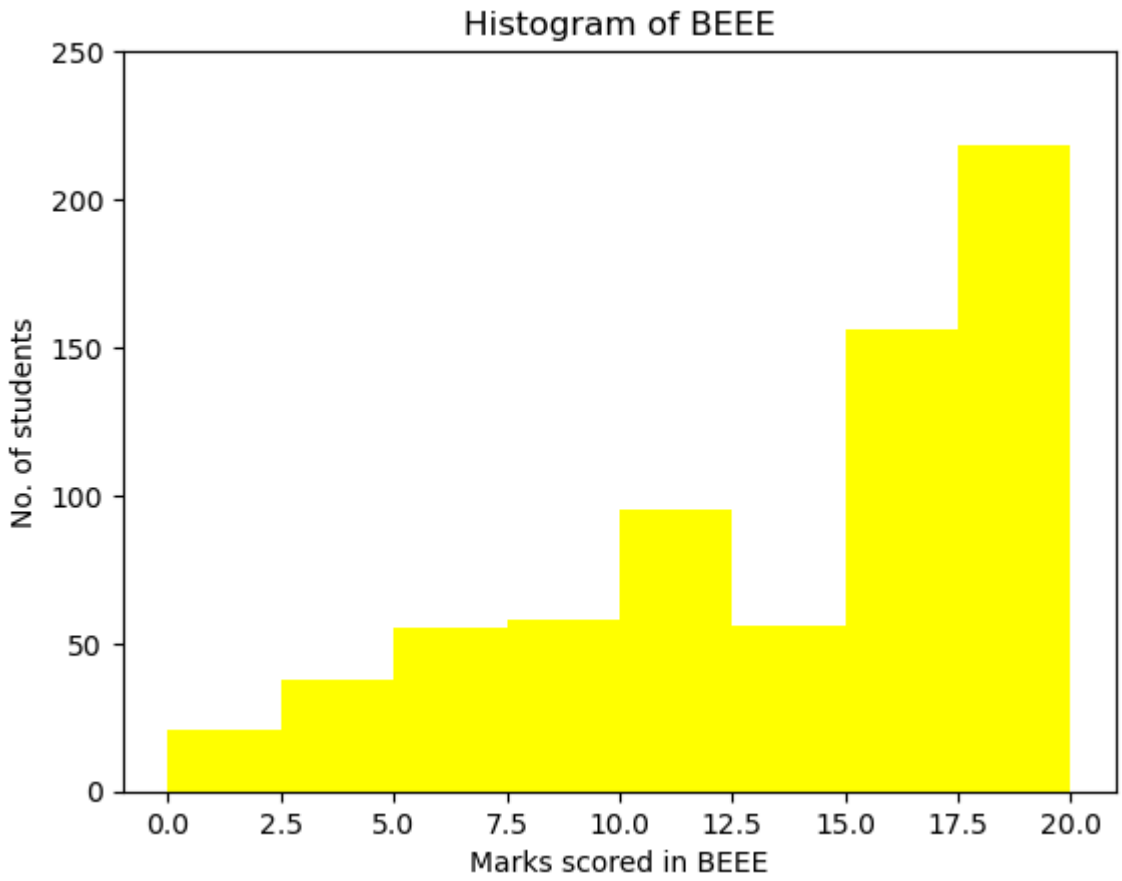
```
plt.hist(df['DV'], color='green', bins=8)
plt.ylim(0, 250)
plt.xlabel("Marks scored in DV")
plt.ylabel("No. of students")
plt.title("Histogram of DV")
plt.show()
```



Creating histogram to visualize 'DV' subject marks distribution

In [148…

```
plt.hist(df['BEEE'], color='yellow', bins=8)
plt.ylim(0, 250)
plt.xlabel("Marks scored in BEEE")
plt.ylabel("No. of students")
plt.title("Histogram of BEEE")
plt.show()
```



Creating histogram to visualize 'BEEE' subject marks distribution

In [ ]: