```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_excel("MIDMARKS.xlsx")
df
```

	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 x 8 columns]

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

df
```

	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 x 8 columns]

```
df.head()
```

	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

```
df = df.fillna(0)
df
```

```
     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   0.0    ZETA  19   8   8   19  17   18
714   0.0    ZETA  12   1   7   10  20    8
715   0.0    ZETA  17   6  14   14  17   18
716   0.0    ZETA  12   1   6    7  15   12
717   0.0    ZETA  19  14  17   16  20   19

[718 rows x 8 columns]
```

```
df[df.M2=='AB']
```

```
      S.NO SECTION  DV  M2  PP BEEE  FL FIMS
207  208.0   DELTA  MP  AB  16   10  20   19
244  245.0   DELTA   A  AB  AB    A   A   AB
395  396.0   GAMMA   A  AB   6    7  13    A
402  403.0   GAMMA   A  AB  AB    A   A    A
414  415.0   GAMMA   A  AB  AB    A   A    A
551  552.0   SIGMA   A  AB   A    A   A   AB
556  557.0   SIGMA   A  AB   A    A   A   AB
```

```
df.replace('AB',0)
```

```
     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   0.0    ZETA  19   8   8   19  17   18
714   0.0    ZETA  12   1   7   10  20    8
715   0.0    ZETA  17   6  14   14  17   18
716   0.0    ZETA  12   1   6    7  15   12
717   0.0    ZETA  19  14  17   16  20   19

[718 rows x 8 columns]
```

```
df.DV.value_counts()
```

```
DV
20    103
17     79
```

```
16     74
18     69
15     63
19     60
11     43
14     41
12     41
13     30
10     26
9      20
6      12
5      11
8      11
A      10
7       8
2       6
4       4
1       3
0       2
3       1
MP      1
Name: count, dtype: int64
```

```
df[df.PP==0]
```

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS
88	89.0	ALPHA	2	17	0	3	15	2
394	395.0	GAMMA	20	8	0	0	13	13
487	488.0	OMEGA	1	5	0	A	A	AB
564	565.0	SIGMA	0	0	0	0	0	0
601	0.0	0	0	0	0	0	0	0
611	0.0	0	2	0	0	3	10	9
673	0.0	ZETA	2	A	0	0	2	1

```
df[df.M2==0]
```

	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS
0	1.0	ALPHA	12	0	17	9	19	15
72	73.0	ALPHA	7	0	15	10	18	11
82	83.0	ALPHA	2	0	2	A	A	A
139	140.0	BETA	5	0	12	4	20	15
144	145.0	BETA	20	0	17	13	10	9
160	161.0	BETA	7	0	14	5	10	5
169	170.0	BETA	9	0	9	3	10	11
176	177.0	BETA	6	0	13	8	18	14
184	185.0	DELTA	10	0	5	5	10	10
187	188.0	DELTA	8	0	5	3	10	20
190	191.0	DELTA	5	0	1	1	10	5
192	193.0	DELTA	12	0	5	6	10	16
194	195.0	DELTA	6	0	1	0	10	13

212	213.0	DELTA	13	0	3	2	10	15
243	244.0	DELTA	10	0	11	7	15	15
260	261.0	DELTA	8	0	3	A	10	14
276	277.0	EPSILON	11	0	5	7	17	14
289	290.0	EPSILON	14	0	5	11	8	11
292	293.0	EPSILON	17	0	8	17	11	14
322	323.0	EPSILON	18	0	3	8	17	3
324	325.0	EPSILON	15	0	5	15	13	13
337	338.0	EPSILON	5	0	3	11	7	10
345	346.0	EPSILON	6	0	1	11	9	4
350	351.0	EPSILON	14	0	1	6	6	1
351	352.0	EPSILON	12	0	4	15	11	9
352	353.0	EPSILON	11	0	2	6	16	10
357	358.0	EPSILON	9	0	2	3	11	1
358	359.0	GAMMA	6	0	3	3	10	4
368	369.0	GAMMA	10	0	4	6	13	9
374	375.0	GAMMA	7	0	3	A	13	8
412	413.0	GAMMA	8	0	2	6	15	8
428	429.0	GAMMA	12	0	6	4	10	9
441	442.0	GAMMA	10	0	4	9	15	13
450	451.0	OMEGA	9	0	A	A	A	AB
455	456.0	OMEGA	16	0	14	6	18	15
517	518.0	OMEGA	12	0	7	8	12	10
525	526.0	OMEGA	11	0	5	11	17	16
543	544.0	SIGMA	10	0	5	9	10	9
562	563.0	SIGMA	7	0	A	5	15	16
563	564.0	SIGMA	5	0	5	4	10	10
564	565.0	SIGMA	0	0	0	0	0	0
601	0.0	0	0	0	0	0	0	0
611	0.0	0	2	0	0	3	10	9
635	0.0	ZETA	9	0	2	2	3	9
636	0.0	ZETA	9	0	5	3	4	5
637	0.0	ZETA	12	0	2	4	4	9
638	0.0	ZETA	11	0	3	4	4	7
645	0.0	ZETA	17	0	7	12	7	10
649	0.0	ZETA	11	0	7	9	12	8
651	0.0	ZETA	14	0	11	8	8	11
679	0.0	ZETA	11	0	8	4	1	11
705	0.0	ZETA	16	0	11	16	20	A

```
df['DV'] = pd.to_numeric(df['DV'], errors='coerce')
df['M2'] = pd.to_numeric(df['M2'], errors='coerce')
df['PP'] = pd.to_numeric(df['PP'], errors='coerce')
df['BEEE'] = pd.to_numeric(df['BEEE'], errors='coerce')
df['FL'] = pd.to_numeric(df['FL'], errors='coerce')
df['FIMS'] = pd.to_numeric(df['FIMS'], errors='coerce')
df.fillna(0, inplace=True)

df
```

```
     S.NO SECTION    DV    M2    PP  BEEE    FL  FIMS
0     1.0   ALPHA  12.0   0.0  17.0   9.0  19.0  15.0
1     2.0   ALPHA  19.0  12.0  16.0  16.0  18.0   3.0
2     3.0   ALPHA  18.0  14.0  18.0  18.0  18.0  16.0
3     4.0   ALPHA  15.0   9.0  19.0  17.0  19.0  15.0
4     5.0   ALPHA  18.0  17.0  19.0  19.0  20.0  18.0
..    ...     ...   ...   ...   ...   ...   ...   ...
713   0.0    ZETA  19.0   8.0   8.0  19.0  17.0  18.0
714   0.0    ZETA  12.0   1.0   7.0  10.0  20.0   8.0
715   0.0    ZETA  17.0   6.0  14.0  14.0  17.0  18.0
716   0.0    ZETA  12.0   1.0   6.0   7.0  15.0  12.0
717   0.0    ZETA  19.0  14.0  17.0  16.0  20.0  19.0

[718 rows x 8 columns]
```

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

```
     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   0.0    ZETA  19.0   8.0   8.0  19.0  17.0  18.0   89.0
714   0.0    ZETA  12.0   1.0   7.0  10.0  20.0   8.0   58.0
715   0.0    ZETA  17.0   6.0  14.0  14.0  17.0  18.0   86.0
716   0.0    ZETA  12.0   1.0   6.0   7.0  15.0  12.0   53.0
717   0.0    ZETA  19.0  14.0  17.0  16.0  20.0  19.0  105.0

[718 rows x 9 columns]
```

```
df["Percentage"] = (df['Total']/120)*100

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   0.0    ZETA  19.0   8.0   8.0  19.0  17.0  18.0   89.0
74.166667
714   0.0    ZETA  12.0   1.0   7.0  10.0  20.0   8.0   58.0
48.333333
715   0.0    ZETA  17.0   6.0  14.0  14.0  17.0  18.0   86.0
71.666667
716   0.0    ZETA  12.0   1.0   6.0   7.0  15.0  12.0   53.0
44.166667
717   0.0    ZETA  19.0  14.0  17.0  16.0  20.0  19.0  105.0
87.500000

[718 rows x 10 columns]
```

```
df['Percentage'] = df['Percentage'].round().astype(int)
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
1     2.0   ALPHA  19.0  12.0  16.0  16.0  18.0   3.0   84.0
70
2     3.0   ALPHA  18.0  14.0  18.0  18.0  18.0  16.0  102.0
85
3     4.0   ALPHA  15.0   9.0  19.0  17.0  19.0  15.0   94.0
78
4     5.0   ALPHA  18.0  17.0  19.0  19.0  20.0  18.0  111.0
92
..    ...     ...   ...   ...   ...   ...   ...   ...    ...          .
..
713   0.0    ZETA  19.0   8.0   8.0  19.0  17.0  18.0   89.0
74
714   0.0    ZETA  12.0   1.0   7.0  10.0  20.0   8.0   58.0
48
715   0.0    ZETA  17.0   6.0  14.0  14.0  17.0  18.0   86.0
72
716   0.0    ZETA  12.0   1.0   6.0   7.0  15.0  12.0   53.0
44
717   0.0    ZETA  19.0  14.0  17.0  16.0  20.0  19.0  105.0
88

[718 rows x 10 columns]
```

```
def assign_grade(percentage):
    if percentage >= 90:
        return 'A'
    elif percentage >= 80:
        return 'B+'
    elif percentage >= 70:
```

```
        return 'B'
    elif percentage >= 60:
        return 'C+'
    elif percentage >=50:
        return 'C'
    elif percentage >=40:
        return 'D'
    else:
        return 'F'
df['Grade'] = df['Percentage'].apply(assign_grade)
df
```

```
     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    C+
1     2.0   ALPHA  19.0  12.0  16.0  16.0  18.0   3.0   84.0
70     B
2     3.0   ALPHA  18.0  14.0  18.0  18.0  18.0  16.0  102.0
85    B+
3     4.0   ALPHA  15.0   9.0  19.0  17.0  19.0  15.0   94.0
78     B
4     5.0   ALPHA  18.0  17.0  19.0  19.0  20.0  18.0  111.0
92     A
..    ...     ...   ...   ...   ...   ...   ...   ...    ...          .
..   ...
713   0.0    ZETA  19.0   8.0   8.0  19.0  17.0  18.0   89.0
74     B
714   0.0    ZETA  12.0   1.0   7.0  10.0  20.0   8.0   58.0
48     D
715   0.0    ZETA  17.0   6.0  14.0  14.0  17.0  18.0   86.0
72     B
716   0.0    ZETA  12.0   1.0   6.0   7.0  15.0  12.0   53.0
44     D
717   0.0    ZETA  19.0  14.0  17.0  16.0  20.0  19.0  105.0
88    B+

[718 rows x 11 columns]
```

```
df.sort_values('Total').tail(10)
```

```
      S.NO SECTION    DV    M2    PP  BEEE    FL  FIMS  Total
Percentage  \
521  522.0   OMEGA  20.0  19.0  20.0  19.0  20.0  20.0  118.0
98
533  534.0   OMEGA  20.0  19.0  20.0  20.0  20.0  20.0  119.0
99
613    0.0       0  20.0  20.0  19.0  20.0  20.0  20.0  119.0
99
453  454.0   OMEGA  20.0  20.0  20.0  20.0  20.0  19.0  119.0
```

```
99
251  252.0   DELTA  20.0  20.0  20.0  19.0  20.0  20.0  119.0
99
618    0.0       0  20.0  20.0  20.0  20.0  20.0  19.0  119.0
99
574  575.0   SIGMA  20.0  20.0  20.0  20.0  20.0  20.0  120.0
100
507  508.0   OMEGA  20.0  20.0  20.0  20.0  20.0  20.0  120.0
100
505  506.0   OMEGA  20.0  20.0  20.0  20.0  20.0  20.0  120.0
100
596  597.0   SIGMA  20.0  20.0  20.0  20.0  20.0  20.0  120.0
100

    Grade
521     A
533     A
613     A
453     A
251     A
618     A
574     A
507     A
505     A
596     A
```

```
df.sort_values('DV').tail(20)
```

```
      S.NO SECTION    DV    M2    PP  BEEE    FL  FIMS  Total
Percentage  \
502  503.0   OMEGA  20.0   7.0  17.0  17.0  20.0  18.0   99.0
82
505  506.0   OMEGA  20.0  20.0  20.0  20.0  20.0  20.0  120.0
100
474  475.0   OMEGA  20.0  18.0  20.0  19.0  20.0  18.0  115.0
96
506  507.0   OMEGA  20.0  18.0  20.0  20.0  20.0  19.0  117.0
98
510  511.0   OMEGA  20.0   6.0  16.0  11.0  20.0  14.0   87.0
72
511  512.0   OMEGA  20.0   6.0  18.0  13.0  20.0  18.0   95.0
79
512  513.0   OMEGA  20.0  20.0   5.0  19.0  18.0  14.0   96.0
80
513  514.0   OMEGA  20.0  16.0  18.0  19.0  20.0  19.0  112.0
93
515  516.0   OMEGA  20.0  15.0  20.0  20.0  20.0  15.0  110.0
92
621    0.0       0  20.0  20.0  20.0  16.0  18.0  17.0  111.0
92
```

```
518  519.0   OMEGA  20.0   1.0  20.0   6.0  14.0  15.0   76.0
63
519  520.0   OMEGA  20.0  17.0  20.0  20.0  20.0  17.0  114.0
95
618    0.0       0  20.0  20.0  20.0  20.0  20.0  19.0  119.0
99
521  522.0   OMEGA  20.0  19.0  20.0  19.0  20.0  20.0  118.0
98
522  523.0   OMEGA  20.0  14.0  19.0  19.0  20.0  18.0  110.0
92
615    0.0       0  20.0  20.0  17.0  16.0  18.0  18.0  109.0
91
614    0.0       0  20.0  19.0  17.0  18.0  20.0  19.0  113.0
94
613    0.0       0  20.0  20.0  19.0  20.0  20.0  20.0  119.0
99
507  508.0   OMEGA  20.0  20.0  20.0  20.0  20.0  20.0  120.0
100
69    70.0   ALPHA  20.0  20.0  20.0  19.0  20.0  18.0  117.0
98

    Grade
502    B+
505     A
474     A
506     A
510     B
511     B
512    B+
513     A
515     A
621     A
518    C+
519     A
618     A
521     A
522     A
615     A
614     A
613     A
507     A
69      A

df.sort_values('DV').head(50)

      S.NO  SECTION   DV    M2    PP  BEEE    FL  FIMS  Total
Percentage  \
601    0.0       0  0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
514  515.0    OMEGA  0.0   0.0  12.0  16.0  20.0  18.0   66.0
```

```
55
414  415.0    GAMMA  0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
551  552.0    SIGMA  0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
556  557.0    SIGMA  0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
564  565.0    SIGMA  0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
402  403.0    GAMMA  0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
395  396.0    GAMMA  0.0   0.0   6.0   7.0  13.0   0.0   26.0
22
336  337.0  EPSILON  0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
244  245.0    DELTA  0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
650    0.0     ZETA  0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
495  496.0    OMEGA  0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
207  208.0    DELTA  0.0   0.0  16.0  10.0  20.0  19.0   65.0
54
487  488.0    OMEGA  1.0   5.0   0.0   0.0   0.0   0.0    6.0
5
503  504.0    OMEGA  1.0   1.0   2.0   0.0  10.0   0.0   14.0
12
50    51.0    ALPHA  1.0  16.0  15.0  13.0  10.0  11.0   66.0
55
673    0.0     ZETA  2.0   0.0   0.0   0.0   2.0   1.0    5.0
4
427  428.0    GAMMA  2.0   5.0   1.0   2.0  10.0   6.0   26.0
22
57    58.0    ALPHA  2.0   2.0   4.0  10.0  10.0   3.0   31.0
26
88    89.0    ALPHA  2.0  17.0   0.0   3.0  15.0   2.0   39.0
32
611    0.0        0  2.0   0.0   0.0   3.0  10.0   9.0   24.0
20
82    83.0    ALPHA  2.0   0.0   2.0   0.0   0.0   0.0    4.0
3
85    86.0    ALPHA  3.0   4.0  14.0  13.0  18.0  13.0   65.0
54
223  224.0    DELTA  4.0  15.0   4.0   5.0  10.0  14.0   52.0
43
70    71.0    ALPHA  4.0   2.0  16.0  10.0  15.0   9.0   56.0
47
549  550.0    SIGMA  4.0   3.0   2.0   6.0  10.0   8.0   33.0
28
```

```
20    21.0    ALPHA  4.0   2.0   5.0   3.0  16.0   9.0   39.0
32
563  564.0    SIGMA  5.0   0.0   5.0   4.0  10.0  10.0   34.0
28
398  399.0    GAMMA  5.0   3.0   3.0   2.0  10.0   9.0   32.0
27
27    28.0    ALPHA  5.0   4.0   3.0  12.0  13.0   5.0   42.0
35
75    76.0    ALPHA  5.0   8.0   7.0  15.0  10.0   2.0   47.0
39
360  361.0    GAMMA  5.0   3.0   9.0  10.0  10.0   7.0   44.0
37
190  191.0    DELTA  5.0   0.0   1.0   1.0  10.0   5.0   22.0
18
125  126.0     BETA  5.0  16.0   9.0   7.0  18.0  14.0   69.0
57
337  338.0  EPSILON  5.0   0.0   3.0  11.0   7.0  10.0   36.0
30
139  140.0     BETA  5.0   0.0  12.0   4.0  20.0  15.0   56.0
47
478  479.0    OMEGA  5.0   2.0  11.0   0.0  10.0   0.0   28.0
23
685    0.0     ZETA  5.0   1.0   4.0  15.0   1.0   6.0   32.0
27
458  459.0    OMEGA  6.0   1.0   0.0   0.0   0.0   0.0    7.0
6
464  465.0    OMEGA  6.0   1.0   9.0  11.0   8.0  10.0   45.0
38
689    0.0     ZETA  6.0   3.0   4.0   9.0  10.0  15.0   47.0
39
484  485.0    OMEGA  6.0   3.0   7.0  11.0  11.0  14.0   52.0
43
51    52.0    ALPHA  6.0  12.0  10.0  11.0  10.0   3.0   52.0
43
176  177.0     BETA  6.0   0.0  13.0   8.0  18.0  14.0   59.0
49
25    26.0    ALPHA  6.0  10.0  10.0  11.0  13.0  10.0   60.0
50
345  346.0  EPSILON  6.0   0.0   1.0  11.0   9.0   4.0   31.0
26
194  195.0    DELTA  6.0   0.0   1.0   0.0  10.0  13.0   30.0
25
98    99.0     BETA  6.0   7.0  16.0   9.0  13.0  13.0   64.0
53
358  359.0    GAMMA  6.0   0.0   3.0   3.0  10.0   4.0   26.0
22
240  241.0    DELTA  6.0   6.0   2.0   3.0  10.0  11.0   38.0
32
```

	Grade
601	F
514	C
414	F
551	F
556	F
564	F
402	F
395	F
336	F
244	F
650	F
495	F
207	C
487	F
503	F
50	C
673	F
427	F
57	F
88	F
611	F
82	F
85	C
223	D
70	D
549	F
20	F
563	F
398	F
27	F
75	F
360	F
190	F
125	C
337	F
139	D
478	F
685	F
458	F
464	F
689	F
484	D
51	D
176	D
25	C
345	F
194	F
98	C

```
358     F
240     F
```

```
df = df[
    (df['DV'] < 10.0) |
    (df['PP'] < 10.0) |
    (df['M2'] < 10.0) |
    (df['BEEE'] < 10.0) |
    (df['FL'] < 10.0) |
    (df['FIMS'] < 10.0)
]
df['SECTION'].value_counts()
```

```
SECTION
ZETA       69
EPSILON    67
OMEGA      62
DELTA      58
GAMMA      57
BETA       54
ALPHA      39
SIGMA      26
0          13
Name: count, dtype: int64
```

```
df['backlogs'] = (df[['DV', 'M2', 'PP', 'BEEE', 'FL', 'FIMS']] <
10).sum(axis=1)
df
```

```
C:\Users\VINITHA\AppData\Local\Temp\ipykernel_16064\4076804967.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df['backlogs'] = (df[['DV', 'M2', 'PP', 'BEEE', 'FL', 'FIMS']] <
10).sum(axis=1)
```

```
     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
1     2.0   ALPHA  19.0  12.0  16.0  16.0  18.0   3.0   84.0
70
3     4.0   ALPHA  15.0   9.0  19.0  17.0  19.0  15.0   94.0
78
5     6.0   ALPHA  17.0  16.0  18.0  10.0  15.0   9.0   85.0
71
```

```
8     9.0   ALPHA  10.0  18.0   0.0  20.0  19.0  15.0   82.0
68
..    ...     ...   ...   ...   ...   ...   ...   ...    ...          .
..
712   0.0    ZETA  15.0  10.0   7.0  18.0  18.0  16.0   84.0
70
713   0.0    ZETA  19.0   8.0   8.0  19.0  17.0  18.0   89.0
74
714   0.0    ZETA  12.0   1.0   7.0  10.0  20.0   8.0   58.0
48
715   0.0    ZETA  17.0   6.0  14.0  14.0  17.0  18.0   86.0
72
716   0.0    ZETA  12.0   1.0   6.0   7.0  15.0  12.0   53.0
44

    Grade  backlogs
0      C+         2
1       B         1
3       B         1
5       B         1
8      C+         1
..    ...       ...
712     B         1
713     B         2
714     D         3
715     B         1
716     D         3

[445 rows x 12 columns]

df.sort_values('backlogs').head(50)

      S.NO  SECTION    DV    M2    PP  BEEE    FL  FIMS  Total
Percentage  \
362  363.0    GAMMA  13.0   9.0  15.0  16.0  20.0  16.0   89.0
74
247  248.0    DELTA  15.0  16.0  14.0   5.0  13.0  19.0   82.0
68
248  249.0    DELTA  16.0   6.0  14.0  13.0  13.0  16.0   78.0
65
500  501.0    OMEGA  17.0   3.0  14.0  11.0  19.0  17.0   81.0
68
499  500.0    OMEGA  12.0   4.0  19.0  17.0  14.0  15.0   81.0
68
497  498.0    OMEGA  19.0   7.0  19.0  18.0  18.0  17.0   98.0
82
494  495.0    OMEGA  18.0  18.0  19.0  18.0   8.0  18.0   99.0
82
259  260.0    DELTA  14.0   8.0  12.0  13.0  13.0  18.0   78.0
65
```

492	493.0	OMEGA	20.0	4.0	16.0	19.0	20.0	16.0	95.0
79									
261	262.0	DELTA	11.0	1.0	10.0	12.0	10.0	16.0	60.0
50									
489	490.0	OMEGA	19.0	7.0	16.0	16.0	20.0	18.0	96.0
80									
488	489.0	OMEGA	20.0	9.0	19.0	17.0	20.0	18.0	103.0
86									
486	487.0	OMEGA	15.0	7.0	18.0	15.0	16.0	15.0	86.0
72									
266	267.0	DELTA	18.0	10.0	17.0	9.0	15.0	17.0	86.0
72									
267	268.0	DELTA	18.0	10.0	17.0	6.0	20.0	17.0	88.0
73									
485	486.0	OMEGA	17.0	1.0	11.0	17.0	20.0	19.0	85.0
71									
270	271.0	EPSILON	15.0	4.0	10.0	12.0	11.0	13.0	65.0
54									
271	272.0	EPSILON	18.0	9.0	11.0	20.0	17.0	15.0	90.0
75									
482	483.0	OMEGA	15.0	7.0	14.0	16.0	16.0	16.0	84.0
70									
479	480.0	OMEGA	20.0	8.0	19.0	14.0	20.0	13.0	94.0
78									
278	279.0	EPSILON	17.0	8.0	11.0	15.0	15.0	12.0	78.0
65									
279	280.0	EPSILON	15.0	4.0	14.0	20.0	19.0	16.0	88.0
73									
501	502.0	OMEGA	20.0	9.0	20.0	18.0	20.0	20.0	107.0
89									
280	281.0	EPSILON	15.0	3.0	10.0	11.0	13.0	11.0	63.0
52									
502	503.0	OMEGA	20.0	7.0	17.0	17.0	20.0	18.0	99.0
82									
510	511.0	OMEGA	20.0	6.0	16.0	11.0	20.0	14.0	87.0
72									
570	571.0	SIGMA	14.0	9.0	13.0	16.0	18.0	16.0	86.0
72									
559	560.0	SIGMA	13.0	9.0	14.0	12.0	15.0	19.0	82.0
68									
558	559.0	SIGMA	12.0	10.0	11.0	3.0	10.0	12.0	58.0
48									
550	551.0	SIGMA	17.0	9.0	17.0	11.0	13.0	17.0	84.0
70									
365	366.0	GAMMA	18.0	11.0	9.0	20.0	15.0	13.0	86.0
72									
536	537.0	OMEGA	15.0	4.0	12.0	17.0	18.0	16.0	82.0
68									
535	536.0	OMEGA	17.0	2.0	11.0	13.0	20.0	14.0	77.0

```
64
534  535.0    OMEGA  20.0  12.0   9.0  18.0  20.0  19.0   98.0
82
213  214.0    DELTA  14.0  13.0   5.0  12.0  13.0  20.0   77.0
64
531  532.0    OMEGA  20.0   6.0  14.0  19.0  20.0  15.0   94.0
78
216  217.0    DELTA  16.0   8.0  18.0  11.0  13.0  14.0   80.0
67
530  531.0    OMEGA  17.0   4.0  15.0  10.0  17.0  13.0   76.0
63
218  219.0    DELTA  15.0   8.0  17.0  14.0  18.0  19.0   91.0
76
529  530.0    OMEGA  19.0   2.0  20.0  13.0  20.0  19.0   93.0
78
528  529.0    OMEGA  19.0   7.0  20.0  11.0  13.0  18.0   88.0
73
527  528.0    OMEGA  16.0   4.0  19.0  15.0  15.0  18.0   87.0
72
526  527.0    OMEGA  20.0   6.0  20.0  18.0  17.0  18.0   99.0
82
225  226.0    DELTA  16.0  12.0  17.0   8.0  18.0  19.0   90.0
75
233  234.0    DELTA  12.0   3.0  10.0  10.0  10.0  18.0   63.0
52
512  513.0    OMEGA  20.0  20.0   5.0  19.0  18.0  14.0   96.0
80
511  512.0    OMEGA  20.0   6.0  18.0  13.0  20.0  18.0   95.0
79
241  242.0    DELTA  15.0   1.0  18.0  18.0  20.0  19.0   91.0
76
572  573.0    SIGMA  14.0   1.0  14.0  10.0  20.0  17.0   76.0
63
471  472.0    OMEGA  16.0   7.0  11.0  17.0  15.0  17.0   83.0
69

    Grade  backlogs
362     B         1
247    C+         1
248    C+         1
500    C+         1
499    C+         1
497    B+         1
494    B+         1
259    C+         1
492     B         1
261     C         1
489    B+         1
488    B+         1
```

```
486     B          1
266     B          1
267     B          1
485     B          1
270     C          1
271     B          1
482     B          1
479     B          1
278    C+          1
279     B          1
501    B+          1
280     C          1
502    B+          1
510     B          1
570     B          1
559    C+          1
558     D          1
550     B          1
365     B          1
536    C+          1
535    C+          1
534    B+          1
213    C+          1
531     B          1
216    C+          1
530    C+          1
218     B          1
529     B          1
528     B          1
527     B          1
526    B+          1
225     B          1
233     C          1
512    B+          1
511     B          1
241     B          1
572    C+          1
471    C+          1
```

```
df=df.sort_values('backlogs')
df
```

```
      S.NO  SECTION    DV    M2    PP  BEEE    FL  FIMS  Total
Percentage  \
362  363.0    GAMMA  13.0   9.0  15.0  16.0  20.0  16.0   89.0
74
247  248.0    DELTA  15.0  16.0  14.0   5.0  13.0  19.0   82.0
68
248  249.0    DELTA  16.0   6.0  14.0  13.0  13.0  16.0   78.0
65
```

```
500  501.0    OMEGA  17.0   3.0  14.0  11.0  19.0  17.0   81.0
68
499  500.0    OMEGA  12.0   4.0  19.0  17.0  14.0  15.0   81.0
68
..     ...      ...   ...   ...   ...   ...   ...   ...    ...
...
556  557.0    SIGMA   0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
244  245.0    DELTA   0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
336  337.0  EPSILON   0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
414  415.0    GAMMA   0.0   0.0   0.0   0.0   0.0   0.0    0.0
0
635    0.0     ZETA   9.0   0.0   2.0   2.0   3.0   9.0   25.0
21

    Grade  backlogs
362     B         1
247    C+         1
248    C+         1
500    C+         1
499    C+         1
..    ...       ...
556     F         6
244     F         6
336     F         6
414     F         6
635     F         6

[445 rows x 12 columns]
```

```
df.value_counts('backlogs')
```

```
backlogs
1    172
2    121
3     69
4     43
5     23
6     17
Name: count, dtype: int64
```

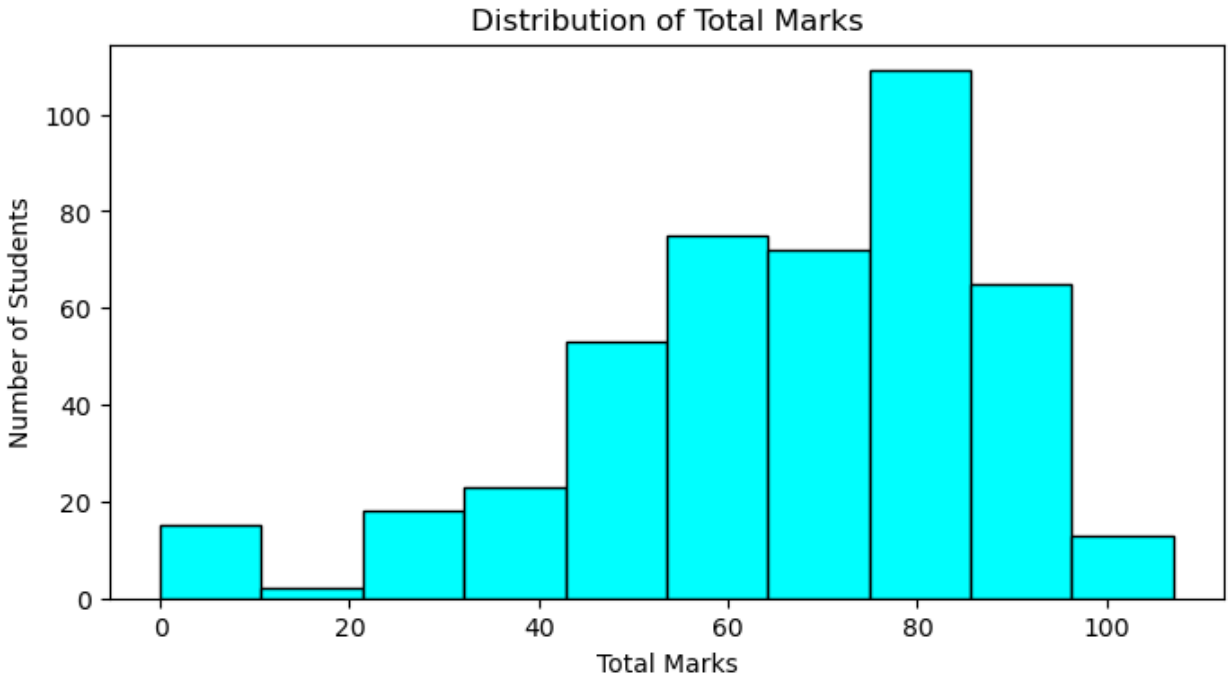
```
import pandas as pd

print("Columns in DataFrame:", df.columns)


passing_marks = 10

# List of subject columns
```

```
subject_columns = ['DV', 'M2', 'PP', 'BEEE', 'FL', 'FIMS']

missing_columns = [col for col in subject_columns if col not in
df.columns]
if missing_columns:
    print(f"Missing columns in DataFrame: {missing_columns}")
else:
    print("All required columns are present.")


df[subject_columns] = df[subject_columns].apply(pd.to_numeric,
errors='coerce')


df['Backlogs'] = (df[subject_columns] < passing_marks).sum(axis=1)

print("DataFrame with backlog counts:")
print(df[['Backlogs']])

total_backlogs = df['Backlogs'].sum()
print(f"Total number of backlogs across all students:
{total_backlogs}")
```

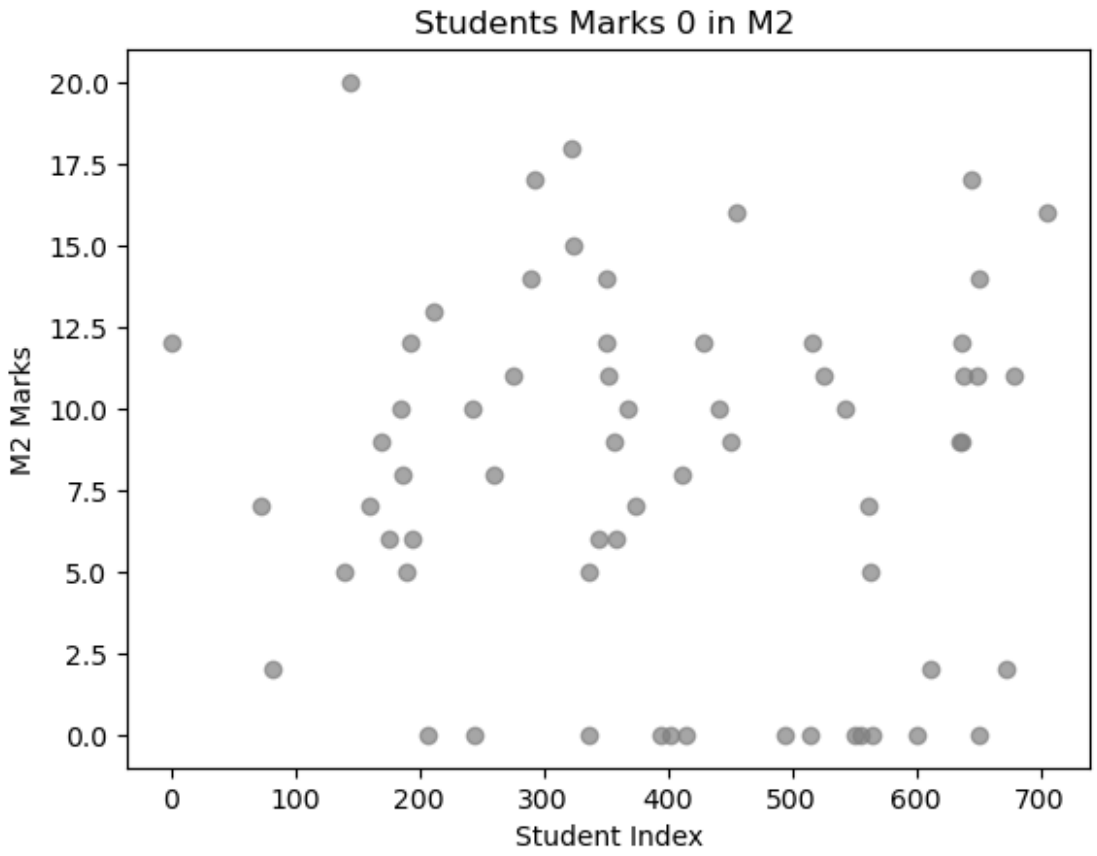
```
Columns in DataFrame: Index(['S.NO', 'SECTION', 'DV', 'M2', 'PP',
'BEEE', 'FL', 'FIMS', 'Total',
       'Percentage', 'Grade', 'backlogs'],
      dtype='object')
All required columns are present.
DataFrame with backlog counts:
     Backlogs
362         1
247         1
248         1
500         1
499         1
..        ...
556         6
244         6
336         6
414         6
635         6

[445 rows x 1 columns]
Total number of backlogs across all students: 1010
```

```
plt.figure(figsize=[8, 4])
plt.hist(df['Total'], color='cyan', bins=10, edgecolor='black')
plt.title("Distribution of Total Marks")
plt.xlabel("Total Marks")
```

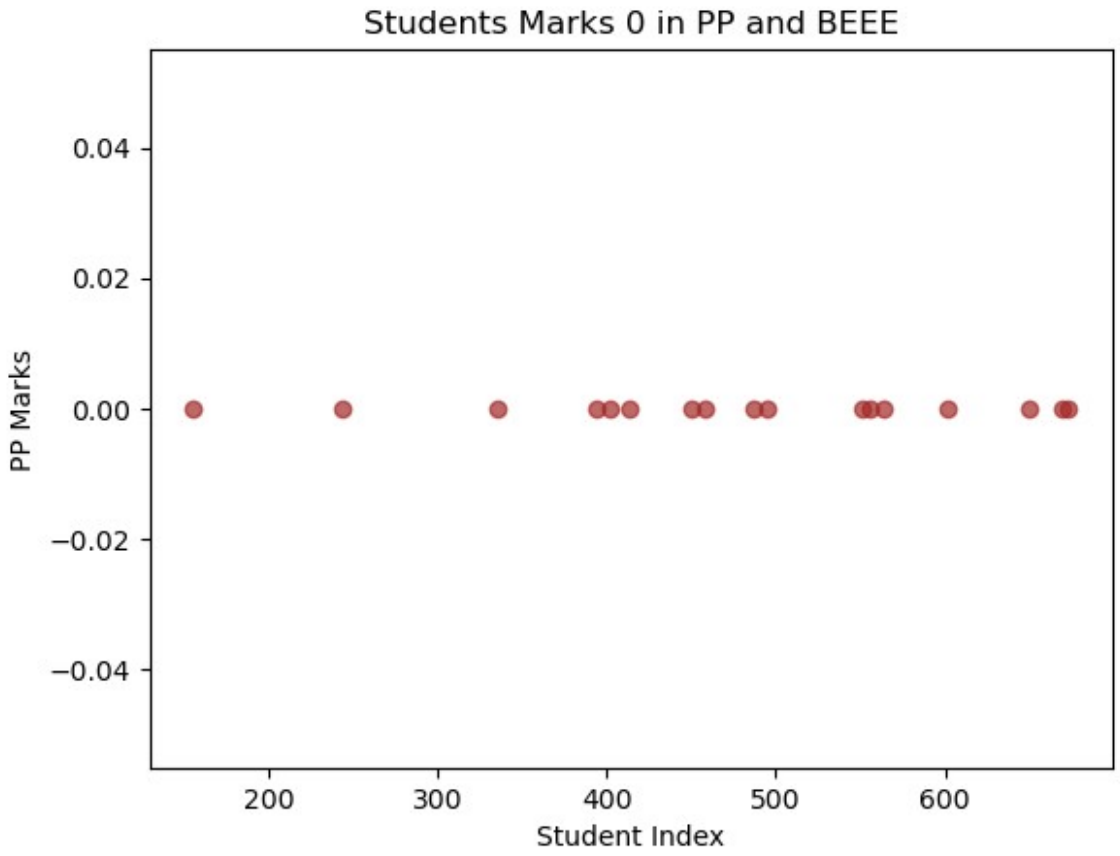
```
plt.ylabel("Number of Students")
plt.show()
```



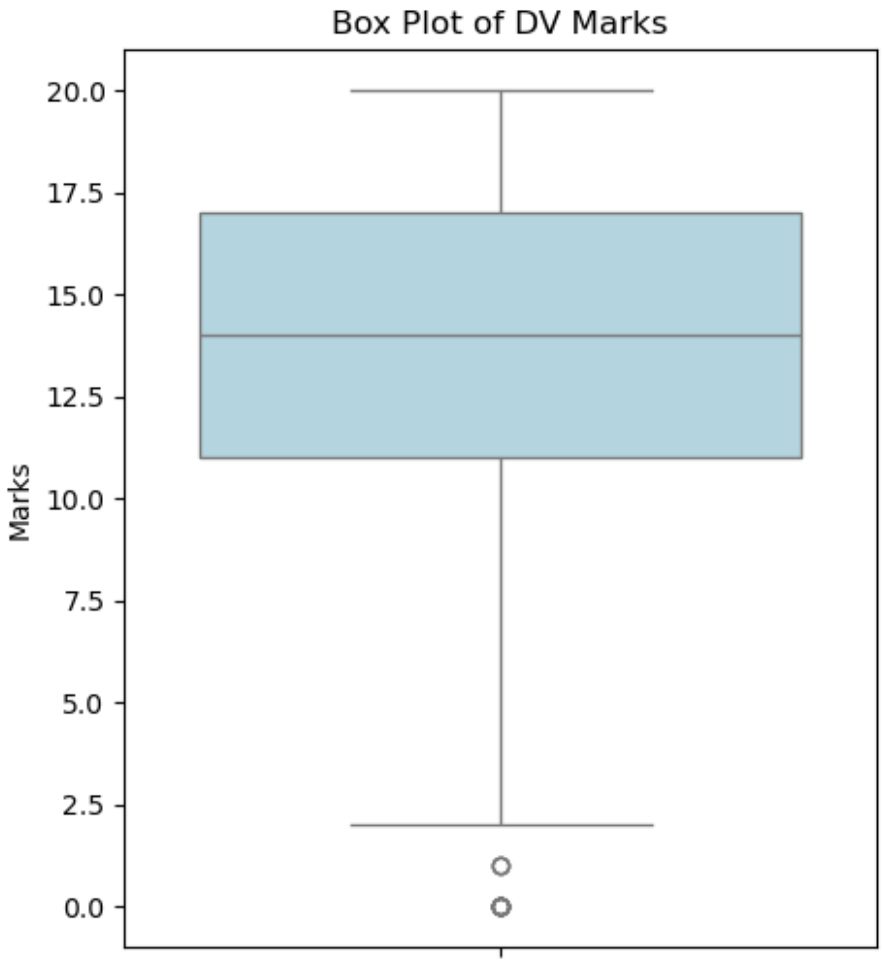

```
filtered_df = df[df['M2'] == 0]
plt.scatter(filtered_df.index, filtered_df['DV'], alpha=0.7,
color='grey')
plt.title("Students Marks 0 in M2 ")
plt.xlabel("Student Index")
plt.ylabel("M2 Marks")
plt.show()
```



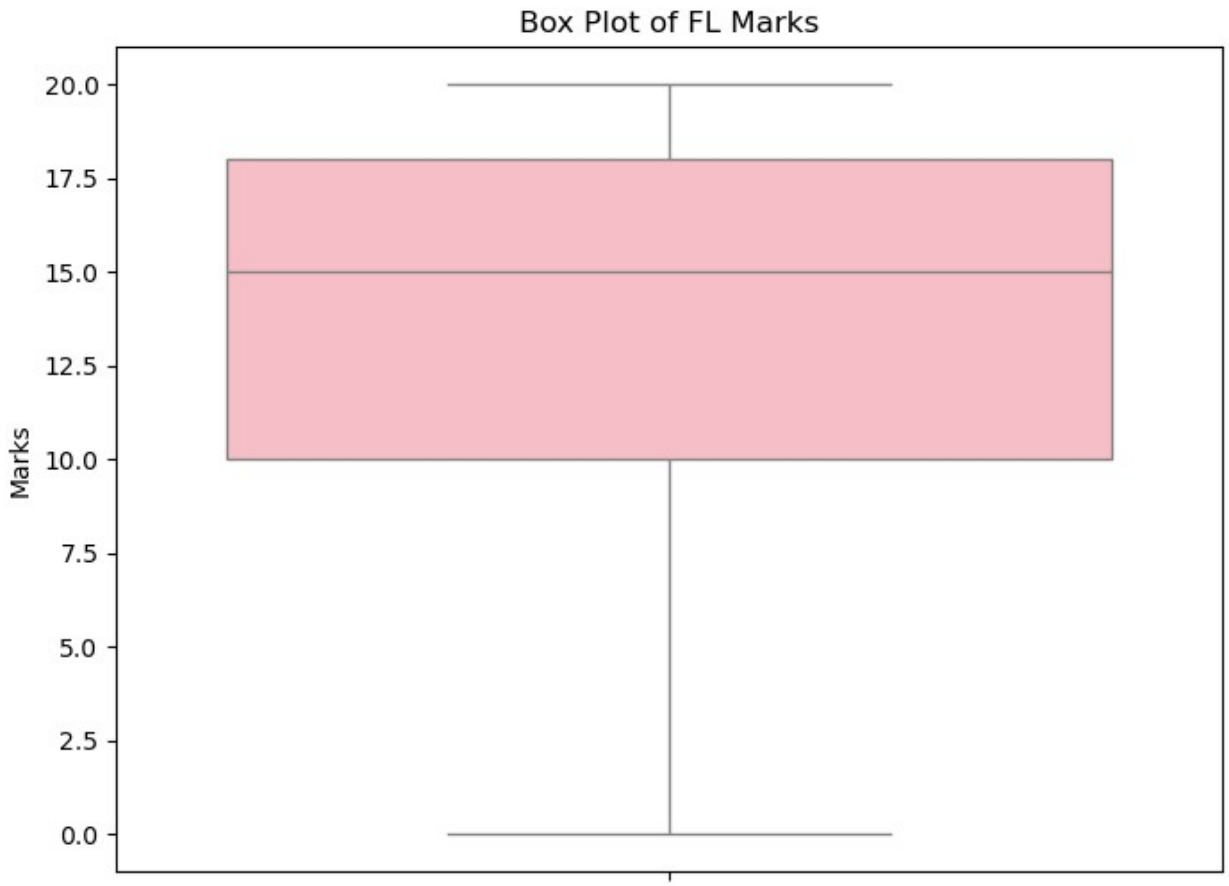

```
filtered_df = df[(df['PP'] == 0) & (df['BEEE'] == 0)]
plt.scatter(filtered_df.index, filtered_df['PP'], alpha=0.7,
color='brown')
plt.title("Students Marks 0 in PP and BEEE")
plt.xlabel("Student Index")
plt.ylabel("PP Marks")
plt.show()
```



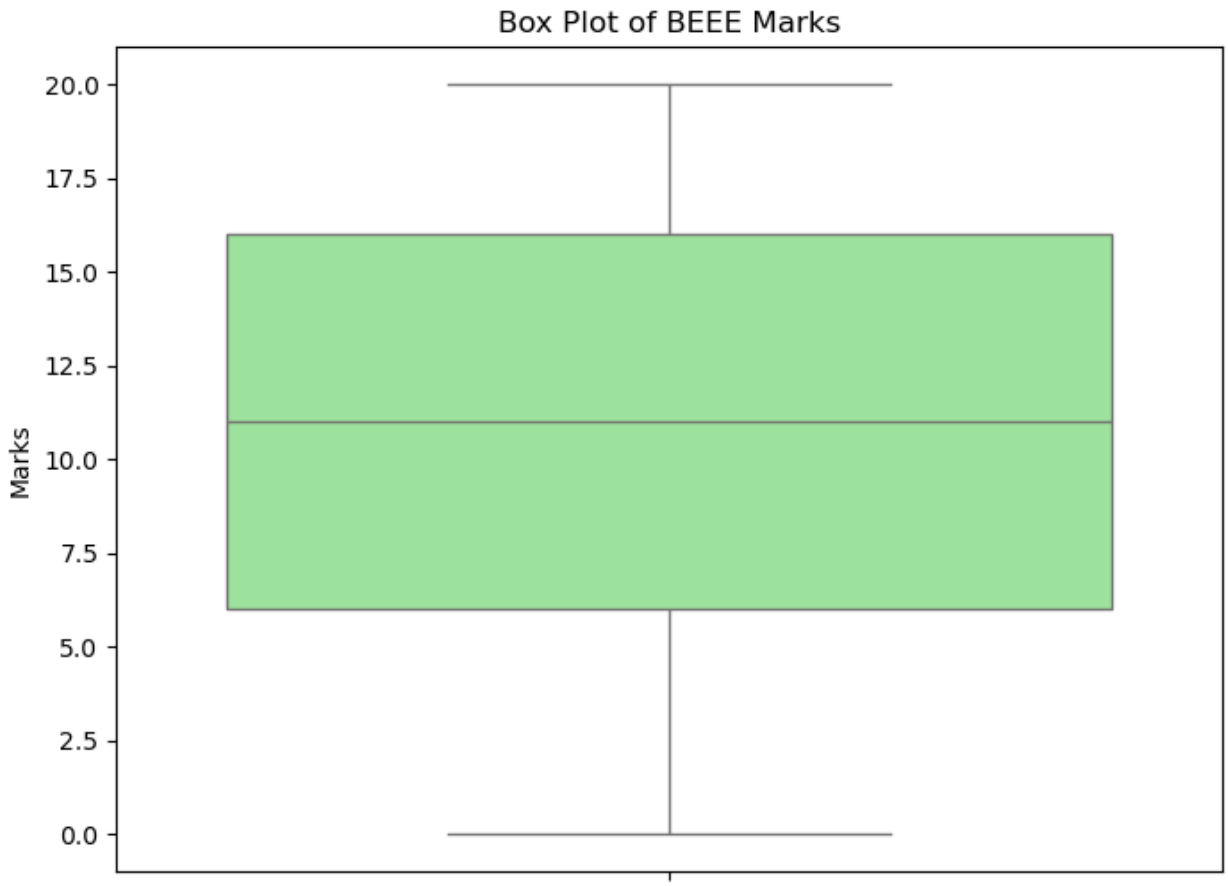

```
plt.figure(figsize=(5, 6))
sns.boxplot(data=df['DV'], color='lightblue')
plt.title("Box Plot of DV Marks")
plt.ylabel("Marks")
plt.show()
```



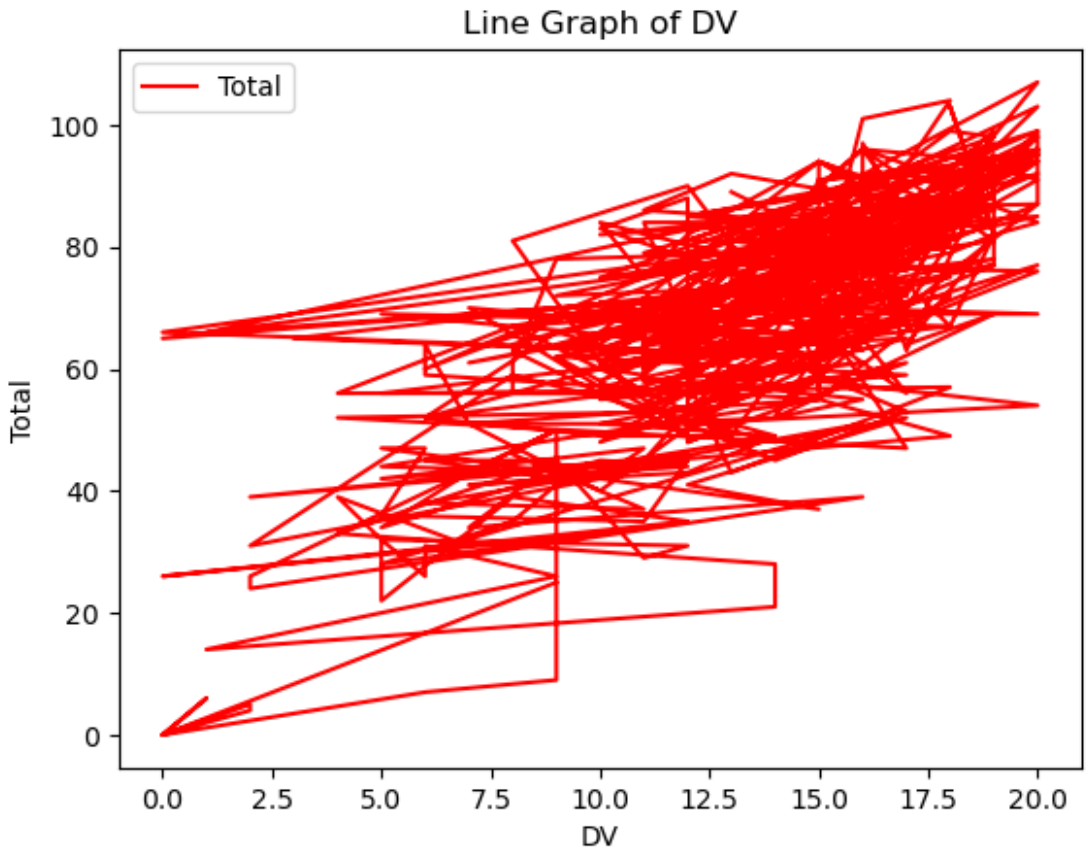

```
plt.figure(figsize=(8, 6))
sns.boxplot(data=df['FL'], color='lightpink')
plt.title("Box Plot of FL Marks")
plt.ylabel("Marks")
plt.show()
```



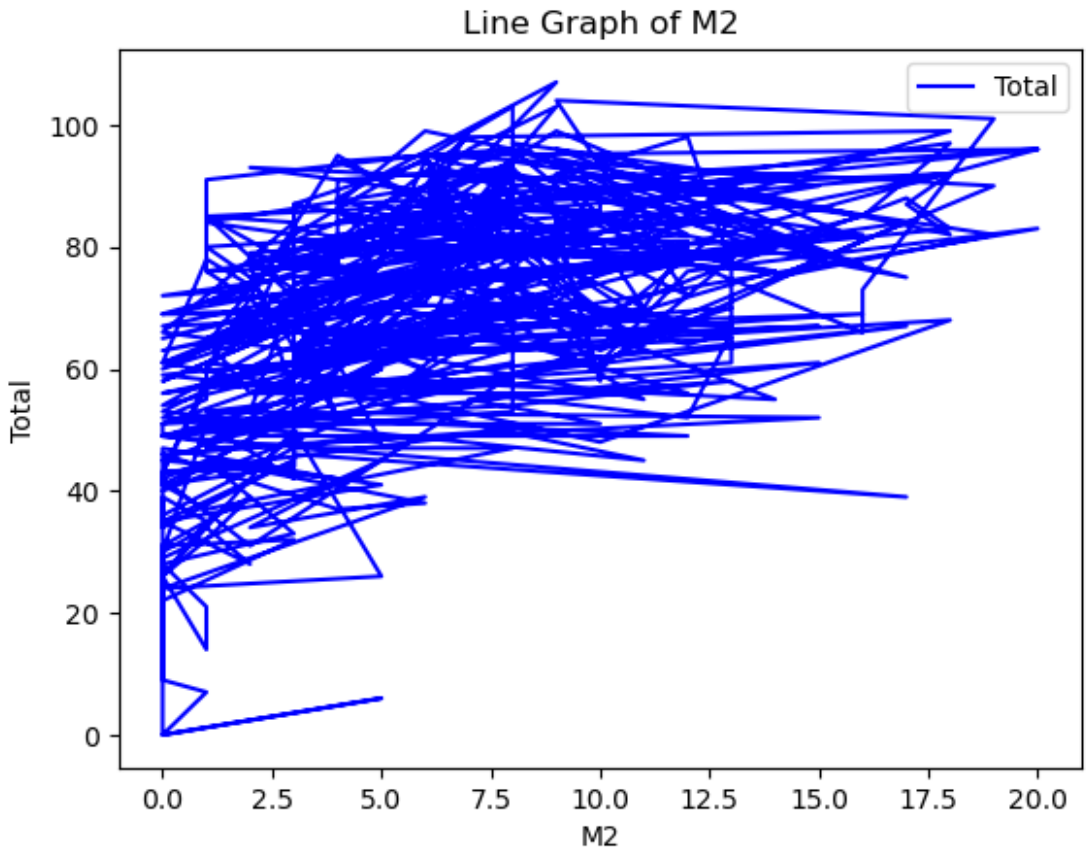

```
plt.figure(figsize=(8, 6))
sns.boxplot(data=df['BEEE'], color='lightgreen')
plt.title("Box Plot of BEEE Marks")
plt.ylabel("Marks")
plt.show()
```



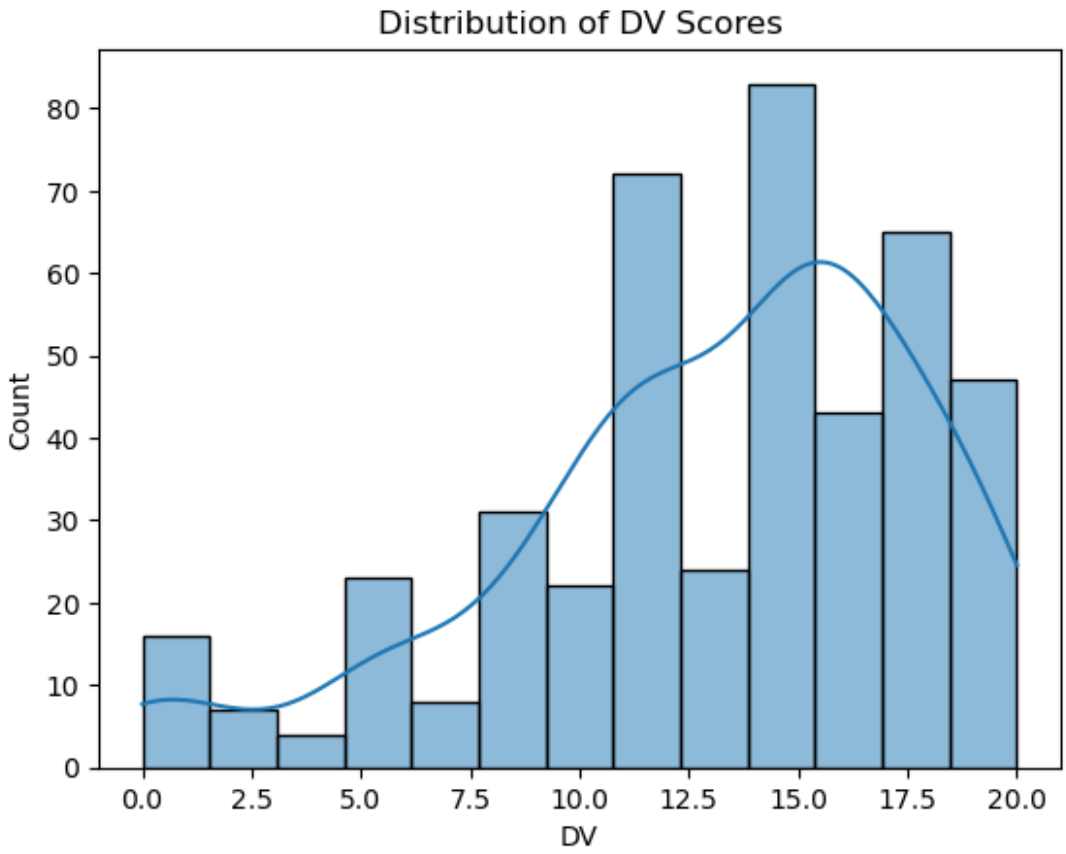

```
df.plot.line(x='DV',y='Total',color='red')
plt.title("Line Graph of DV")
plt.ylabel("Total")
plt.show()
```

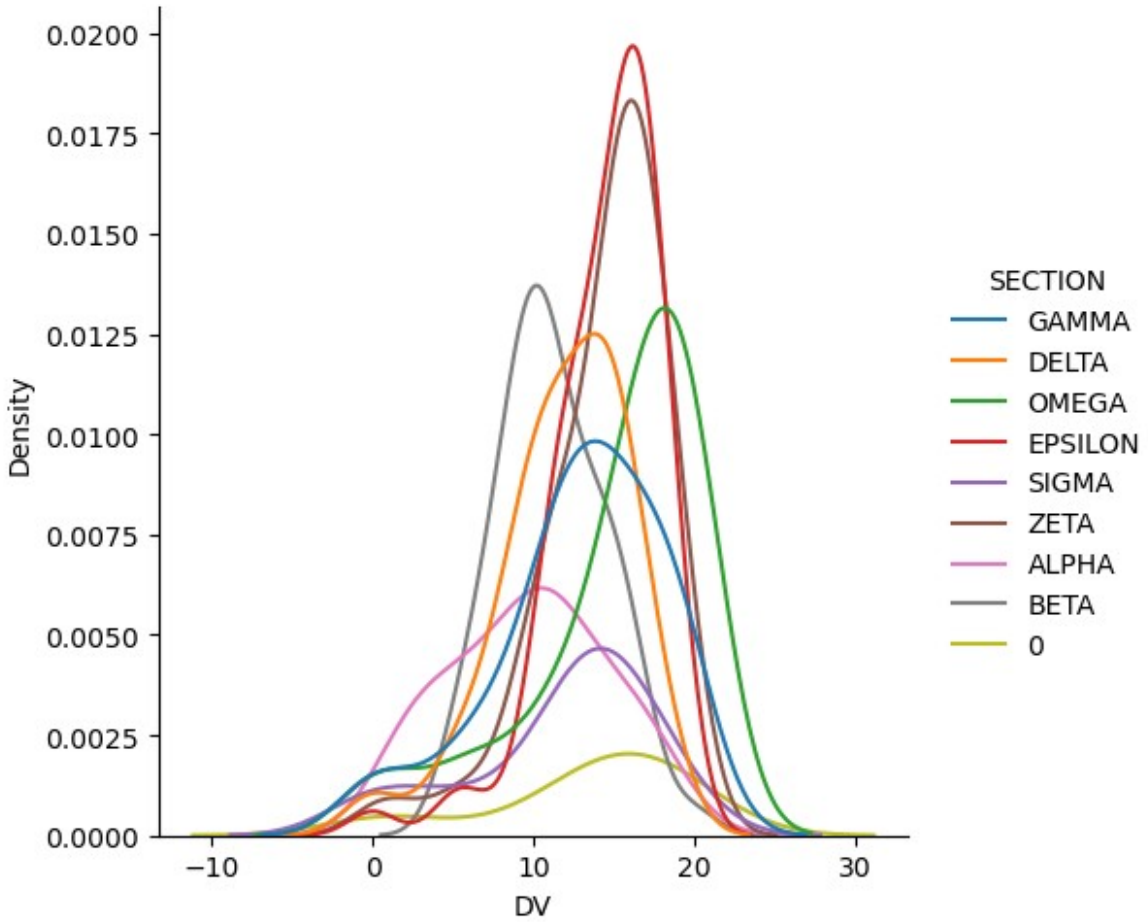



```
df.plot.line(x='M2',y='Total',color='blue')
plt.title("Line Graph of M2")
plt.ylabel("Total")
plt.show()
```

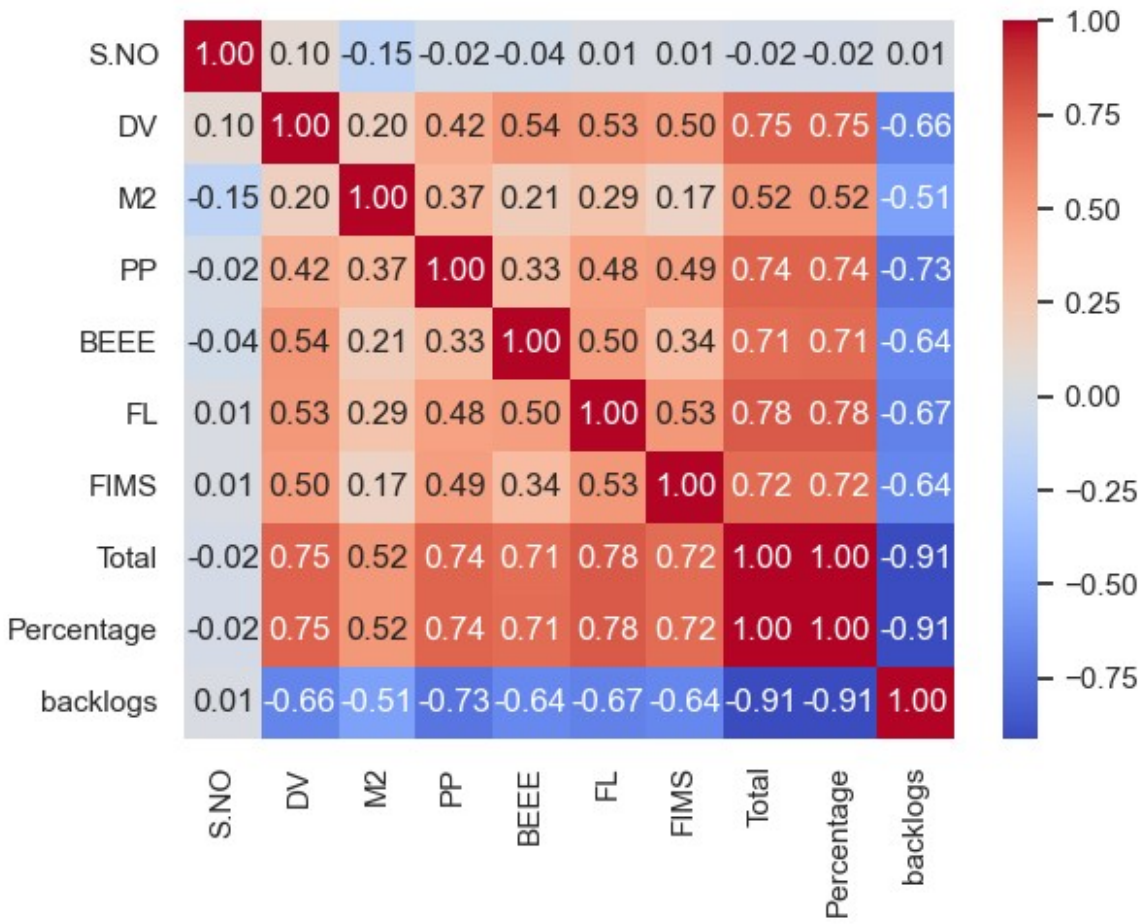



```
sns.histplot(df['DV'], kde=True)  # kde adds a kernel density estimate
plt.title('Distribution of DV Scores')
plt.show()

sns.displot(df, x="DV", hue="SECTION", kind="kde")
plt.show()
```

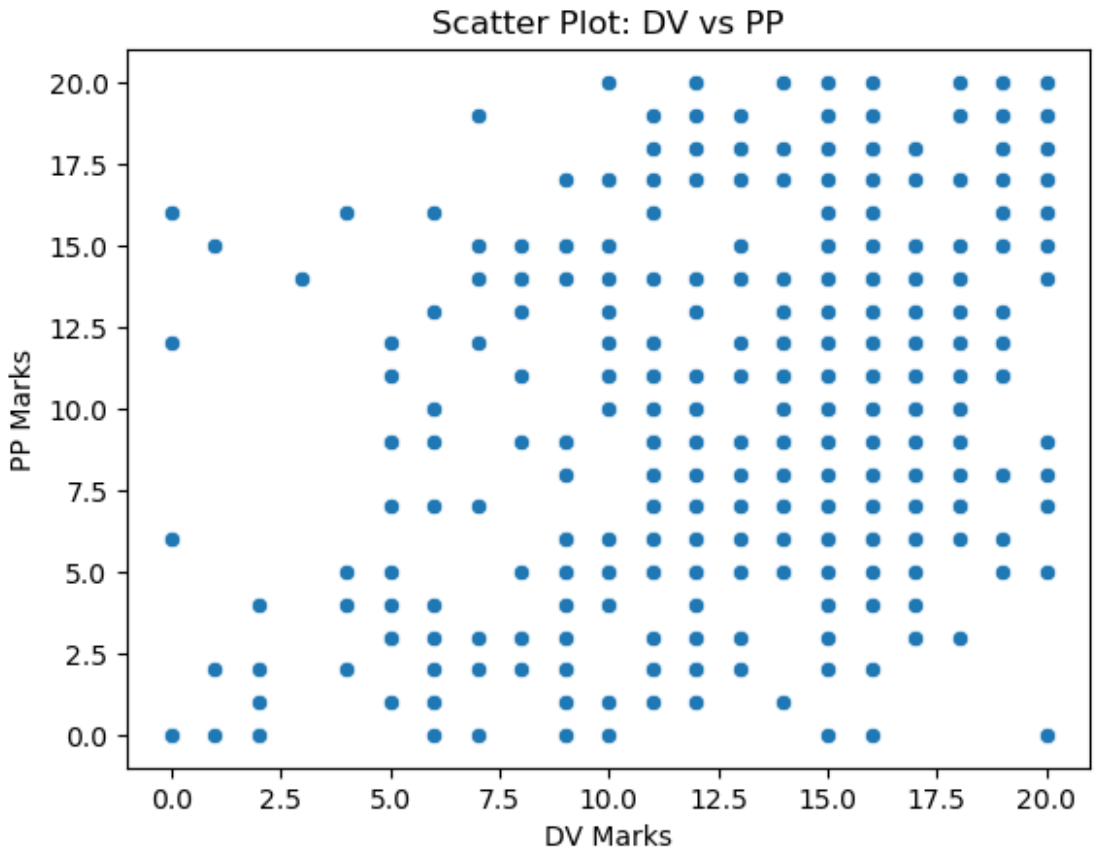

Distribution of DV Scores
Count
DV
0
10
20
30
40
50
60
70
80
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0



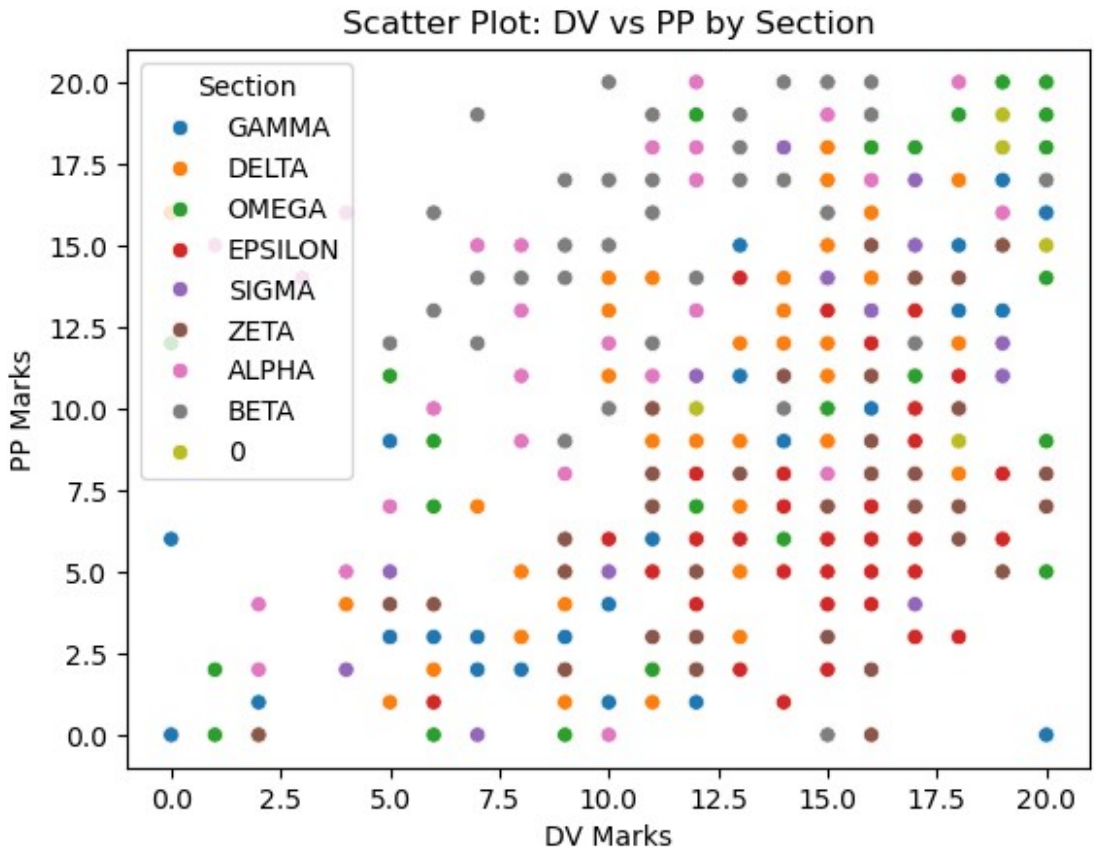

```
numeric_df = df.select_dtypes(include=['number'])

corr_matrix = numeric_df.corr()

sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.show()
```



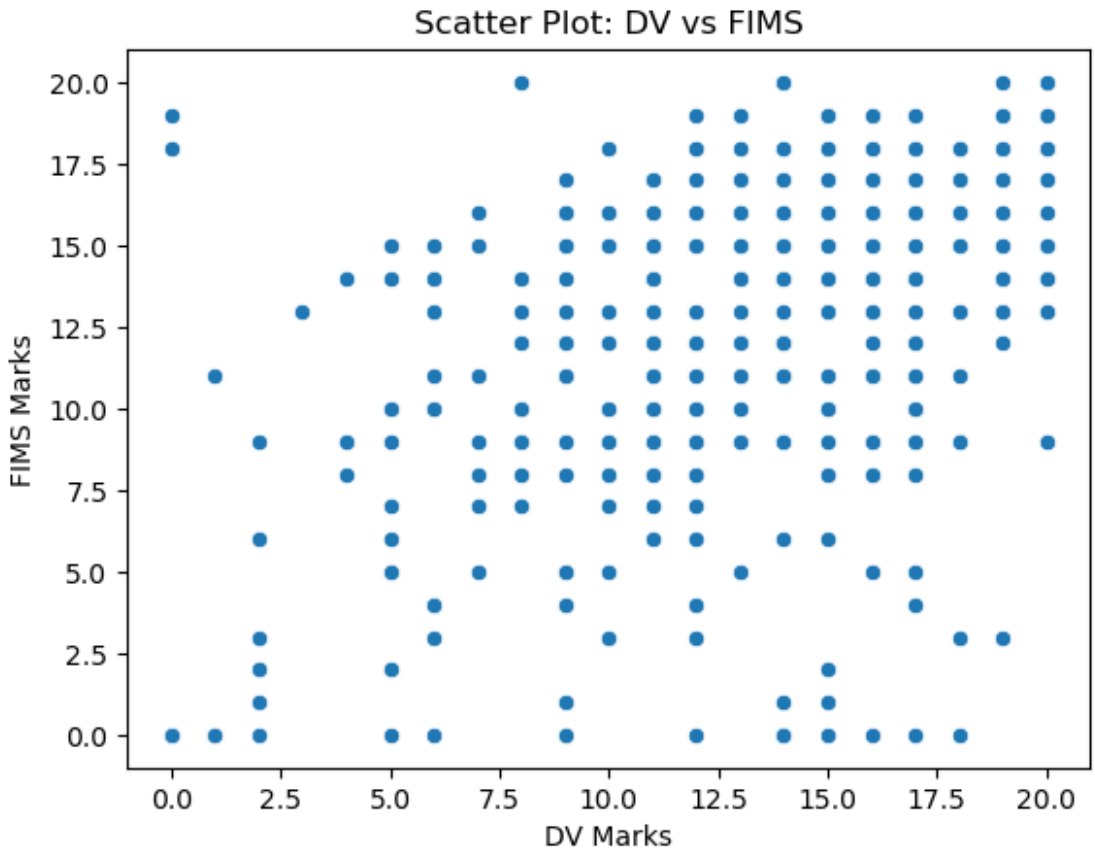

```
sns.scatterplot(x='DV', y='PP', data=df)
plt.title('Scatter Plot: DV vs PP')
plt.xlabel('DV Marks')
plt.ylabel('PP Marks')
plt.show()
```



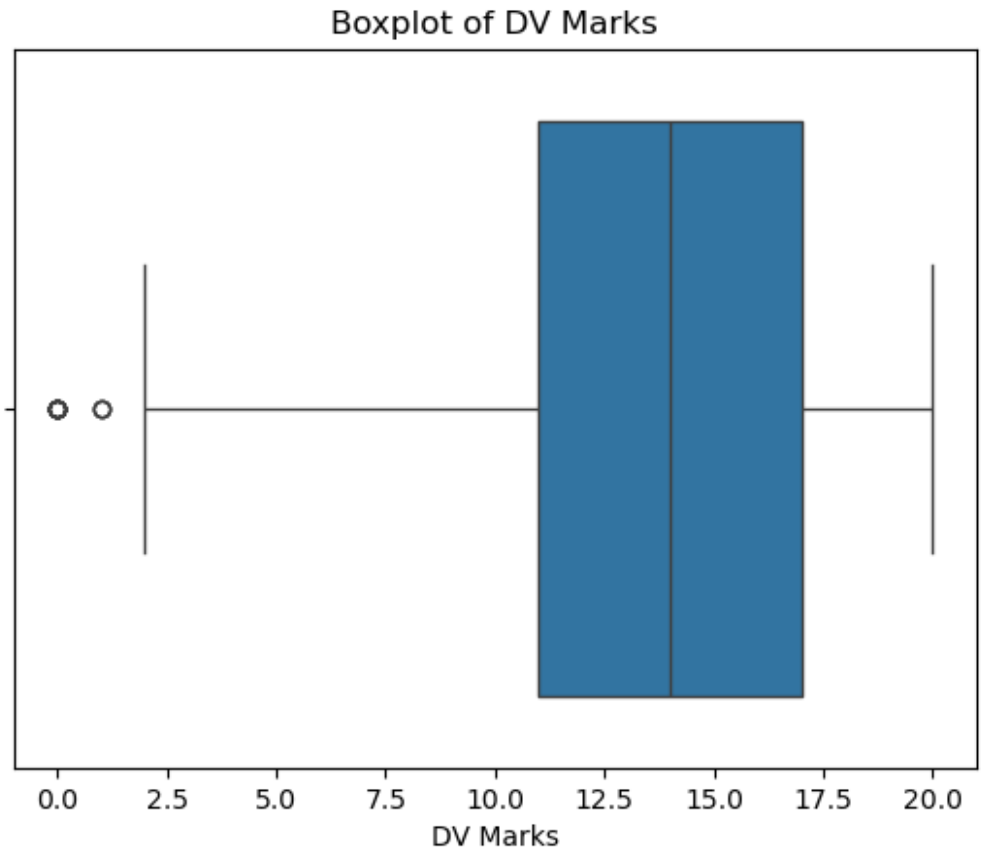

```
sns.scatterplot(x='DV', y='PP', hue='SECTION', data=df)
plt.title('Scatter Plot: DV vs PP by Section')
plt.xlabel('DV Marks')
plt.ylabel('PP Marks')
plt.legend(title='Section')
plt.show()
```



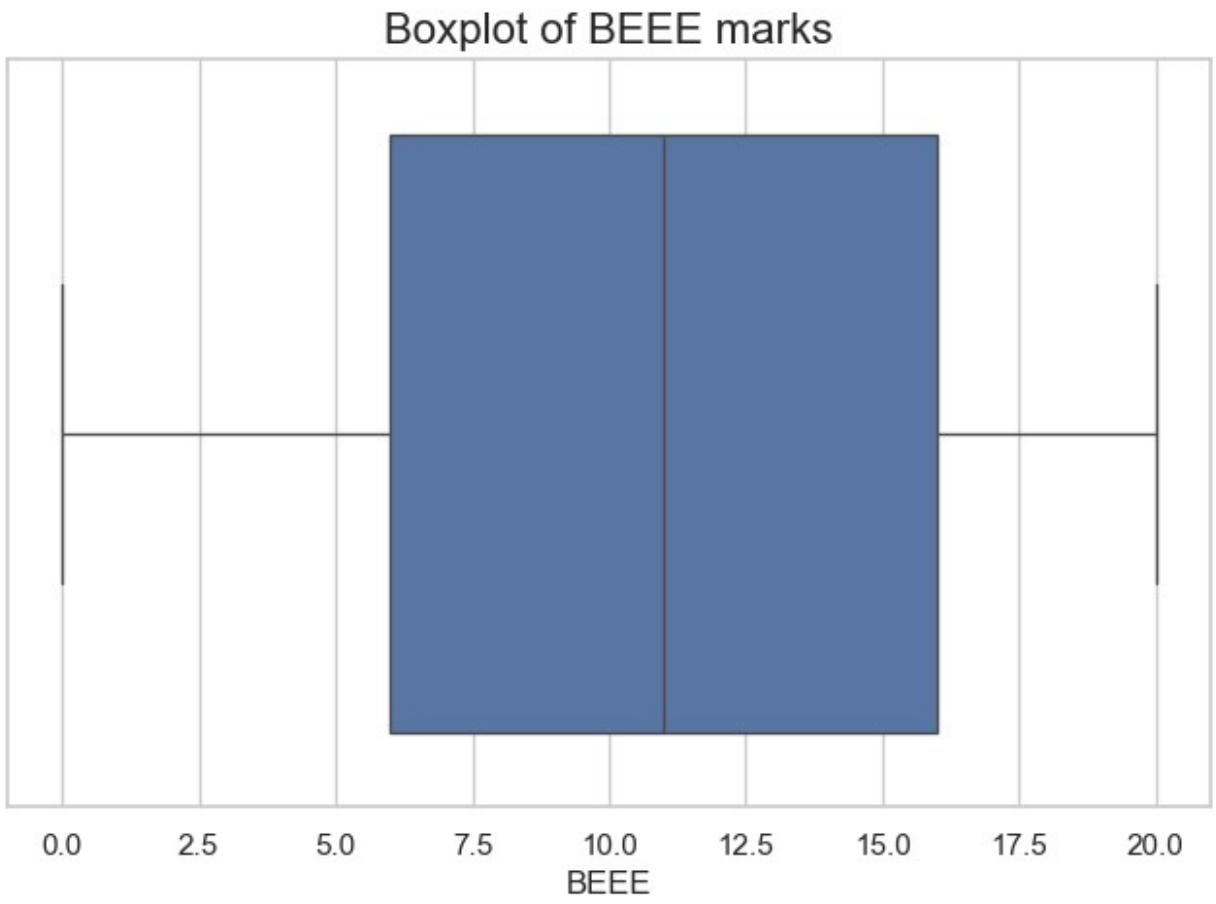

```
sns.scatterplot(x='DV', y='FIMS', data=df)
plt.title('Scatter Plot: DV vs FIMS')
plt.xlabel('DV Marks')
plt.ylabel('FIMS Marks')
plt.show()
```



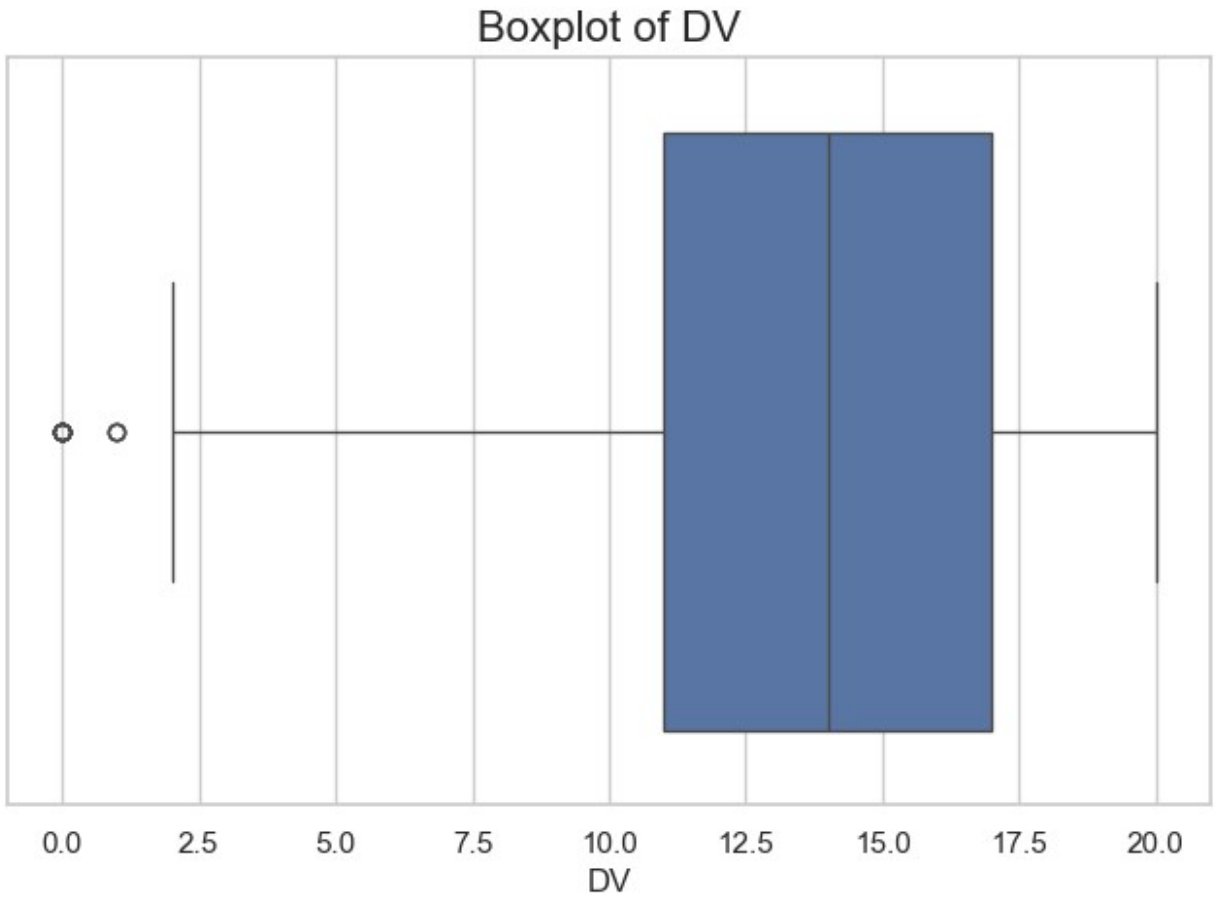

```
sns.boxplot(x='DV', data=df)
plt.title('Boxplot of DV Marks')
plt.xlabel('DV Marks')
plt.show()
```



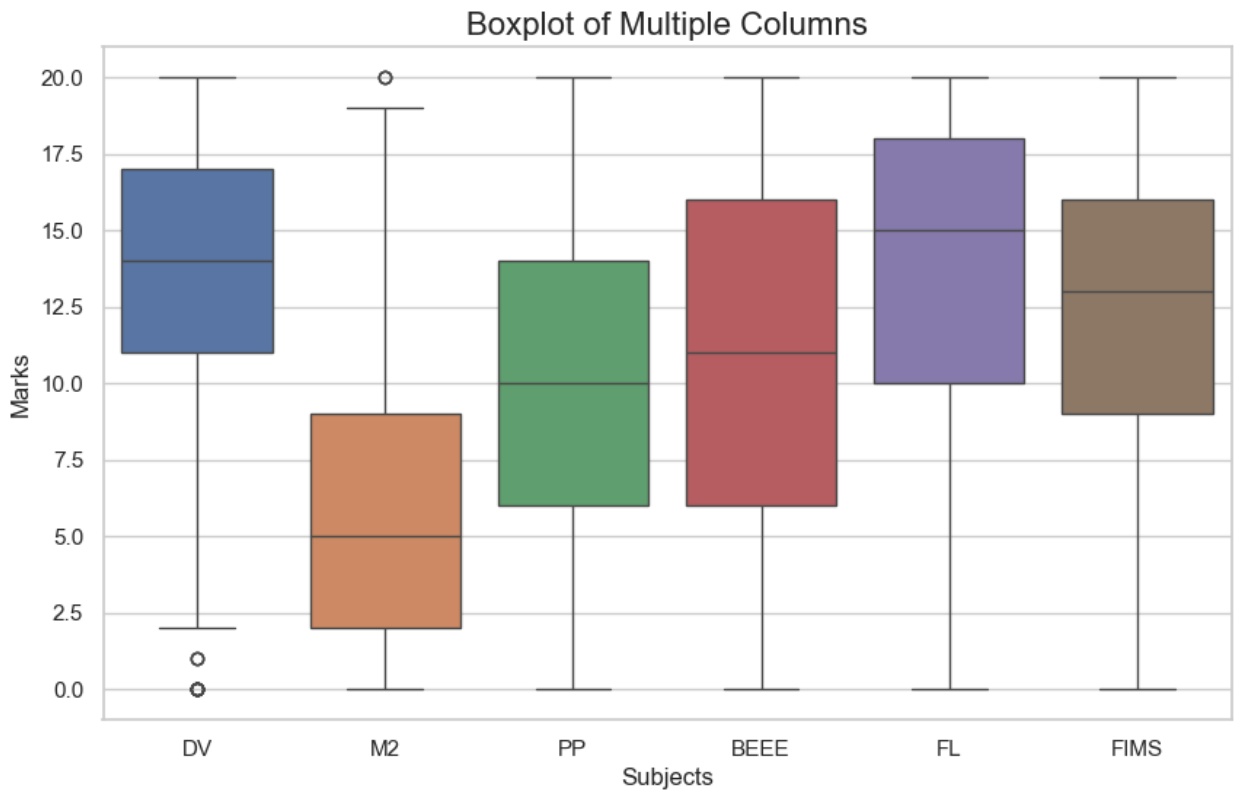

```
sns.set(style="whitegrid")
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x="BEEE")  # Replace "ColumnName" with the column
name

# Add titles and labels
plt.title("Boxplot of BEEE marks", fontsize=16)
plt.xlabel("BEEE", fontsize=12)

# Show the plot
plt.show()
```




```
sns.set(style="whitegrid")
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x="DV")  # Replace "ColumnName" with the column
name

# Add titles and labels
plt.title("Boxplot of DV", fontsize=16)
plt.xlabel("DV", fontsize=12)

# Show the plot
plt.show()
```




```
columns_to_plot = ["DV","M2","PP","BEEE","FL","FIMS"]
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.boxplot(data=df[columns_to_plot])

# Add titles and labels
plt.title("Boxplot of Multiple Columns", fontsize=16)
plt.xlabel("Subjects", fontsize=12)
plt.ylabel("Marks", fontsize=12)

# Show the plot
plt.show()
```




```
df[df.PP==0]
```

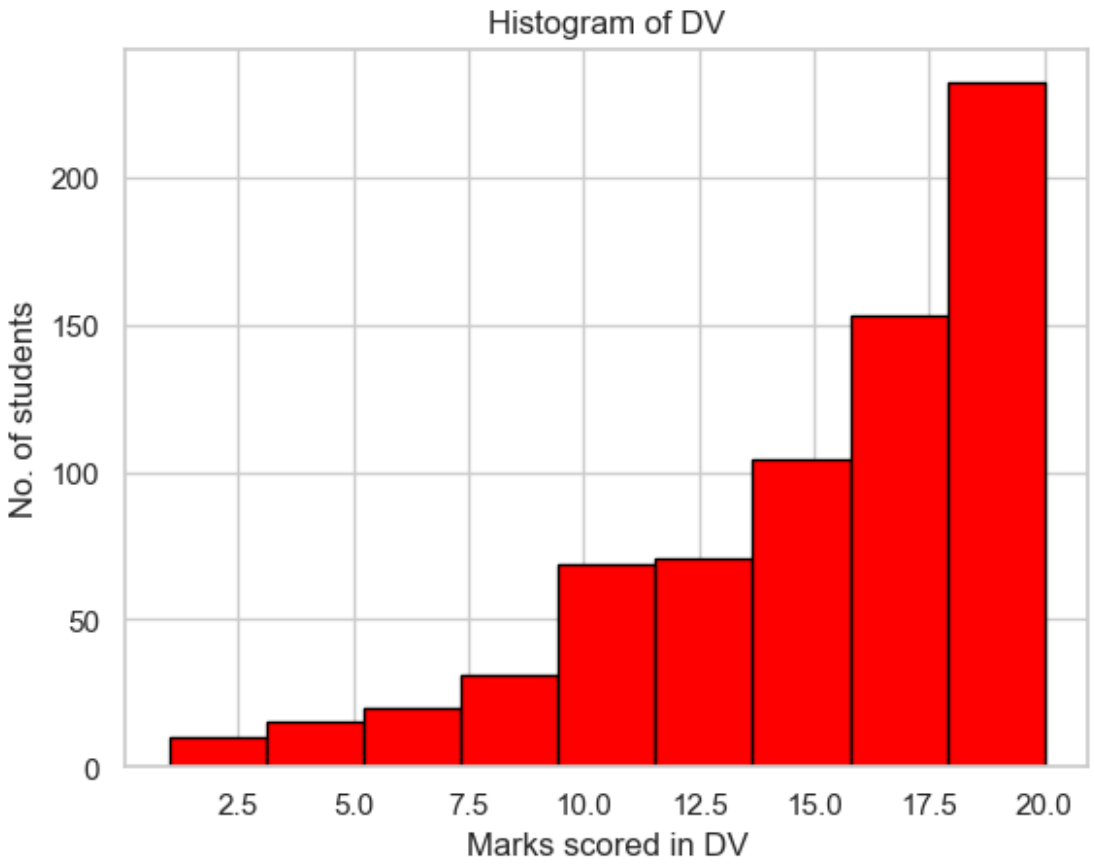
	S.NO	SECTION	DV	M2	PP	BEEE	FL	FIMS	Total Percentage
399	400.0	GAMMA	20.0	18.0	0.0	19.0	18.0	17.0	92.0 77
8	9.0	ALPHA	10.0	18.0	0.0	20.0	19.0	15.0	82.0 68
156	157.0	BETA	15.0	5.0	0.0	0.0	20.0	15.0	55.0 46
394	395.0	GAMMA	20.0	8.0	0.0	0.0	13.0	13.0	54.0 45
88	89.0	ALPHA	2.0	17.0	0.0	3.0	15.0	2.0	39.0 32
562	563.0	SIGMA	7.0	0.0	0.0	5.0	15.0	16.0	43.0 36
669	0.0	ZETA	16.0	6.0	0.0	0.0	8.0	9.0	39.0 32
611	0.0	0	2.0	0.0	0.0	3.0	10.0	9.0	24.0 20
450	451.0	OMEGA	9.0	0.0	0.0	0.0	0.0	0.0	9.0 8
458	459.0	OMEGA	6.0	1.0	0.0	0.0	0.0	0.0	7.0 6
650	0.0	ZETA	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
0
601    0.0         0   0.0   0.0  0.0   0.0   0.0   0.0     0.0
0
564  565.0     SIGMA   0.0   0.0  0.0   0.0   0.0   0.0     0.0
0
495  496.0     OMEGA   0.0   0.0  0.0   0.0   0.0   0.0     0.0
0
487  488.0     OMEGA   1.0   5.0  0.0   0.0   0.0   0.0     6.0
5
551  552.0     SIGMA   0.0   0.0  0.0   0.0   0.0   0.0     0.0
0
402  403.0     GAMMA   0.0   0.0  0.0   0.0   0.0   0.0     0.0
0
673    0.0      ZETA   2.0   0.0  0.0   0.0   2.0   1.0     5.0
4
556  557.0     SIGMA   0.0   0.0  0.0   0.0   0.0   0.0     0.0
0
244  245.0     DELTA   0.0   0.0  0.0   0.0   0.0   0.0     0.0
0
336  337.0   EPSILON   0.0   0.0  0.0   0.0   0.0   0.0     0.0
0
414  415.0     GAMMA   0.0   0.0  0.0   0.0   0.0   0.0     0.0
0
```

```
    Grade  backlogs
399     B         1
8      C+         1
156     D         3
394     D         3
88      F         4
562     F         4
669     F         5
611     F         5
450     F         6
458     F         6
650     F         6
601     F         6
564     F         6
495     F         6
487     F         6
551     F         6
402     F         6
673     F         6
556     F         6
244     F         6
336     F         6
414     F         6
```

```
import pandas as pd
import matplotlib.pyplot as plt
```

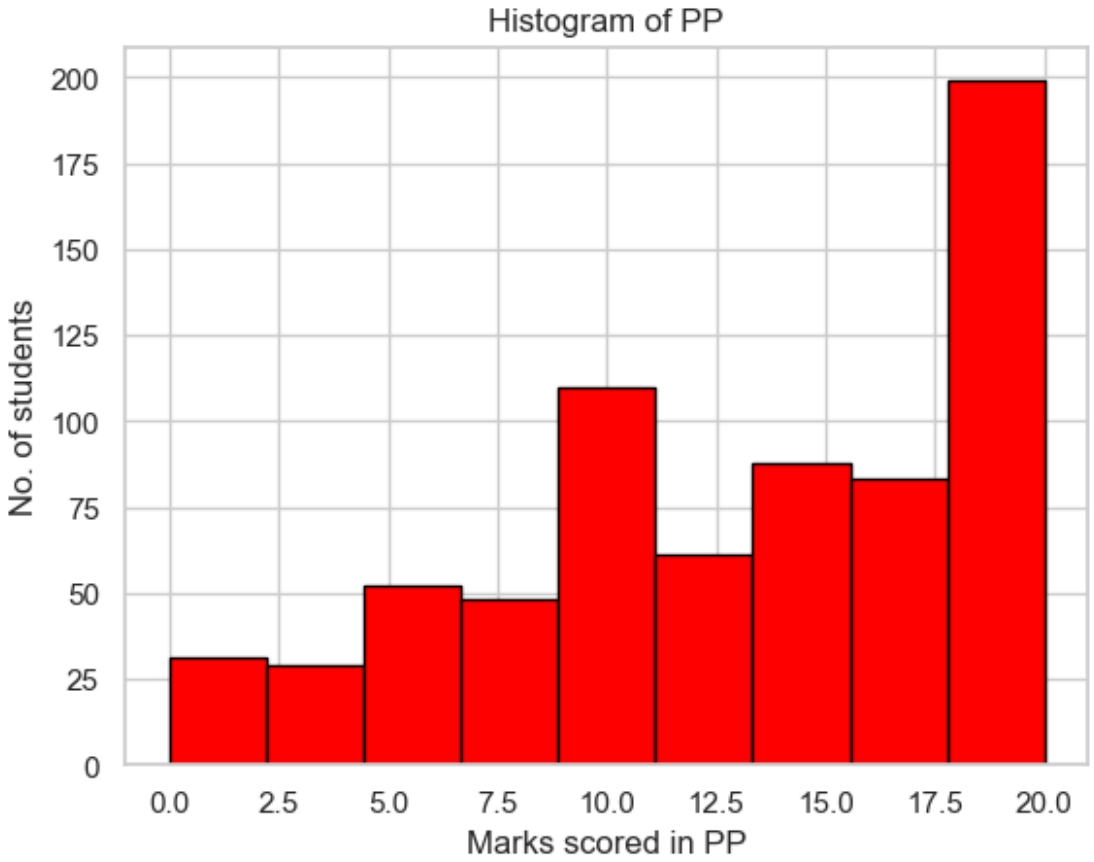
```
df = pd.read_excel("MIDMARKS.XLSX")


df['DV'] = pd.to_numeric(df['DV'], errors='coerce')


df = df.dropna(subset=['DV'])

# Plot histogram
plt.hist(df['DV'], color='red', edgecolor='black', bins=9)
plt.xlabel("Marks scored in DV")
plt.ylabel("No. of students")
plt.title("Histogram of DV")
plt.show()

# Calculate failures
passing_marks = 10
failures = df[df['DV'] < passing_marks]
num_failures = len(failures)

print(f"Number of students who failed in DV: {num_failures}")
```




```
Number of students who failed in DV: 76
```

```
import pandas as pd
import matplotlib.pyplot as plt



df = pd.read_excel("MIDMARKS.XLSX")


df['PP'] = pd.to_numeric(df['PP'], errors='coerce')


df = df.dropna(subset=['PP'])

# Plot histogram
plt.hist(df['PP'], color='red', edgecolor='black', bins=9)
plt.xlabel("Marks scored in PP")
plt.ylabel("No. of students")
plt.title("Histogram of PP")
plt.show()

# Calculate failures
passing_marks = 10
```

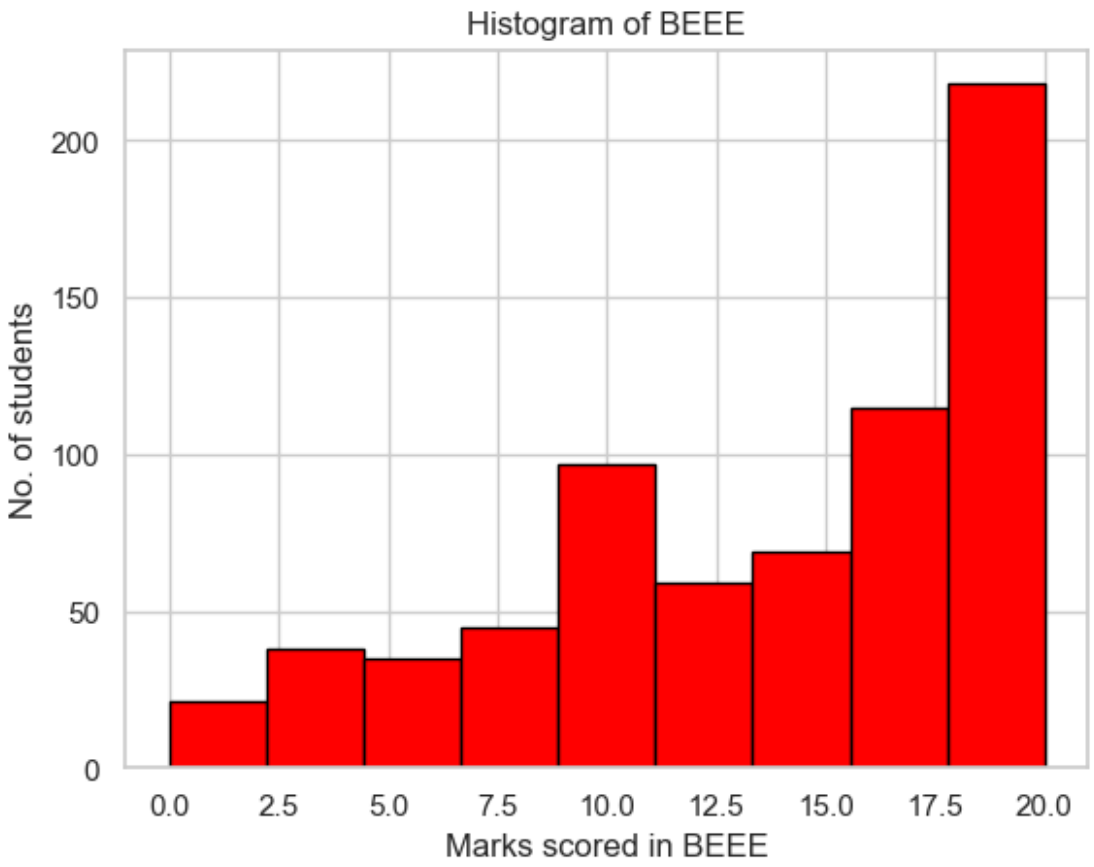
```
failures = df[df['PP'] < passing_marks]
num_failures = len(failures)

print(f"Number of students who failed in DV: {num_failures}")
```




```
Number of students who failed in DV: 198
```

```
import pandas as pd
import matplotlib.pyplot as plt



df = pd.read_excel("MIDMARKS.XLSX")


df['BEEE'] = pd.to_numeric(df['BEEE'], errors='coerce')


df = df.dropna(subset=['BEEE'])

# Plot histogram
plt.hist(df['BEEE'], color='red', edgecolor='black', bins=9)
plt.xlabel("Marks scored in BEEE")
```

```
plt.ylabel("No. of students")
plt.title("Histogram of BEEE")
plt.show()

# Calculate failures
passing_marks = 10
failures = df[df['BEEE'] < passing_marks]
num_failures = len(failures)

print(f"Number of students who failed in BEEE: {num_failures}")
```




```
Number of students who failed in BEEE: 172
```