M.Yashwanth

2211cs010353

Group-4

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

In [9]:
```
p=pd.read_excel("MIDMARKS.xlsx")
```

In [10]:
```
p
```

Out[10]:

	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 [11]:
```
p.head(91)
```

Out[11]:

	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
...	...	...	...	...	...	...	...	...
86	87.0	ALPHA	17	18	19	20	20	18
87	88.0	ALPHA	13	17	14	19	15	17
88	89.0	ALPHA	2	17	0	3	15	2
89	90.0	ALPHA	10	6	15	10	15	10
90	91.0	BETA	17	19	20	17	20	18

91 rows × 8 columns

In [12]:
```
p
```

Out[12]:

	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

In [13]:
```
p.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
```

In [14]:
```
p['DV'] = pd.to_numeric(p['DV'], errors='coerce').astype('Int64')
p['M-II'] = pd.to_numeric(p['M-II'], errors='coerce').astype('Int64')
p['PP'] = pd.to_numeric(p['PP'], errors='coerce').astype('Int64')
p['BEEE'] = pd.to_numeric(p['BEEE'], errors='coerce').astype('Int64')
p['FL'] = pd.to_numeric(p['FL'], errors='coerce').astype('Int64')
p['FIMS'] = pd.to_numeric(p['FIMS'], errors='coerce').astype('Int64')
```

In [15]:
```
p.info()
p["Total"]=p["DV"]+p["M-II"]+p["PP"]+p["BEEE"]+p["FL"]+p["FIMS"]
p
```

```
<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       705 non-null    Int64  
 3   M-II     704 non-null    Int64  
 4   PP       701 non-null    Int64  
 5   BEEE     697 non-null    Int64  
 6   FL       702 non-null    Int64  
 7   FIMS     694 non-null    Int64  
dtypes: Int64(6), float64(1), object(1)
memory usage: 49.2+ KB
```

Out[15]:

	S.NO	SECTION	DV	M-II	PP	BEEE	FL	FIMS	Total
0	1.0	ALPHA	12	0	17	9	19	15	72
1	2.0	ALPHA	19	12	16	16	18	3	84
2	3.0	ALPHA	18	14	18	18	18	16	102
3	4.0	ALPHA	15	9	19	17	19	15	94
4	5.0	ALPHA	18	17	19	19	20	18	111
...	...	...	...	...	...	...	...	...	...
713	NaN	ZETA	19	8	8	19	17	18	89
714	NaN	ZETA	12	1	7	10	20	8	58
715	NaN	ZETA	17	6	14	14	17	18	86
716	NaN	ZETA	12	1	6	7	15	12	53
717	NaN	ZETA	19	14	17	16	20	19	105

718 rows × 9 columns

In [16]:
```
p.rename(columns={'Total':'TOTAL'},inplace=True)
p.rename(columns={'M-II':'M2'},inplace=True)
p
```

Out[16]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	TOTAL
0	1.0	ALPHA	12	0	17	9	19	15	72
1	2.0	ALPHA	19	12	16	16	18	3	84
2	3.0	ALPHA	18	14	18	18	18	16	102
3	4.0	ALPHA	15	9	19	17	19	15	94
4	5.0	ALPHA	18	17	19	19	20	18	111
...	...	...	...	...	...	...	...	...	...
713	NaN	ZETA	19	8	8	19	17	18	89
714	NaN	ZETA	12	1	7	10	20	8	58
715	NaN	ZETA	17	6	14	14	17	18	86
716	NaN	ZETA	12	1	6	7	15	12	53
717	NaN	ZETA	19	14	17	16	20	19	105

718 rows × 9 columns

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

In [17]:
```
p.fillna(0)
```

Out[17]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	TOTAL
0	1.0	ALPHA	12	0	17	9	19	15	72
1	2.0	ALPHA	19	12	16	16	18	3	84
2	3.0	ALPHA	18	14	18	18	18	16	102
3	4.0	ALPHA	15	9	19	17	19	15	94
4	5.0	ALPHA	18	17	19	19	20	18	111
...	...	...	...	...	...	...	...	...	...
713	0.0	ZETA	19	8	8	19	17	18	89
714	0.0	ZETA	12	1	7	10	20	8	58
715	0.0	ZETA	17	6	14	14	17	18	86
716	0.0	ZETA	12	1	6	7	15	12	53
717	0.0	ZETA	19	14	17	16	20	19	105

718 rows × 9 columns

In [18]:
```
p.head(10)
```

Out[18]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	TOTAL
0	1.0	ALPHA	12	0	17	9	19	15	72
1	2.0	ALPHA	19	12	16	16	18	3	84
2	3.0	ALPHA	18	14	18	18	18	16	102
3	4.0	ALPHA	15	9	19	17	19	15	94
4	5.0	ALPHA	18	17	19	19	20	18	111
5	6.0	ALPHA	17	16	18	10	15	9	85
6	7.0	ALPHA	15	10	20	20	15	14	94
7	8.0	ALPHA	17	17	19	20	19	13	105
8	9.0	ALPHA	10	18	<NA>	20	19	15	<NA>
9	10.0	ALPHA	18	19	20	20	20	15	112

In [19]:
```
p = p.fillna(-1)
```

In [20]:
```
p.head(10)
```

Out[20]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	TOTAL
0	1.0	ALPHA	12	0	17	9	19	15	72
1	2.0	ALPHA	19	12	16	16	18	3	84
2	3.0	ALPHA	18	14	18	18	18	16	102
3	4.0	ALPHA	15	9	19	17	19	15	94
4	5.0	ALPHA	18	17	19	19	20	18	111
5	6.0	ALPHA	17	16	18	10	15	9	85
6	7.0	ALPHA	15	10	20	20	15	14	94
7	8.0	ALPHA	17	17	19	20	19	13	105
8	9.0	ALPHA	10	18	-1	20	19	15	-1
9	10.0	ALPHA	18	19	20	20	20	15	112

In [21]:
```
p.loc[600:630]
```

Out[21]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	TOTAL
600	601.0	SIGMA	20	19	20	18	18	19	114
601	-1.0	-1	-1	-1	-1	-1	-1	-1	-1
602	-1.0	-1	20	19	16	15	13	18	101
603	-1.0	-1	12	3	10	6	10	19	60
604	-1.0	-1	20	9	15	17	15	20	96
605	-1.0	-1	11	18	19	15	19	16	98
606	-1.0	-1	20	20	17	16	18	19	110
607	-1.0	-1	20	19	18	15	18	20	110
608	-1.0	-1	16	20	17	16	18	18	105
609	-1.0	-1	15	13	15	7	18	17	85
610	-1.0	-1	10	10	11	2	10	5	48
611	-1.0	-1	2	0	0	3	10	9	24
612	-1.0	-1	18	8	9	8	13	16	72
613	-1.0	-1	20	20	19	20	20	20	119
614	-1.0	-1	20	19	17	18	20	19	113
615	-1.0	-1	20	20	17	16	18	18	109
616	-1.0	-1	15	18	9	-1	15	11	-1
617	-1.0	-1	12	12	15	14	13	11	77
618	-1.0	-1	20	20	20	20	20	19	119
619	-1.0	-1	14	6	17	14	10	18	79
620	-1.0	-1	19	8	19	13	18	20	97
621	-1.0	-1	20	20	20	16	18	17	111
622	-1.0	-1	16	3	11	17	15	16	78
623	-1.0	-1	18	20	17	18	15	20	108
624	-1.0	-1	18	15	12	13	18	16	92
625	-1.0	-1	19	3	18	15	15	17	87
626	-1.0	-1	17	20	16	19	18	18	108
627	-1.0	-1	15	1	11	8	13	17	65
628	-1.0	ZETA	17	3	8	16	15	-1	-1
629	-1.0	ZETA	16	1	6	16	14	12	65
630	-1.0	ZETA	15	20	14	19	20	17	105

In [22]:
```
p=p.drop([564])
```

In [23]:
```
p.loc[560:570]
```

Out[23]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	TOTAL
560	561.0	SIGMA	18	13	11	11	10	11	74
561	562.0	SIGMA	17	15	20	18	15	19	104
562	563.0	SIGMA	7	0	-1	5	15	16	-1
563	564.0	SIGMA	5	0	5	4	10	10	34
565	566.0	SIGMA	18	13	18	15	20	19	103
566	567.0	SIGMA	11	17	19	12	20	14	93
567	568.0	SIGMA	20	19	19	16	19	19	112
568	569.0	SIGMA	17	13	18	18	20	19	105
569	570.0	SIGMA	19	14	15	17	20	15	100
570	571.0	SIGMA	14	9	13	16	18	16	86

In [24]:
```
p.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 717 entries, 0 to 717
Data columns (total 9 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   S.NO     717 non-null    float64
 1   SECTION  717 non-null    object
 2   DV       717 non-null    Int64
 3   M2       717 non-null    Int64
 4   PP       717 non-null    Int64
 5   BEEE     717 non-null    Int64
 6   FL       717 non-null    Int64
 7   FIMS     717 non-null    Int64
 8   TOTAL    717 non-null    Int64
dtypes: Int64(7), float64(1), object(1)
memory usage: 77.1+ KB
```

In [25]:
```
p
```

Out[25]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	TOTAL
0	1.0	ALPHA	12	0	17	9	19	15	72
1	2.0	ALPHA	19	12	16	16	18	3	84
2	3.0	ALPHA	18	14	18	18	18	16	102
3	4.0	ALPHA	15	9	19	17	19	15	94
4	5.0	ALPHA	18	17	19	19	20	18	111
...	...	...	...	...	...	...	...	...	...
713	-1.0	ZETA	19	8	8	19	17	18	89
714	-1.0	ZETA	12	1	7	10	20	8	58
715	-1.0	ZETA	17	6	14	14	17	18	86
716	-1.0	ZETA	12	1	6	7	15	12	53
717	-1.0	ZETA	19	14	17	16	20	19	105

717 rows × 9 columns

In [26]:
```
p["percentage"]=(p["TOTAL"]/120)*100
```

In [27]:
```
p
```

Out[27]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	TOTAL	percentage
0	1.0	ALPHA	12	0	17	9	19	15	72	60.0
1	2.0	ALPHA	19	12	16	16	18	3	84	70.0
2	3.0	ALPHA	18	14	18	18	18	16	102	85.0
3	4.0	ALPHA	15	9	19	17	19	15	94	78.333333
4	5.0	ALPHA	18	17	19	19	20	18	111	92.5
...	...	...	...	...	...	...	...	...	...	...
713	-1.0	ZETA	19	8	8	19	17	18	89	74.166667
714	-1.0	ZETA	12	1	7	10	20	8	58	48.333333
715	-1.0	ZETA	17	6	14	14	17	18	86	71.666667
716	-1.0	ZETA	12	1	6	7	15	12	53	44.166667
717	-1.0	ZETA	19	14	17	16	20	19	105	87.5

717 rows × 10 columns

Calculating percentage based on 'Total' column values

In [28]:
```
p["grade"]=((p["TOTAL"]/120)*10).round()
```

In [29]:
```
p
```

Out[29]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	TOTAL	percentage	grade
0	1.0	ALPHA	12	0	17	9	19	15	72	60.0	6.0
1	2.0	ALPHA	19	12	16	16	18	3	84	70.0	7.0
2	3.0	ALPHA	18	14	18	18	18	16	102	85.0	8.0
3	4.0	ALPHA	15	9	19	17	19	15	94	78.333333	8.0
4	5.0	ALPHA	18	17	19	19	20	18	111	92.5	9.0
...	...	...	...	...	...	...	...	...	...	...	...
713	-1.0	ZETA	19	8	8	19	17	18	89	74.166667	7.0
714	-1.0	ZETA	12	1	7	10	20	8	58	48.333333	5.0
715	-1.0	ZETA	17	6	14	14	17	18	86	71.666667	7.0
716	-1.0	ZETA	12	1	6	7	15	12	53	44.166667	4.0
717	-1.0	ZETA	19	14	17	16	20	19	105	87.5	9.0

717 rows × 11 columns

In [30]:
```
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'

p['Grade'] = p['percentage'].apply(assign_grade)
p
```

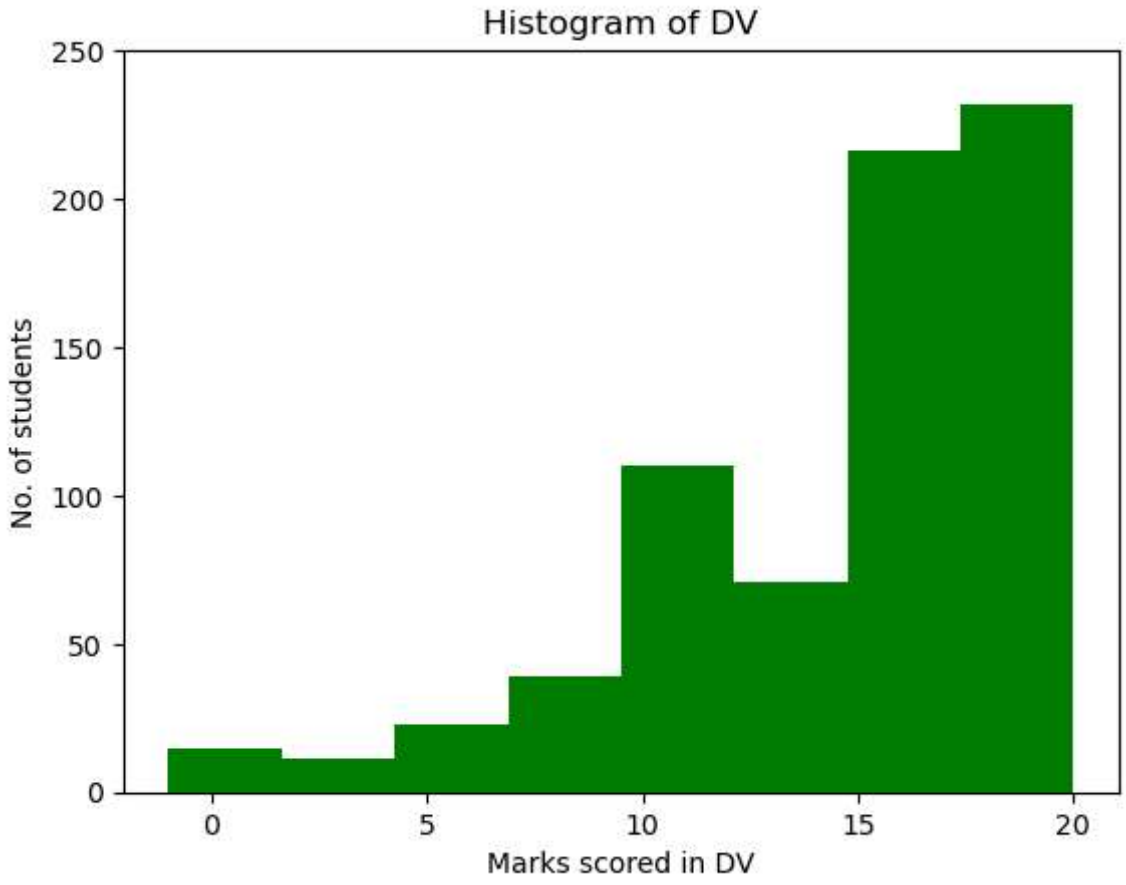
Out[30]:

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	TOTAL	percentage	grade	Grade
0	1.0	ALPHA	12	0	17	9	19	15	72	60.0	6.0	D
1	2.0	ALPHA	19	12	16	16	18	3	84	70.0	7.0	C
2	3.0	ALPHA	18	14	18	18	18	16	102	85.0	8.0	B
3	4.0	ALPHA	15	9	19	17	19	15	94	78.333333	8.0	C
4	5.0	ALPHA	18	17	19	19	20	18	111	92.5	9.0	A
...	...	...	...	...	...	...	...	...	...	...	...	...
713	-1.0	ZETA	19	8	8	19	17	18	89	74.166667	7.0	C
714	-1.0	ZETA	12	1	7	10	20	8	58	48.333333	5.0	F
715	-1.0	ZETA	17	6	14	14	17	18	86	71.666667	7.0	C
716	-1.0	ZETA	12	1	6	7	15	12	53	44.166667	4.0	F
717	-1.0	ZETA	19	14	17	16	20	19	105	87.5	9.0	B

717 rows × 12 columns

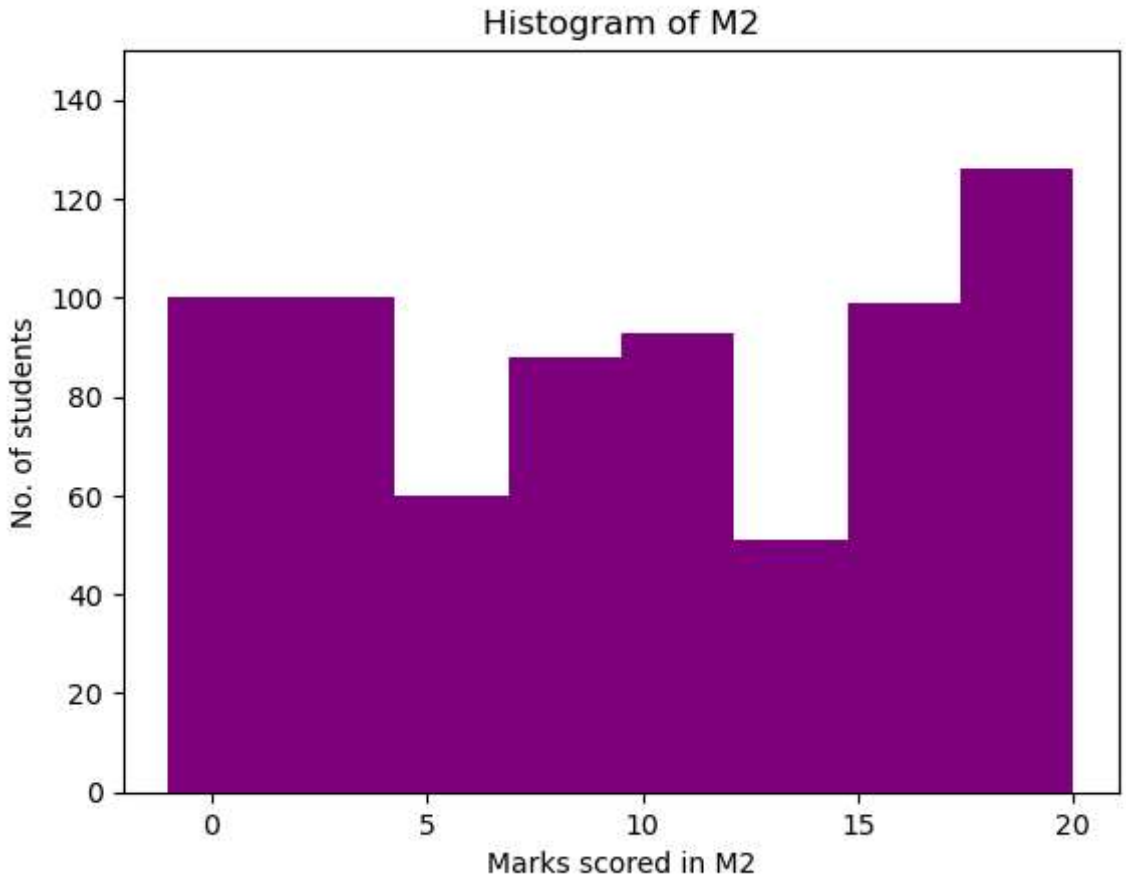
Assigning grades based on percentage values in dataframe

In [66]:
```
import matplotlib.pyplot as plt
plt.hist(p['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()
#print(p['DV'].value_counts())
```



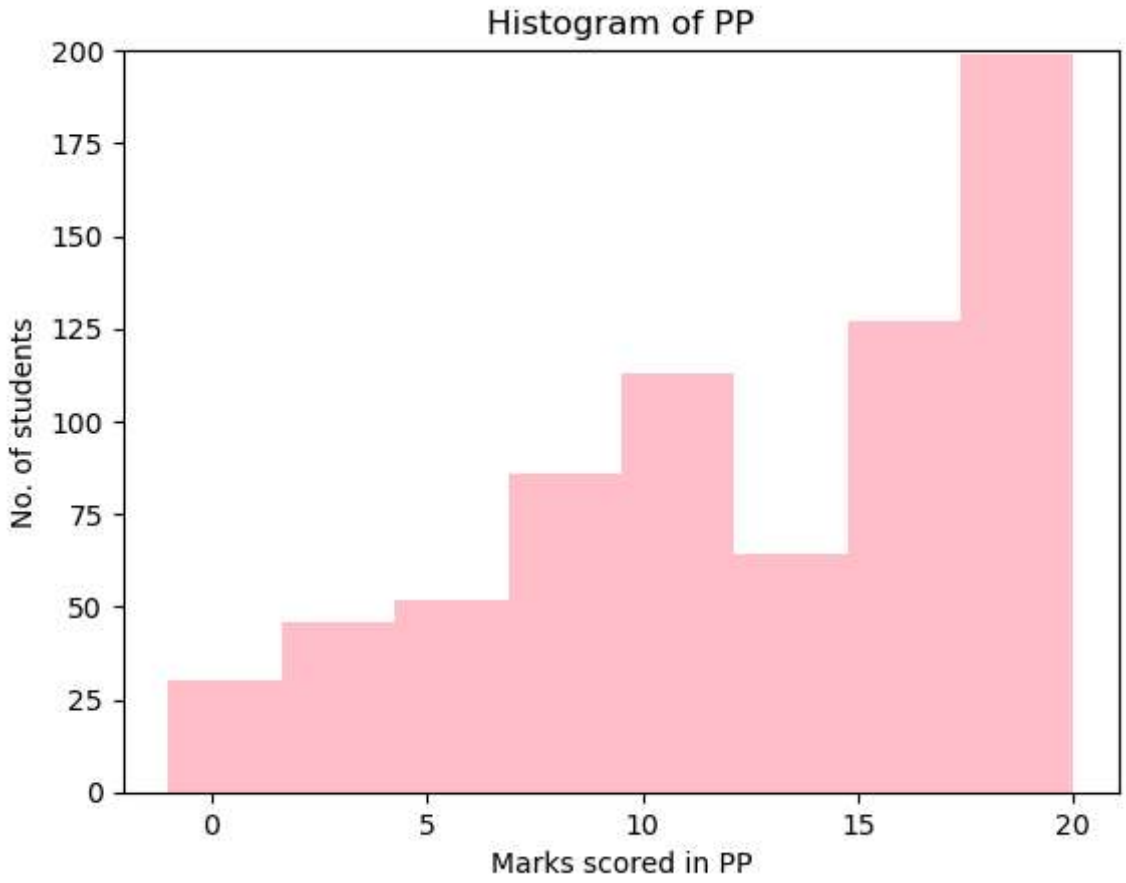
Creating histogram to visualize 'DV' subject marks distribution

In [67]:
```
plt.hist(p['M2'], color='purple', bins=8)
plt.ylim(0, 150)
plt.xlabel("Marks scored in M2")
plt.ylabel("No. of students")
plt.title("Histogram of M2")
plt.show()
#print(p['M2'].value_counts())
```



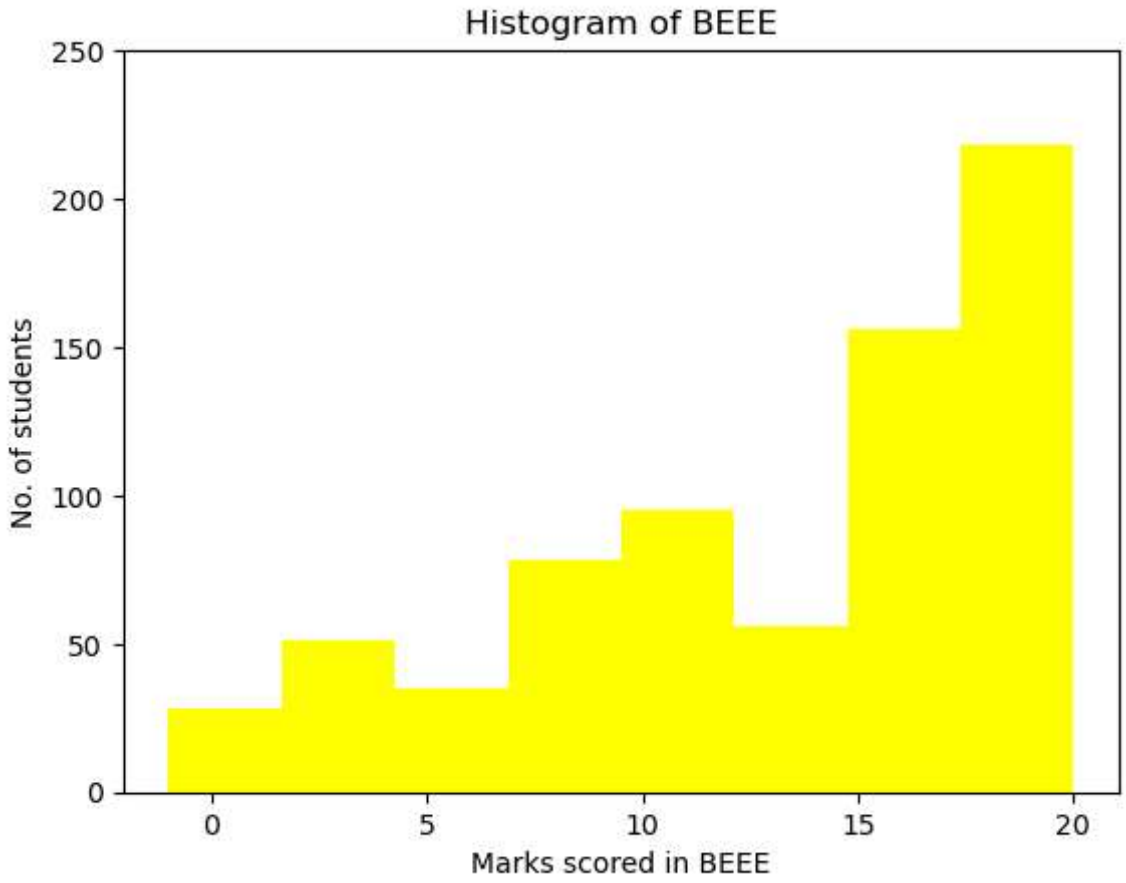
Creating histogram to visualize 'M2' subject marks distribution

In [65]:
```
plt.hist(p['PP'], color='pink', bins=8)
plt.ylim(0, 200)
plt.xlabel("Marks scored in PP")
plt.ylabel("No. of students")
plt.title("Histogram of PP")
plt.show()
```



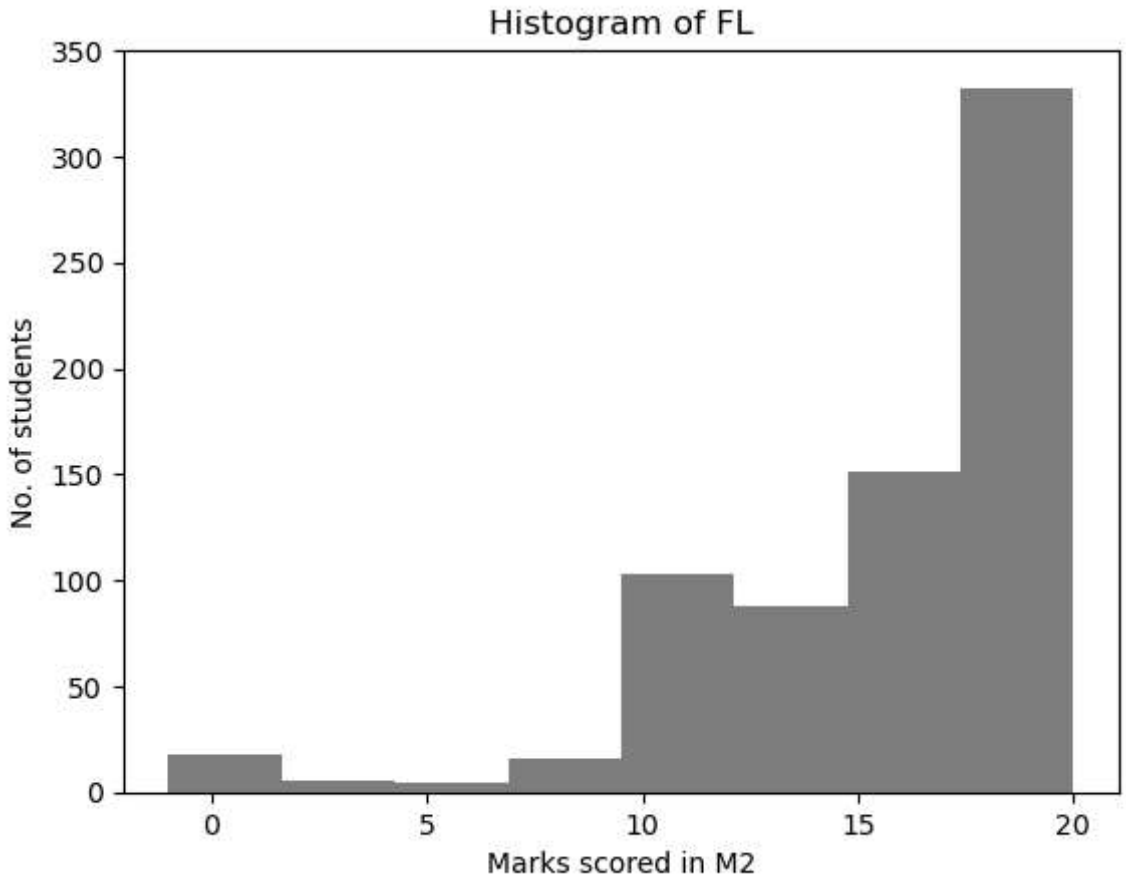
Creating histogram to visualize 'PP' subject marks distribution

In [34]:
```
plt.hist(p['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()
```



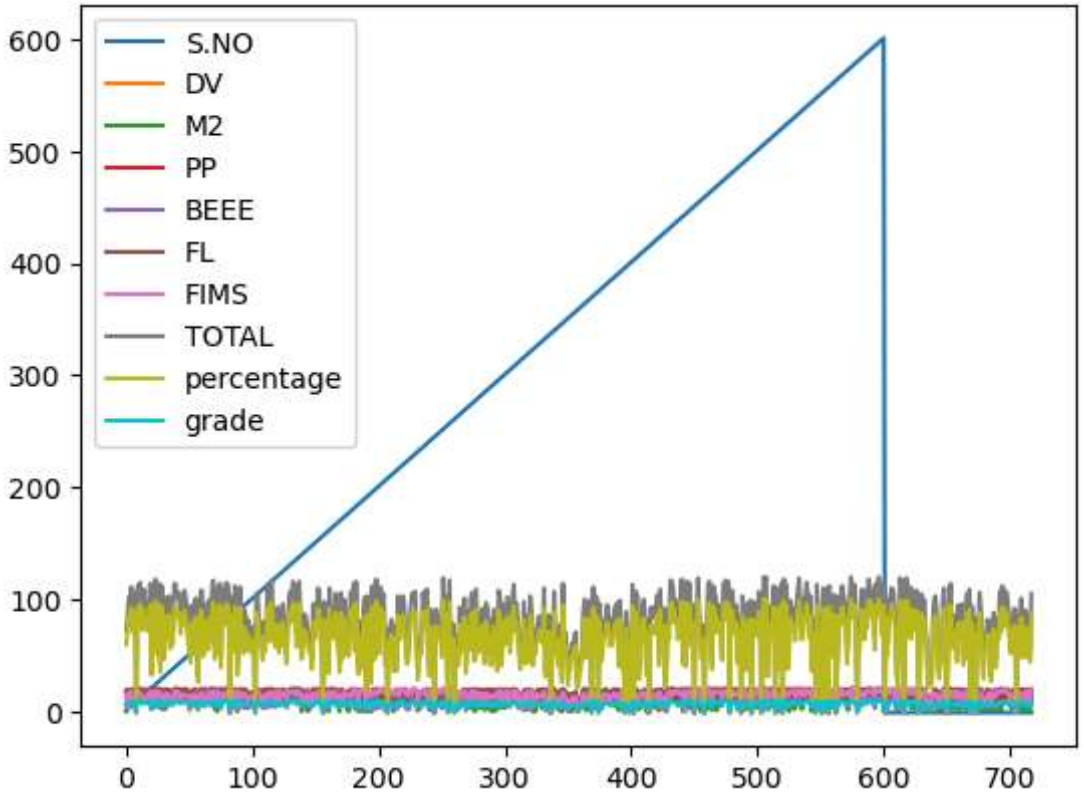
Creating histogram to visualize 'BEEE' subject marks distribution

In [64]:
```
plt.hist(p['FL'], color='gray', bins=8)
plt.ylim(0, 350)
plt.xlabel("Marks scored in M2")
plt.ylabel("No. of students")
plt.title("Histogram of FL")
plt.show()
```



Creating histogram to visualize 'FL' subject marks distribution

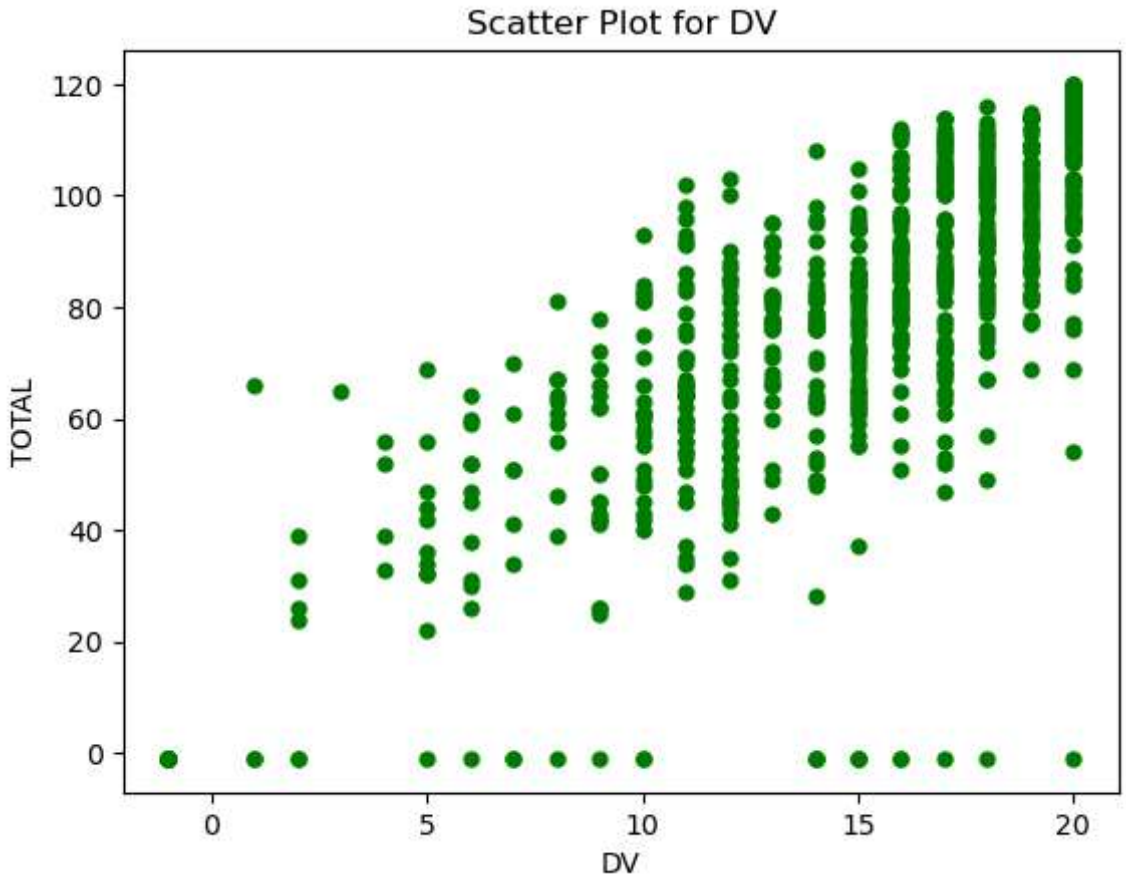
In [37]:
```
p.plot()
plt.show()
```



Plotting all columns in the dataframe for visualization

In [38]:
```
p.plot.scatter(x = 'DV', y = 'TOTAL',color='green',s=25)
plt.title("Scatter Plot for DV")
```

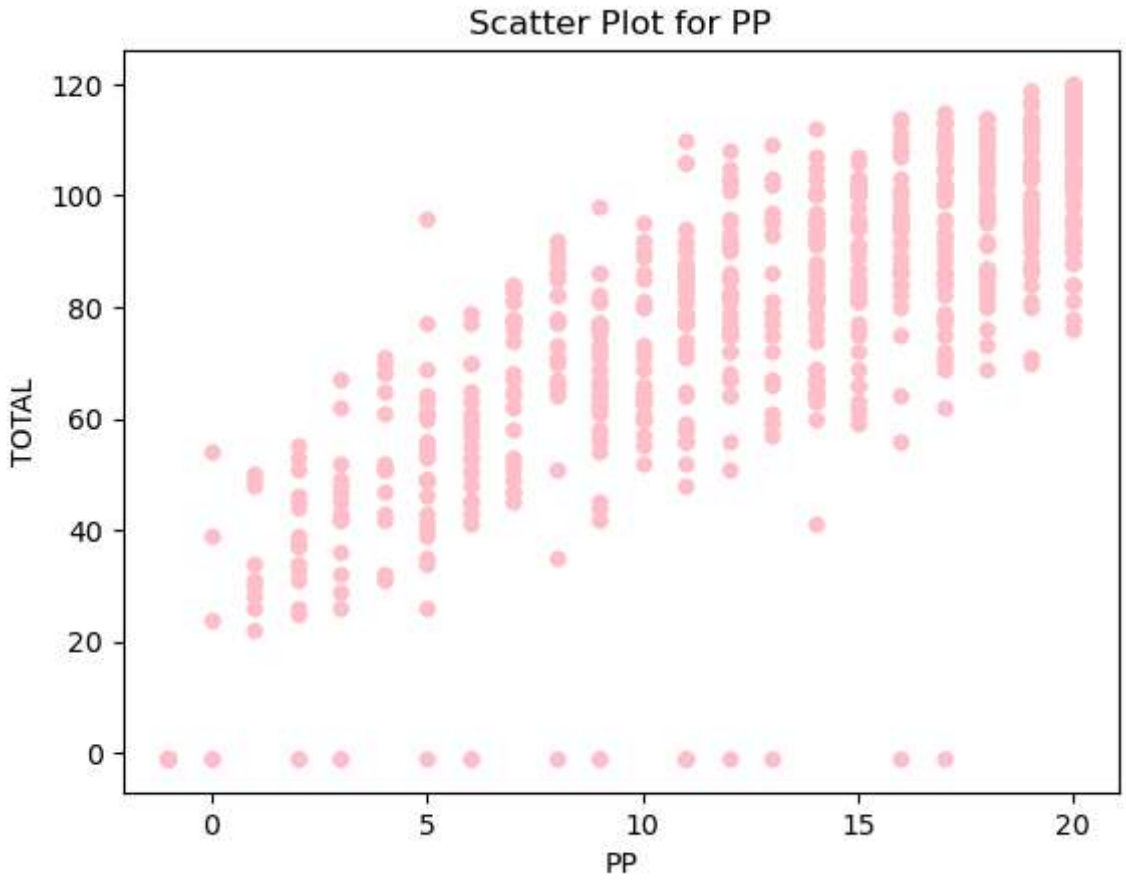
Out[38]:
```
Text(0.5, 1.0, 'Scatter Plot for DV')
```



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

In [62]:
```
p.plot.scatter(x = 'PP', y = 'TOTAL',color='pink',s=25)
plt.title("Scatter Plot for PP")
```

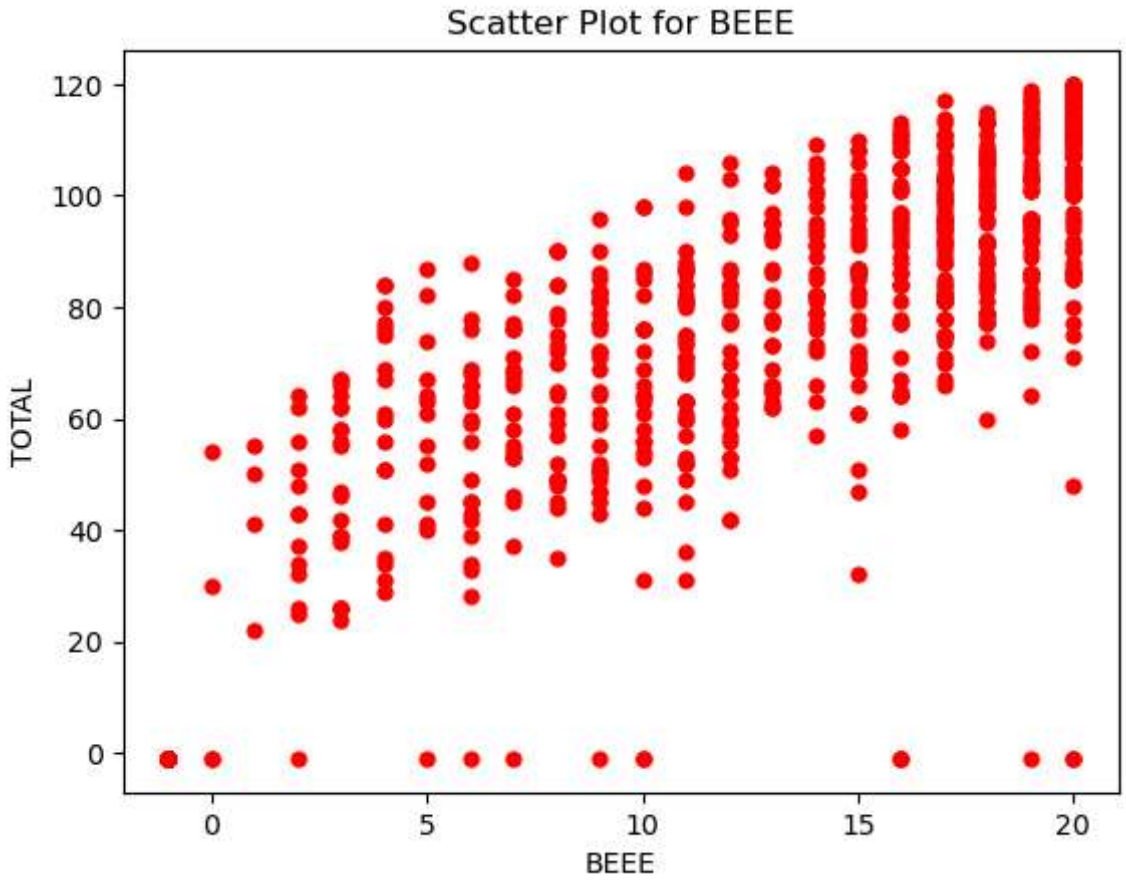
Out[62]:
```
Text(0.5, 1.0, 'Scatter Plot for PP')
```



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

In [61]:
```
p.plot.scatter(x = 'BEEE', y = 'TOTAL',color='red',s=25)
plt.title("Scatter Plot for BEEE")
```

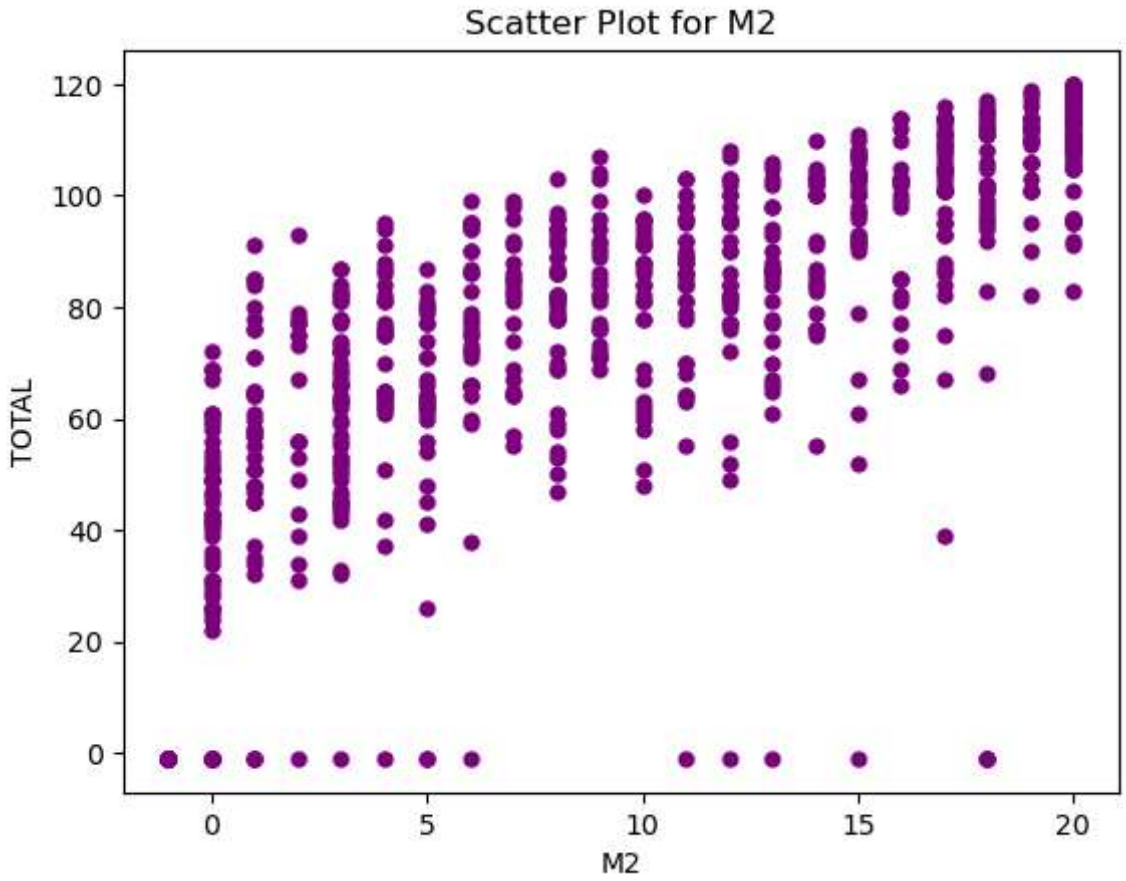
Out[61]:
```
Text(0.5, 1.0, 'Scatter Plot for BEEE')
```



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

In [60]:
```
p.plot.scatter(x = 'M2', y = 'TOTAL',color='purple',s=25)
plt.title("Scatter Plot for M2")
```

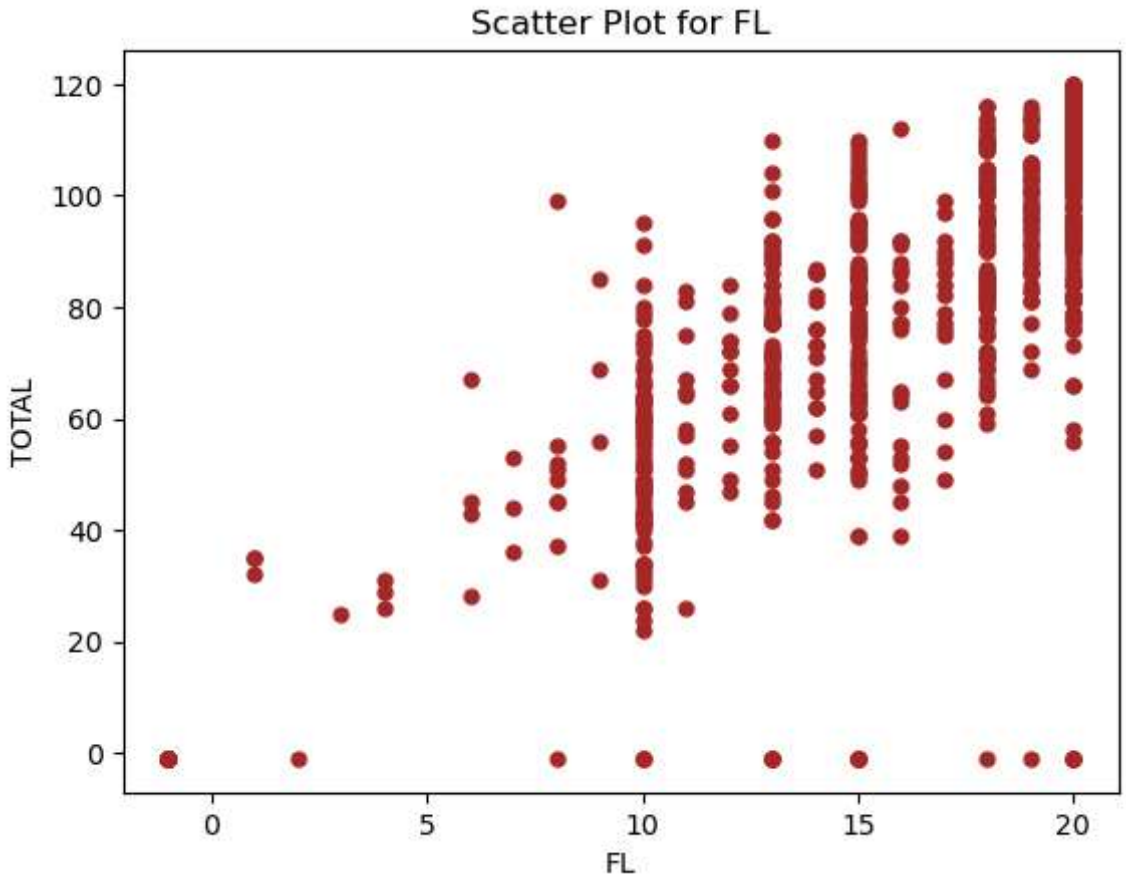
Out[60]:
```
Text(0.5, 1.0, 'Scatter Plot for M2')
```



Creating scatter plot for 'M2' vs 'Total' values

In [59]:
```
p.plot.scatter(x = 'FL', y = 'TOTAL',color='brown',s=25)
plt.title("Scatter Plot for FL")
```

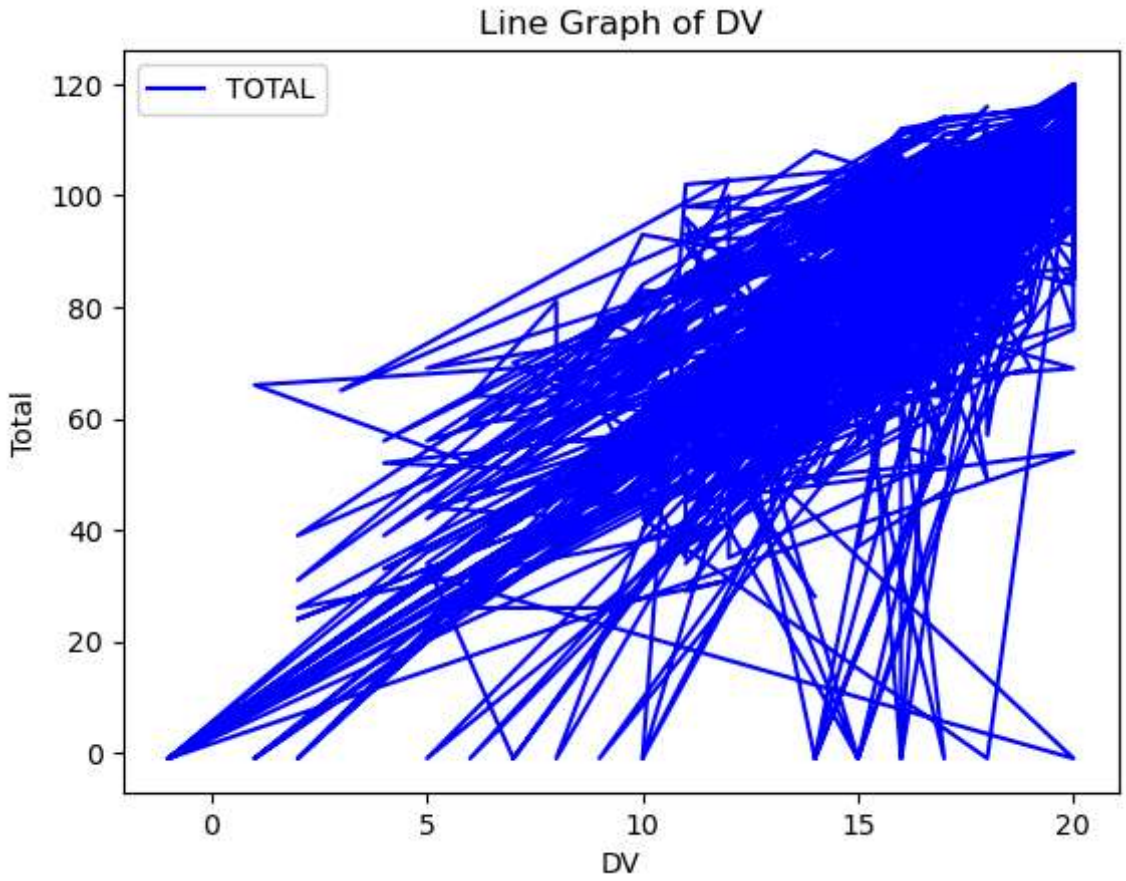
Out[59]:
```
Text(0.5, 1.0, 'Scatter Plot for FL')
```



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

In [58]:
```
p.plot.line(x='DV',y='TOTAL',color='blue')
plt.title("Line Graph of DV")
plt.ylabel("Total")
```

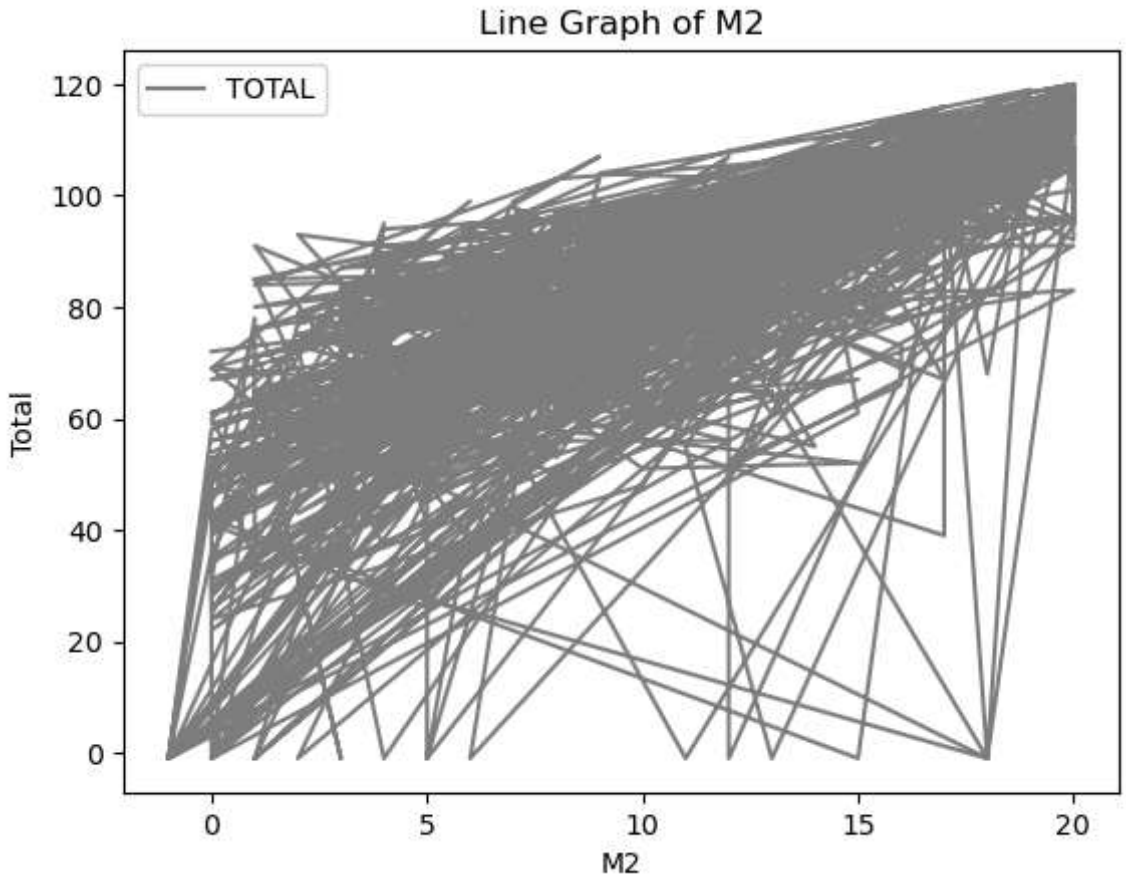
Out[58]:
```
Text(0, 0.5, 'Total')
```



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

In [57]:
```
p.plot.line(x='M2',y='TOTAL',color='gray')
plt.title("Line Graph of M2")
plt.ylabel("Total")
```

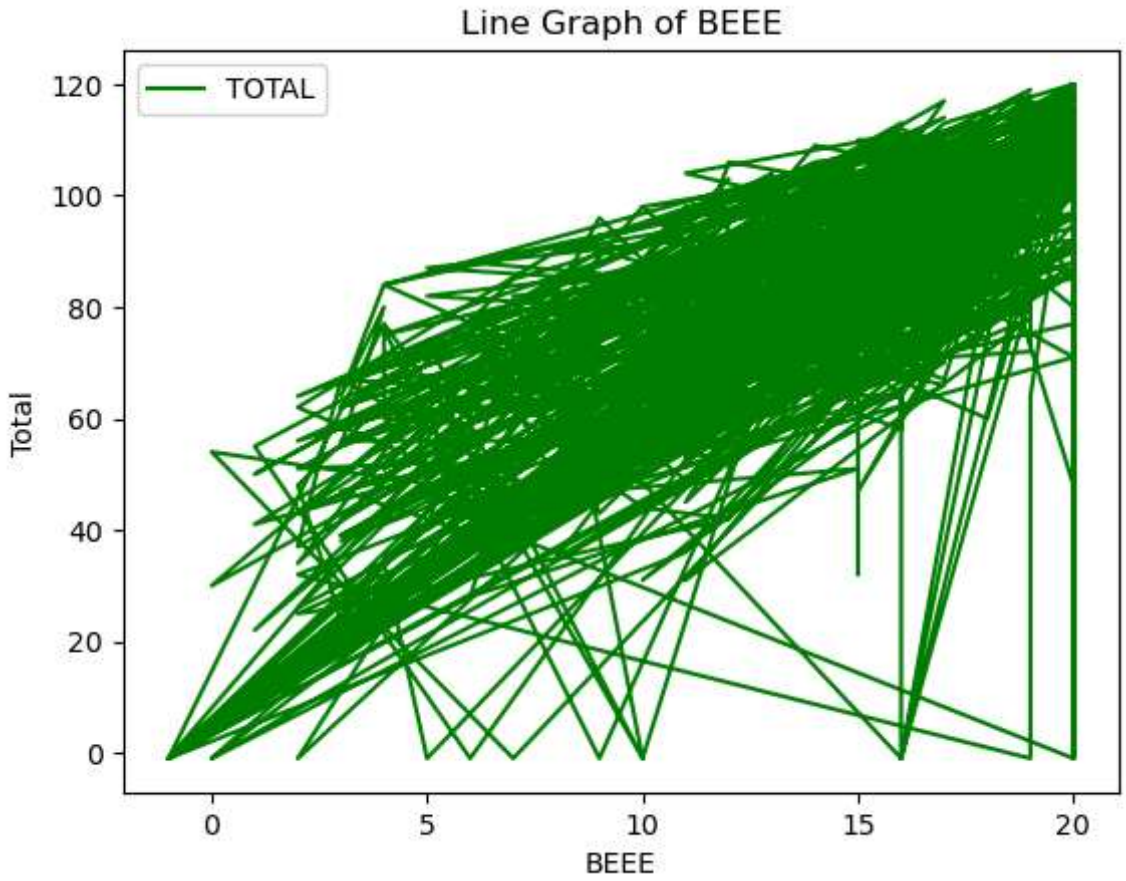
Out[57]:
```
Text(0, 0.5, 'Total')
```



Plotting line graph of 'M2' vs 'Total' values

In [56]:
```
p.plot.line(x='BEEE',y='TOTAL',color='green')
plt.title("Line Graph of BEEE")
plt.ylabel("Total")
```

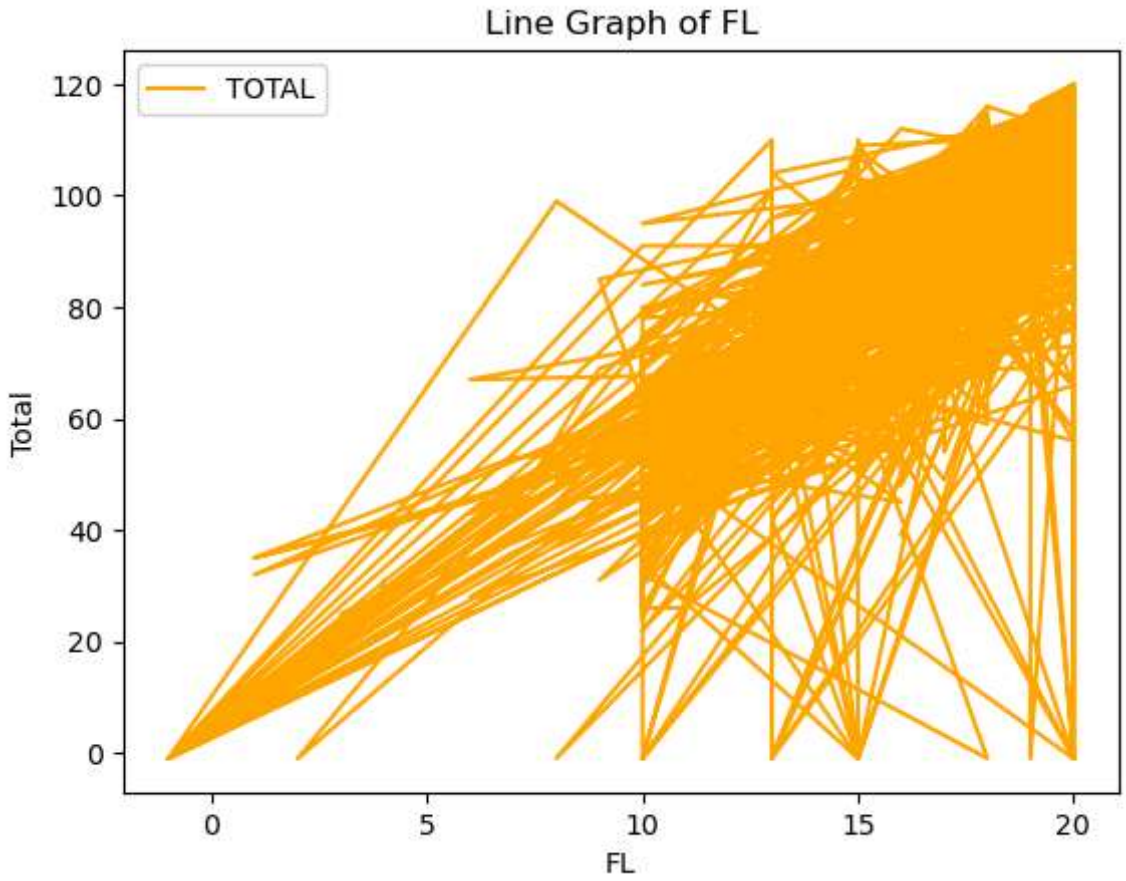
Out[56]:
```
Text(0, 0.5, 'Total')
```



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

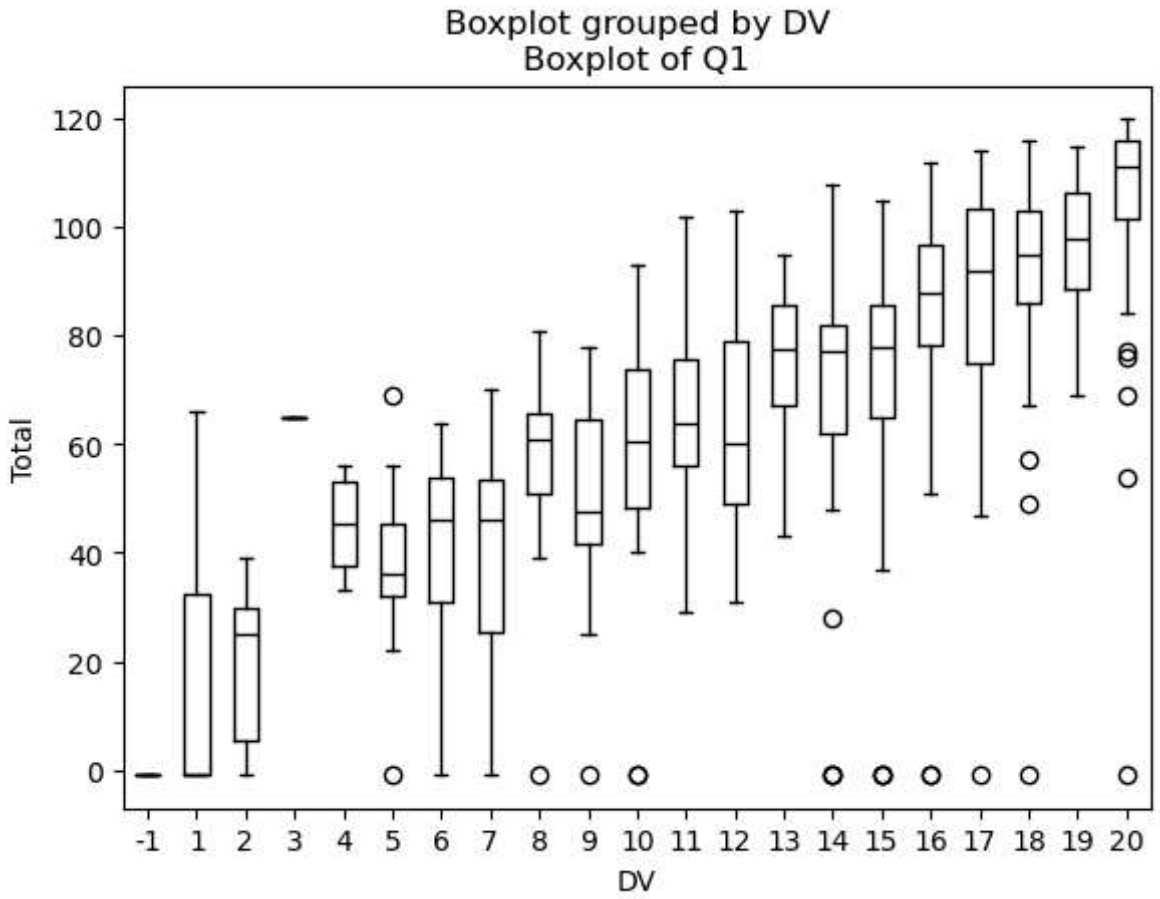
In [55]:
```
p.plot.line(x='FL',y='TOTAL',color='orange')
plt.title("Line Graph of FL")
plt.ylabel("Total")
```

Out[55]:
```
Text(0, 0.5, 'Total')
```



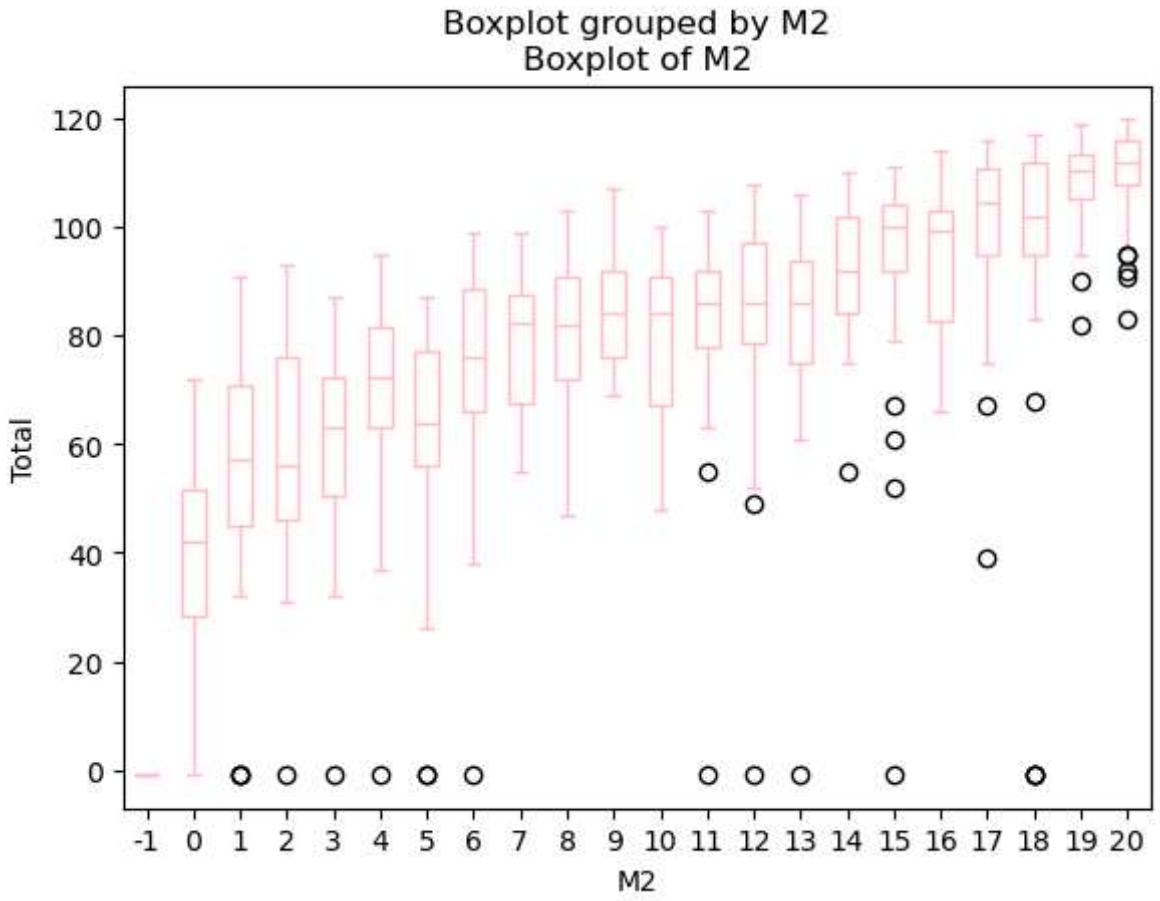
Plotting line graph of 'FL' vs 'Total' values

In [54]:
```
p.boxplot(by='DV', column =['TOTAL'], grid = False,color='black')
plt.title("Boxplot of Q1")
plt.ylabel("Total")
plt.show()
```



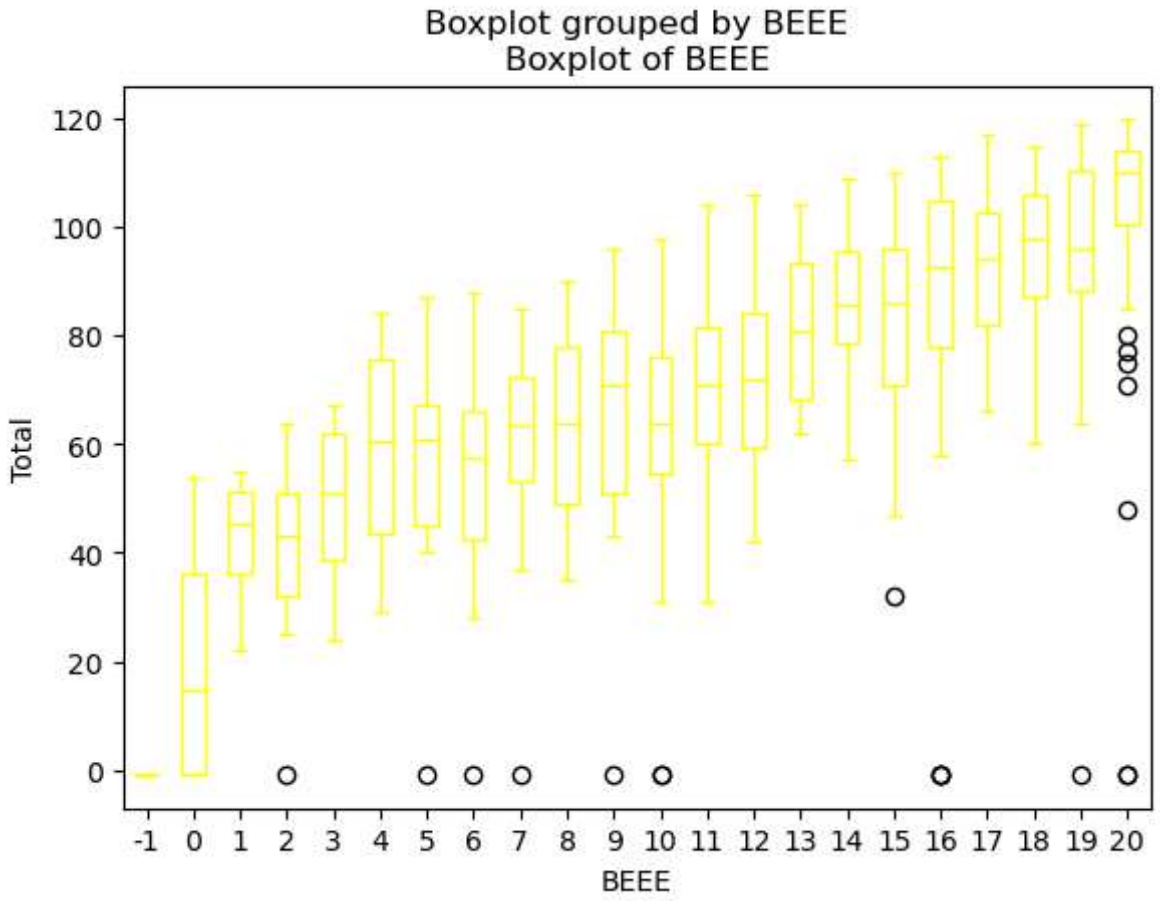
Creating boxplot to visualize 'Total' distribution by 'DV'

In [53]:
```
p.boxplot(by='M2', column =['TOTAL'], grid = False,color='pink')
plt.title("Boxplot of M2")
plt.ylabel("Total")
plt.show()
```



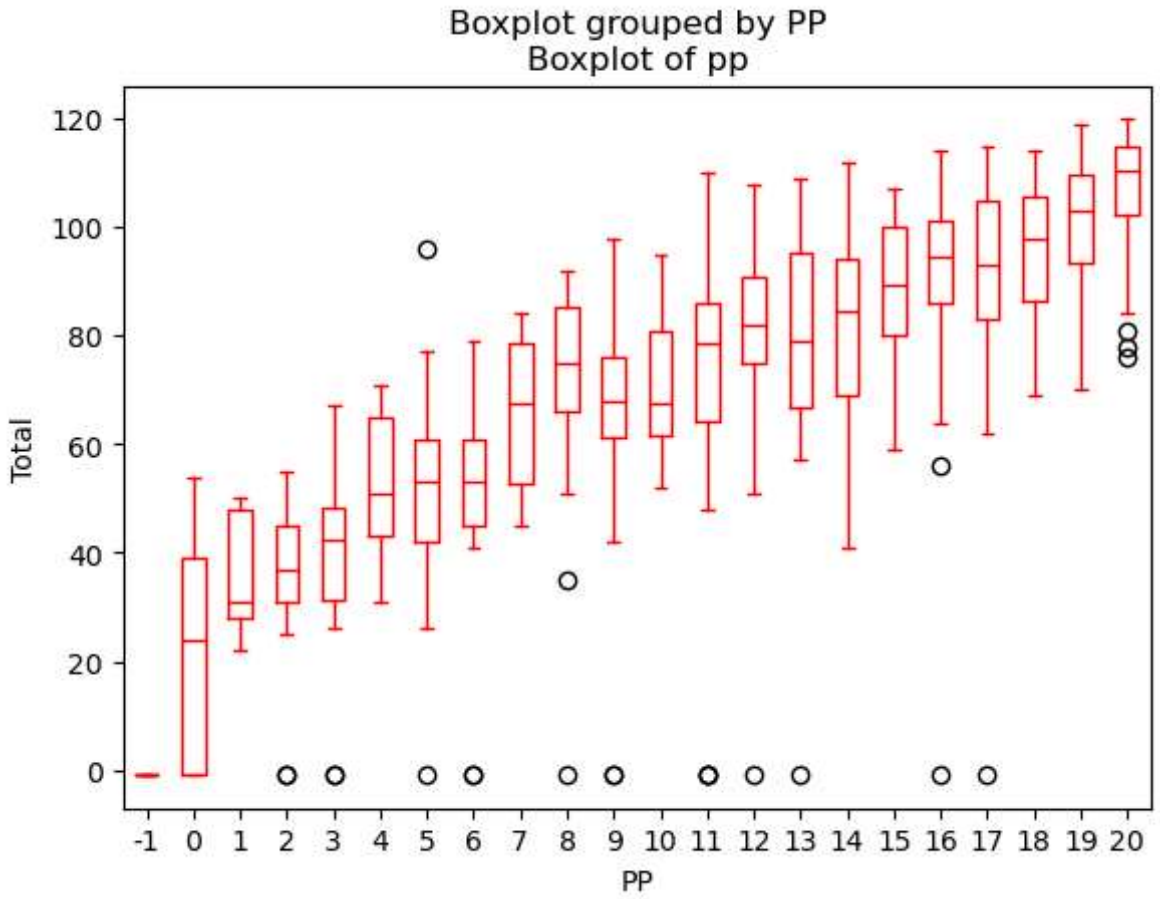
Creating boxplot to visualize 'Total' distribution by 'M2'

In [52]:
```
p.boxplot(by='BEEE', column =['TOTAL'], grid = False,color='yellow')
plt.title("Boxplot of BEEE")
plt.ylabel("Total")
plt.show()
```



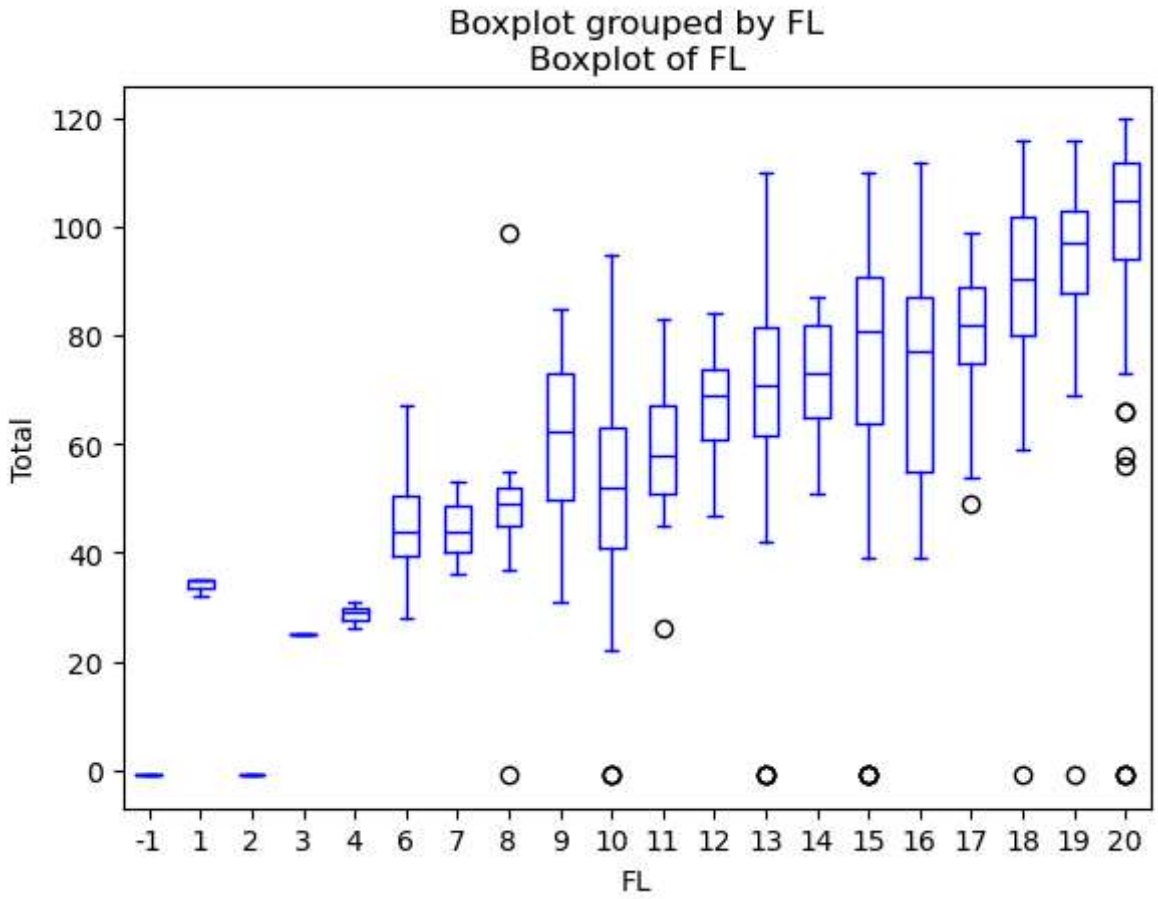
Creating boxplot to visualize 'Total' distribution by 'BEEE'

In [50]:
```
p.boxplot(by='PP', column =['TOTAL'], grid = False,color='red')
plt.title("Boxplot of pp")
plt.ylabel("Total")
plt.show()
```



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

In [51]:
```
p.boxplot(by='FL', column =['TOTAL'], grid = False,color='blue')
plt.title("Boxplot of FL")
plt.ylabel("Total")
plt.show()
```



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

Thank You