exam-markes

March 22, 2025

1 2211cs010309

2 k.likhith

3 class marks

3.0.1 Dataset Description:

The dataset consists of student marks across multiple questions in an exam. It has **86 entries** (students) and **12 columns** representing different question scores and the total marks.

Columns Explanation: # 1. **Total** – The overall score obtained by the student. # 2. **Q1aM4** – Marks for part (a) of Question 1, out of 4. # 3. **Q1bM6** – Marks for part (b) of Question 1, out of 6. # 4. **Q2aM6** – Marks for part (a) of Question 2, out of 6. # 5. **Q2bM4** – Marks for part (b) of Question 2, out of 4. # 6. **Q3aM5** – Marks for part (a) of Question 3, out of 5. # 7. **Q3bM5** – Marks for part (b) of Question 3, out of 5. # 8. **Q4aM3** – Marks for part (a) of Question 4, out of 3. # 9. **Q4bM7** – Marks for part (b) of Question 4, out of 7. # 10. **Q5M10** – Marks for Question 5, out of 10. # 11. **Q6aM4** – Marks for part (a) of Question 6, out of 4. # 12. **Q6bM6** – Marks for part (b) of Question 6, out of 6.

```
[2]: import pandas as pd
     a=pd.read_csv("class_marks.csv")
     import matplotlib.pyplot as plt
     a
```

```
[2]:     Total  Q1aM4  Q1bM6  Q2aM6  Q2bM4  Q3aM5  Q3bM5  Q4aM3  Q4bM7  Q5M10  \
     0      37    4.0    5.0    6.0    4.0    2.0    1.0    NaN    5.0    8.0
     1      32    4.0    3.0    4.0    3.0    NaN    NaN    3.0    6.0    9.0
     2      33    4.0    5.0    5.0    1.0    5.0    5.0    NaN    NaN    8.0
     3      24    4.0    6.0    6.0    3.0    2.0    2.0    NaN    NaN    NaN
     4      36    3.0    6.0    4.0    4.0    5.0    4.0    NaN    NaN   10.0
     ..    …      …      …      …      …      …      …      …      …      …
     81     32    3.0    6.0    3.0    4.0    5.0    3.0    NaN    NaN    NaN
     82     27    2.0    2.0    5.0    3.0    NaN    NaN    NaN    NaN    7.0
     83     37    4.0    6.0    6.0    2.0    NaN    NaN    NaN    NaN    9.0
     84     28    4.0    NaN    5.0    4.0    5.0    4.0    NaN    NaN    6.0
     85     29    4.0    6.0    NaN    NaN    NaN    NaN    3.0    5.0    7.0
```

```
    Q6aM4  Q6bM6
0     4.0    6.0
1     NaN    NaN
2     NaN    NaN
3     2.0    NaN
4     NaN    NaN
..    …    …
81    4.0    6.0
82    3.0    5.0
83    4.0    6.0
84    NaN    NaN
85    1.0    4.0

[86 rows x 12 columns]
```

Dataset Description:

The dataset contains 86 entries and 12 columns. The “Total” column seems to represent the total marks. Other columns represent marks for different questions, with scores ranging across different maximum marks. ***Missing Values: columns with missing values: Q3aM5 and Q3bM5 (almost half missing) Q4aM3, Q4bM7, Q6aM4, Q6bM6 (significant missing values)

[5]:
```
a.shape
```

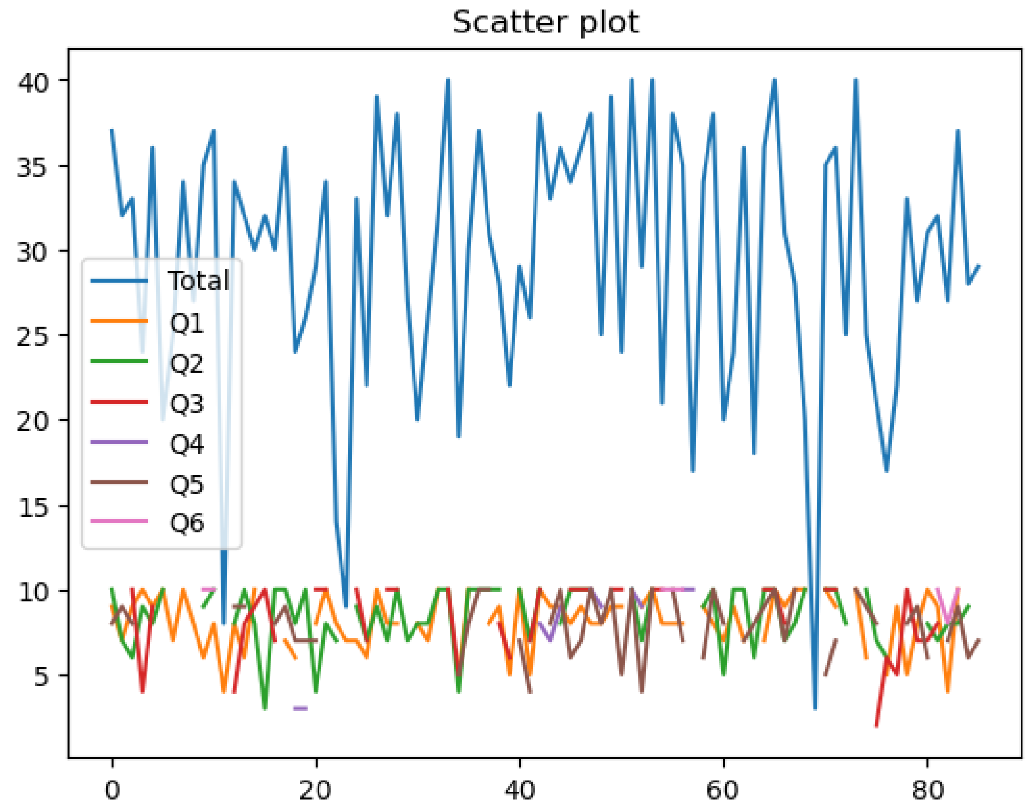
[5]:
```
(86, 12)
```

[6]:
```
a["Q1"] = a["Q1aM4"] + a["Q1bM6"]
a["Q2"] = a["Q2aM6"] + a["Q2bM4"]
a["Q3"] = a["Q3aM5"] + a["Q3bM5"]
a["Q4"] = a["Q4aM3"] + a["Q4bM7"]
a["Q5"] = a["Q5M10"]
a["Q6"] = a["Q6aM4"] + a["Q6bM6"]
a.drop(["Q1aM4", "Q1bM6","Q2aM6","Q2bM4" ,"Q3aM5" , "Q3bM5",␣
 ↪"Q4aM3","Q4bM7","Q5M10","Q6aM4","Q6bM6"], axis=1, inplace=True)
a
```

[6]:
```
    Total    Q1    Q2    Q3   Q4    Q5    Q6
0      37   9.0  10.0   3.0  NaN   8.0  10.0
1      32   7.0   7.0   NaN  9.0   9.0   NaN
2      33   9.0   6.0  10.0  NaN   8.0   NaN
3      24  10.0   9.0   4.0  NaN   NaN   NaN
4      36   9.0   8.0   9.0  NaN  10.0   NaN
..    …   …   …   …  …   …   …
81     32   9.0   7.0   8.0  NaN   NaN  10.0
82     27   4.0   8.0   NaN  NaN   7.0   8.0
83     37  10.0   8.0   NaN  NaN   9.0  10.0
84     28   NaN   9.0   9.0  NaN   6.0   NaN
85     29  10.0   NaN   NaN  8.0   7.0   5.0
```

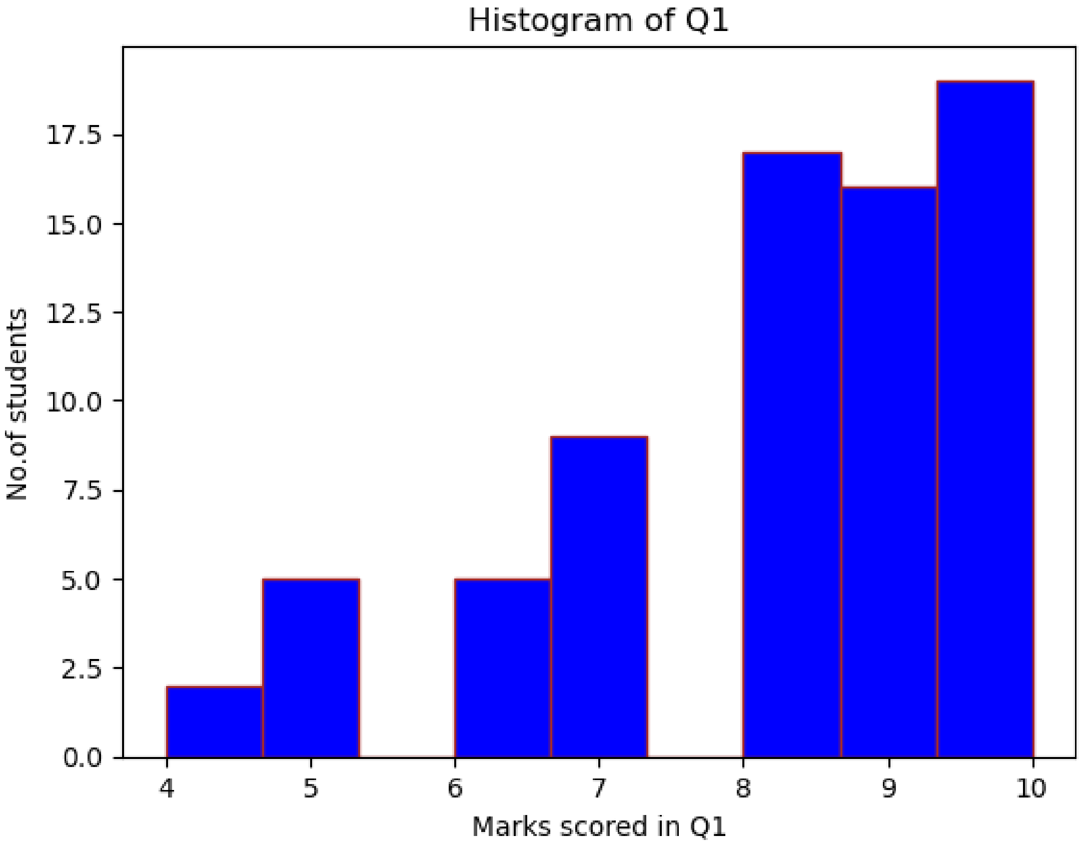
```
[86 rows x 7 columns]
```

[17]:
```
a.plot()
plt.title('Scatter plot')
plt.show()
```



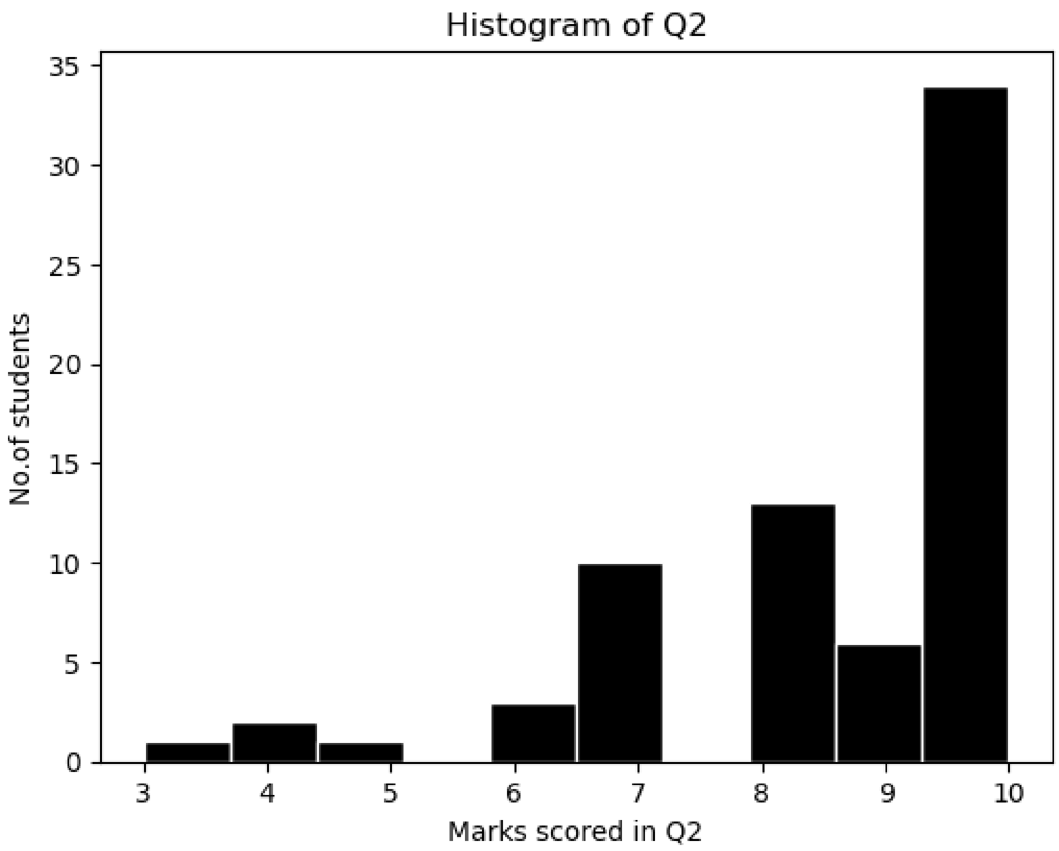
[18]:
```
plt.hist(a['Q1'],color='blue',edgecolor='brown',bins=9)
plt.xlabel("Marks scored in Q1")
plt.ylabel("No.of students")
plt.title("Histogram of Q1")
```

[18]:
```
Text(0.5, 1.0, 'Histogram of Q1')
```



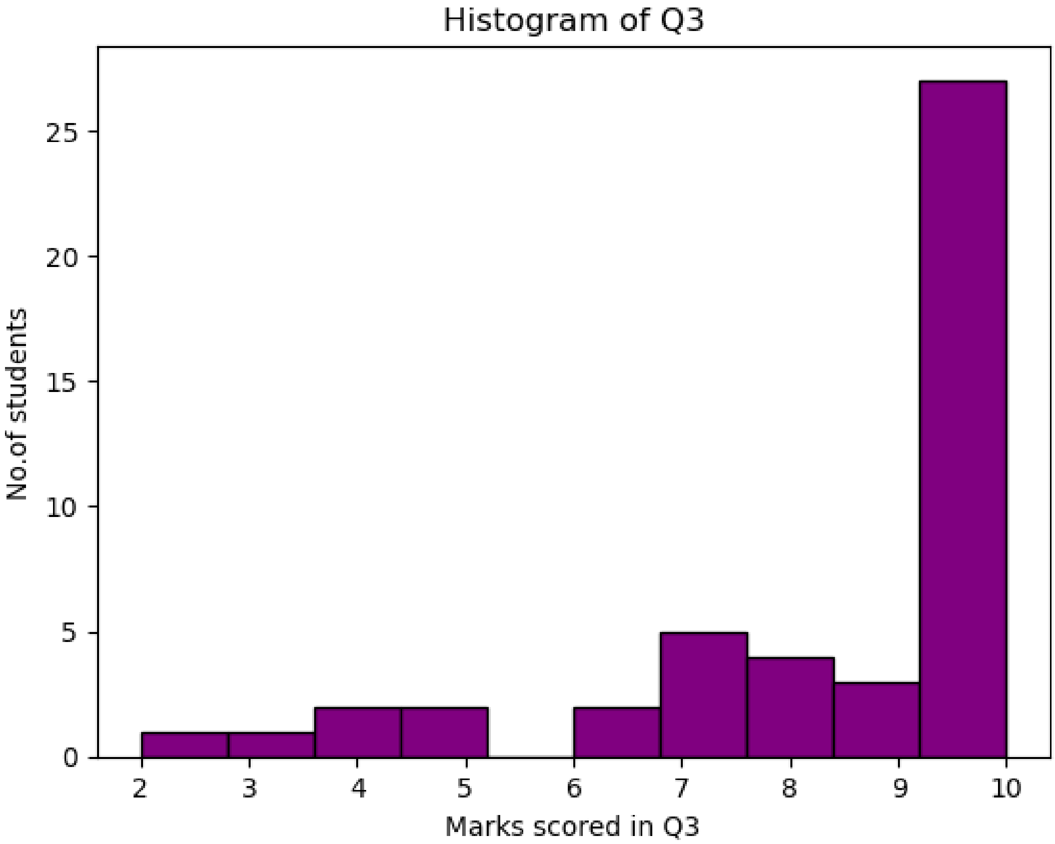
[21]:
```
plt.hist(a["Q2"],color='black',bins=10,edgecolor='white')
plt.xlabel("Marks scored in Q2")
plt.ylabel("No.of students")
plt.title("Histogram of Q2")
```

[21]:
```
Text(0.5, 1.0, 'Histogram of Q2')
```



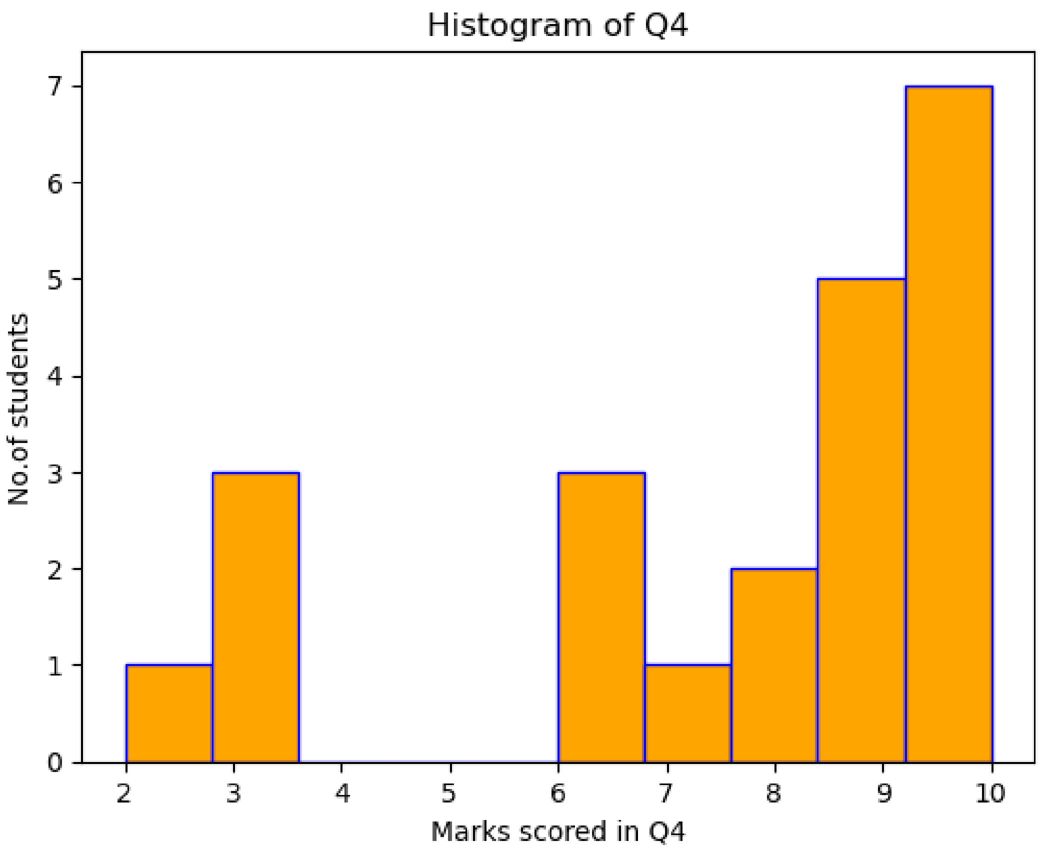
```
[23]: plt.hist(a["Q3"],color='purple',bins=10,edgecolor='black')
      plt.xlabel("Marks scored in Q3")
      plt.ylabel("No.of students")
      plt.title("Histogram of Q3")
```

```
[23]: Text(0.5, 1.0, 'Histogram of Q3')
```



```
[25]: plt.hist(a["Q4"],color='orange',bins=10,edgecolor='blue')
      plt.xlabel("Marks scored in Q4")
      plt.ylabel("No.of students")
      plt.title("Histogram of Q4")
```

```
[25]: Text(0.5, 1.0, 'Histogram of Q4')
```

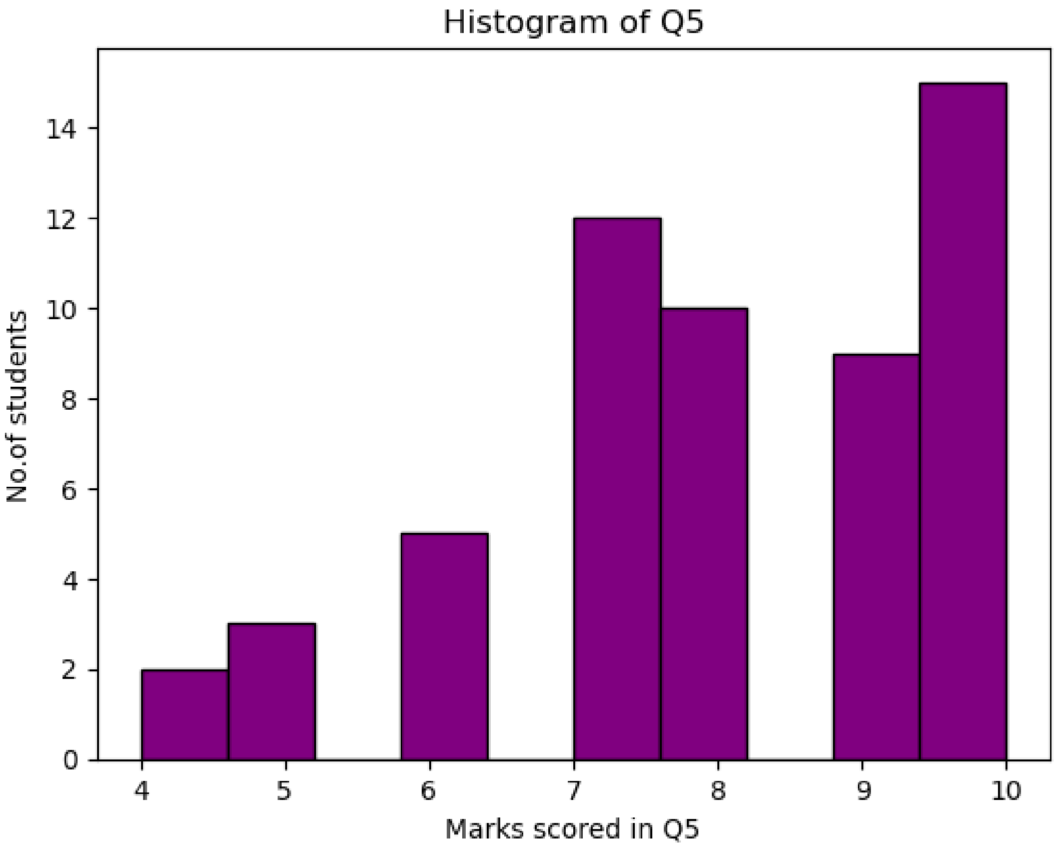


[27]:

```
plt.hist(a["Q5"],color='purple',bins=10,edgecolor='black')
plt.xlabel("Marks scored in Q5")
plt.ylabel("No.of students")
plt.title("Histogram of Q5")
```

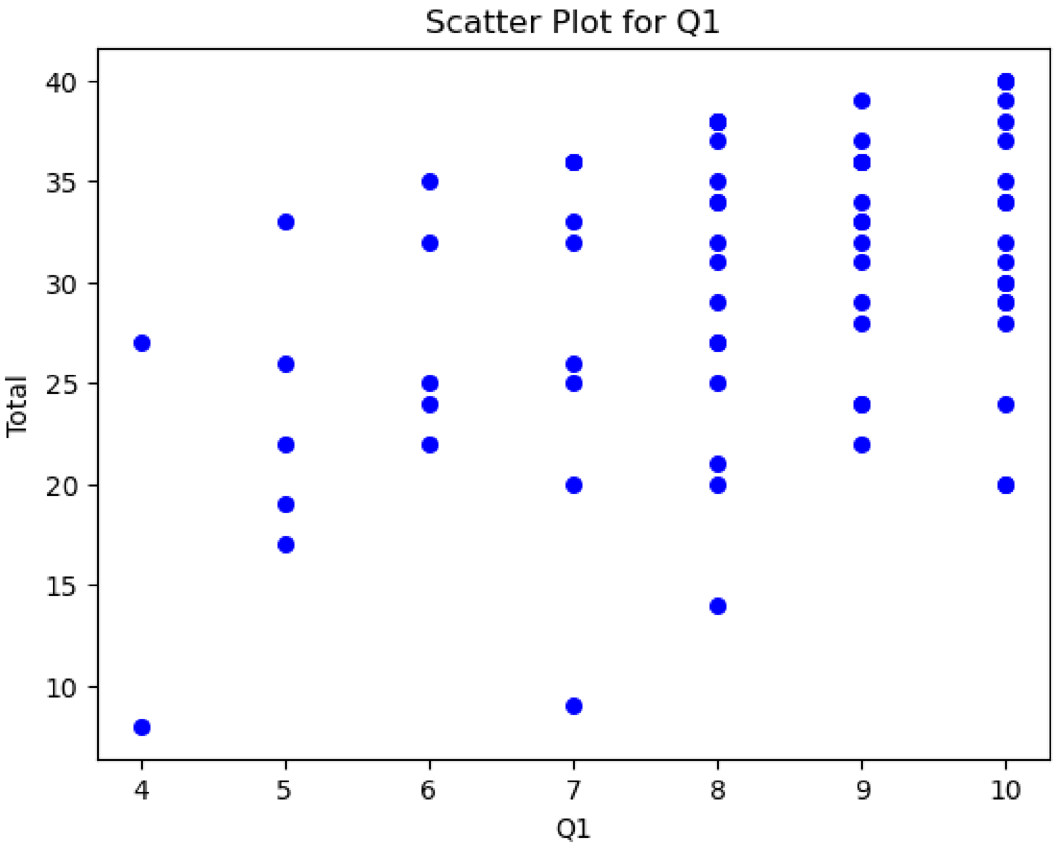
[27]:

```
Text(0.5, 1.0, 'Histogram of Q5')
```



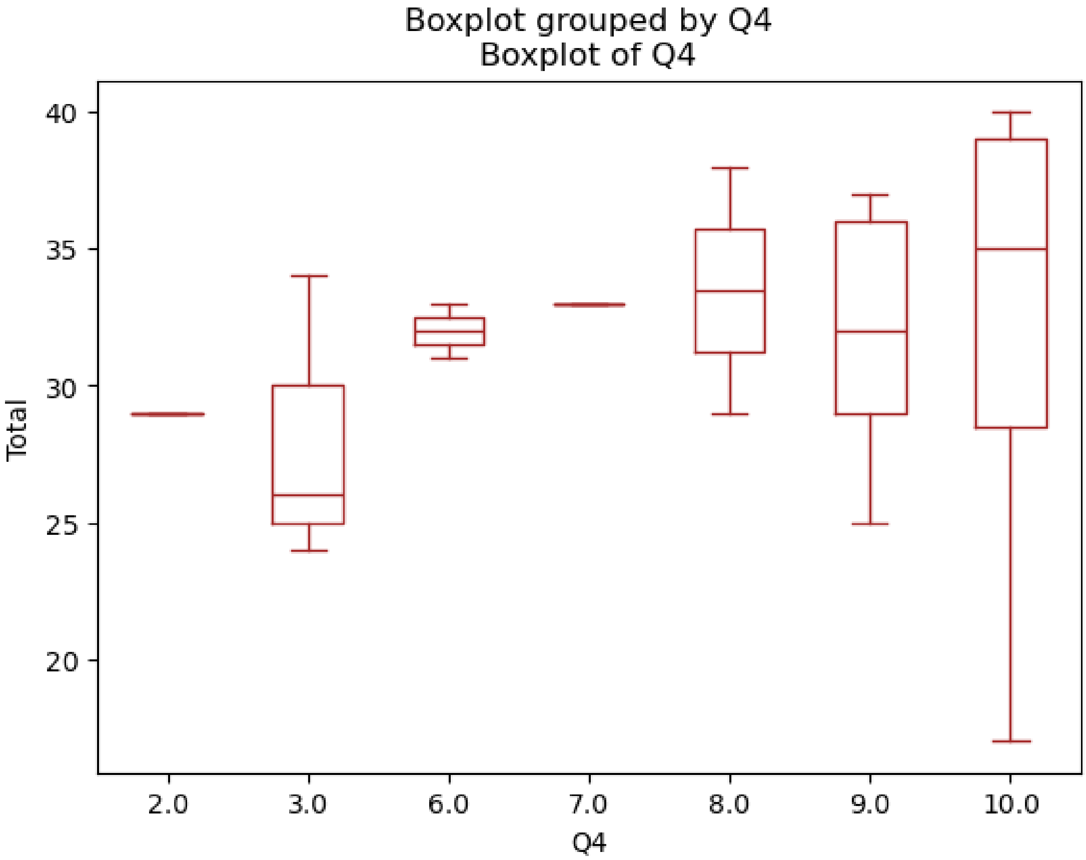
```
[29]: a.plot.scatter(x = 'Q1', y = 'Total',color='blue',s=30)
      plt.title("Scatter Plot for Q1")
```

```
[29]: Text(0.5, 1.0, 'Scatter Plot for Q1')
```

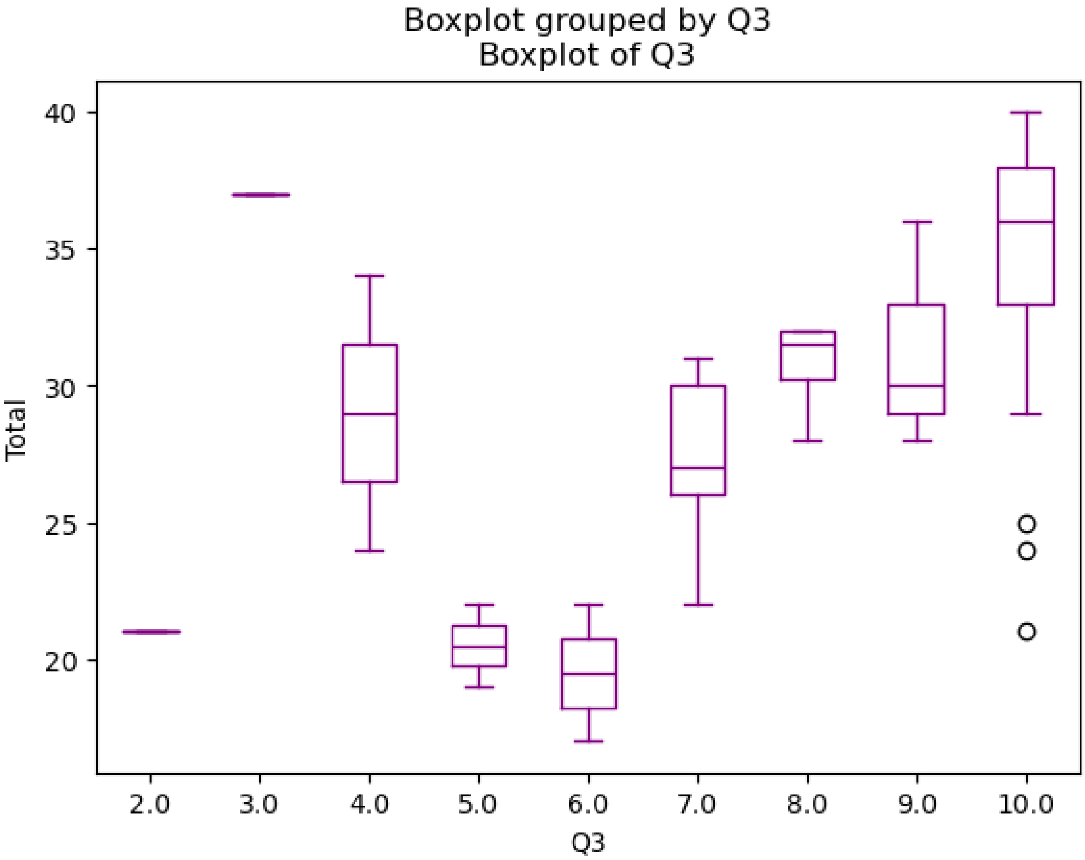


[31]:

```
a.boxplot(by='Q4', column =['Total'], grid = False,color='brown')
plt.title("Boxplot of Q4")
plt.ylabel("Total")
plt.show()
```

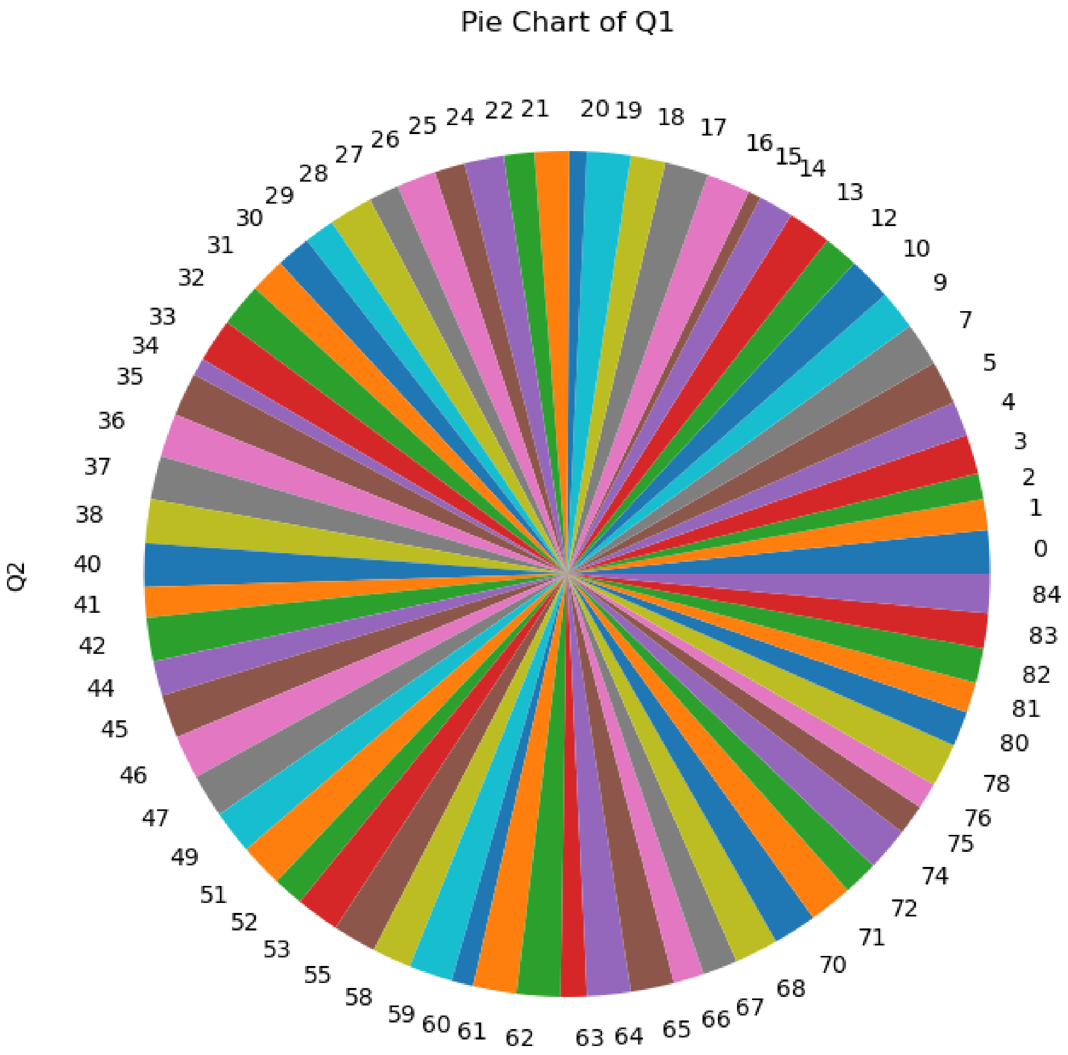


```
[33]: a.boxplot(by='Q3', column =['Total'], grid = False,color='purple')
      plt.title("Boxplot of Q3")
      plt.ylabel("Total")
      plt.show()
```



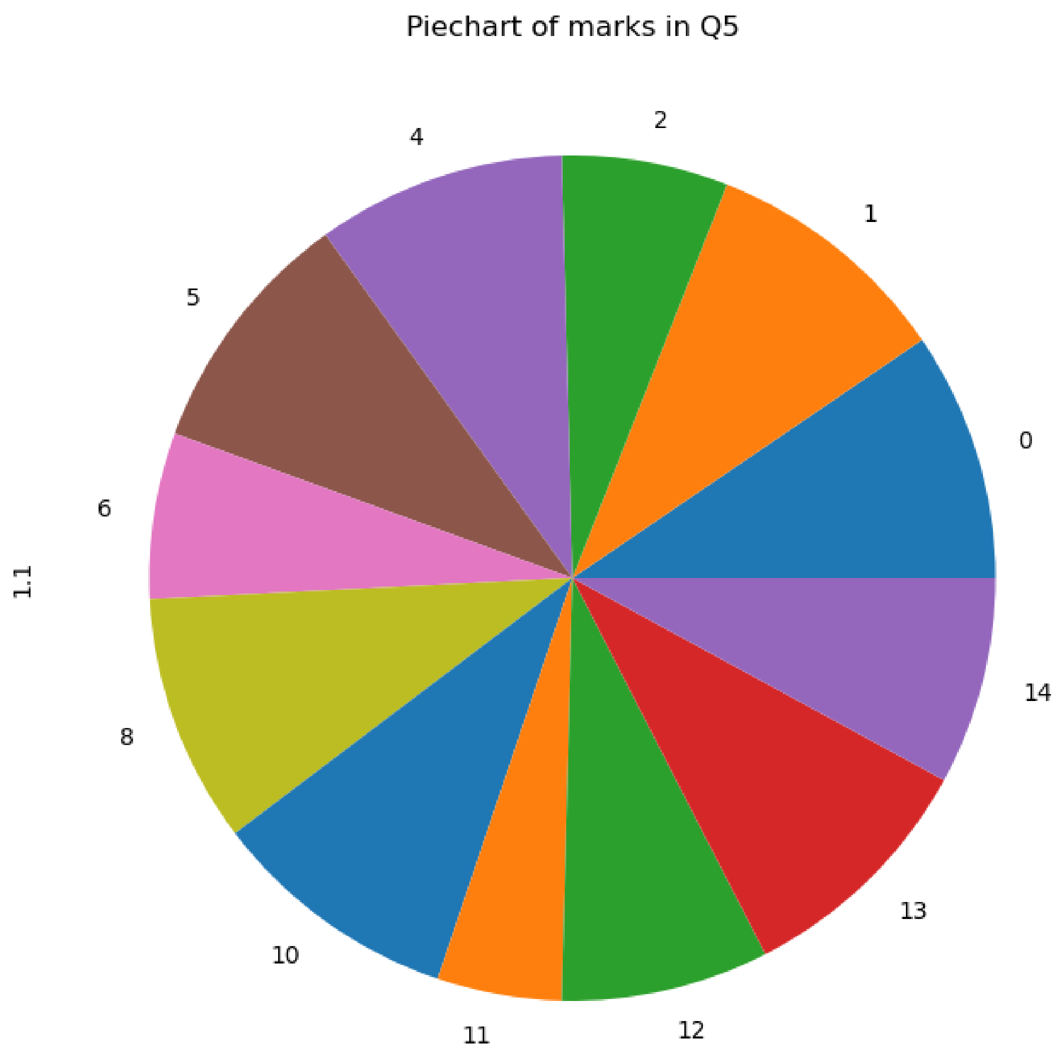
```
[35]: a["Q2"].plot(kind='pie',subplots=True,figsize=(8,10))
      plt.title("Pie Chart of Q1")
```

```
[35]: Text(0.5, 1.0, 'Pie Chart of Q1')
```



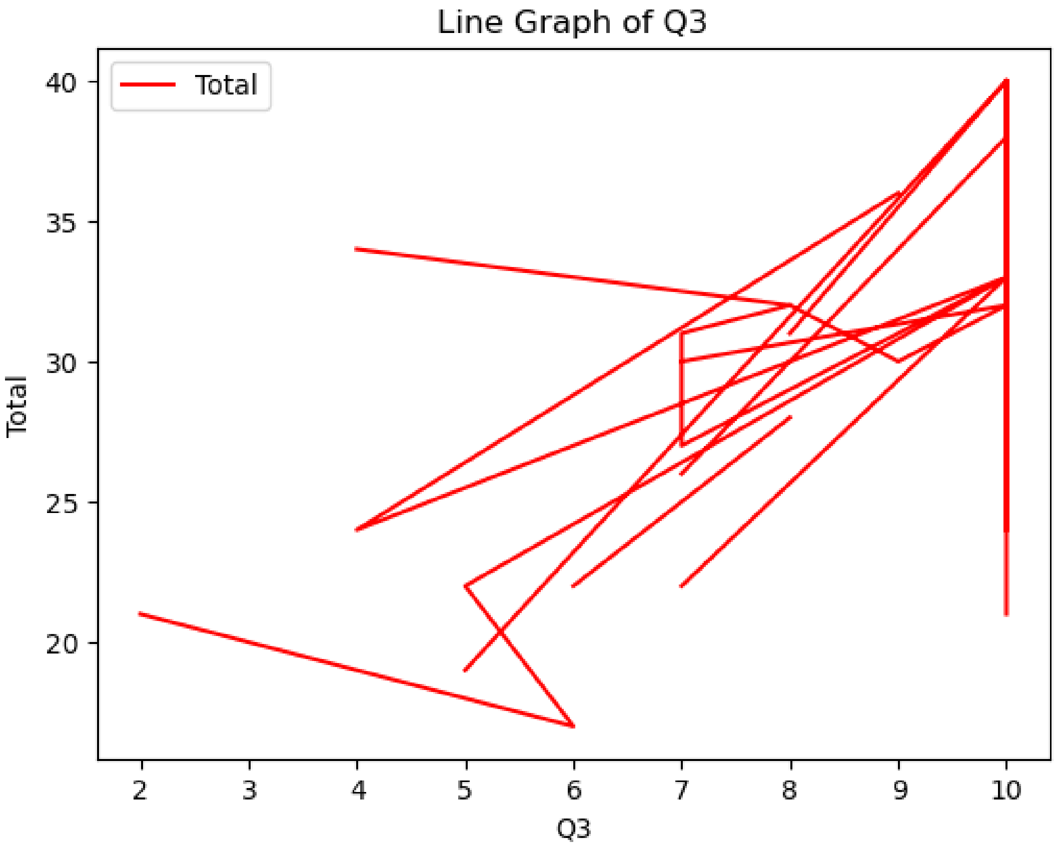
```
[37]: n=pd.read_csv("class_marks.csv",skiprows=70)
      n
      n['1.1'].plot(kind='pie',figsize=(8,10))
      plt.title("Piechart of marks in Q5")
```

```
[37]: Text(0.5, 1.0, 'Piechart of marks in Q5')
```



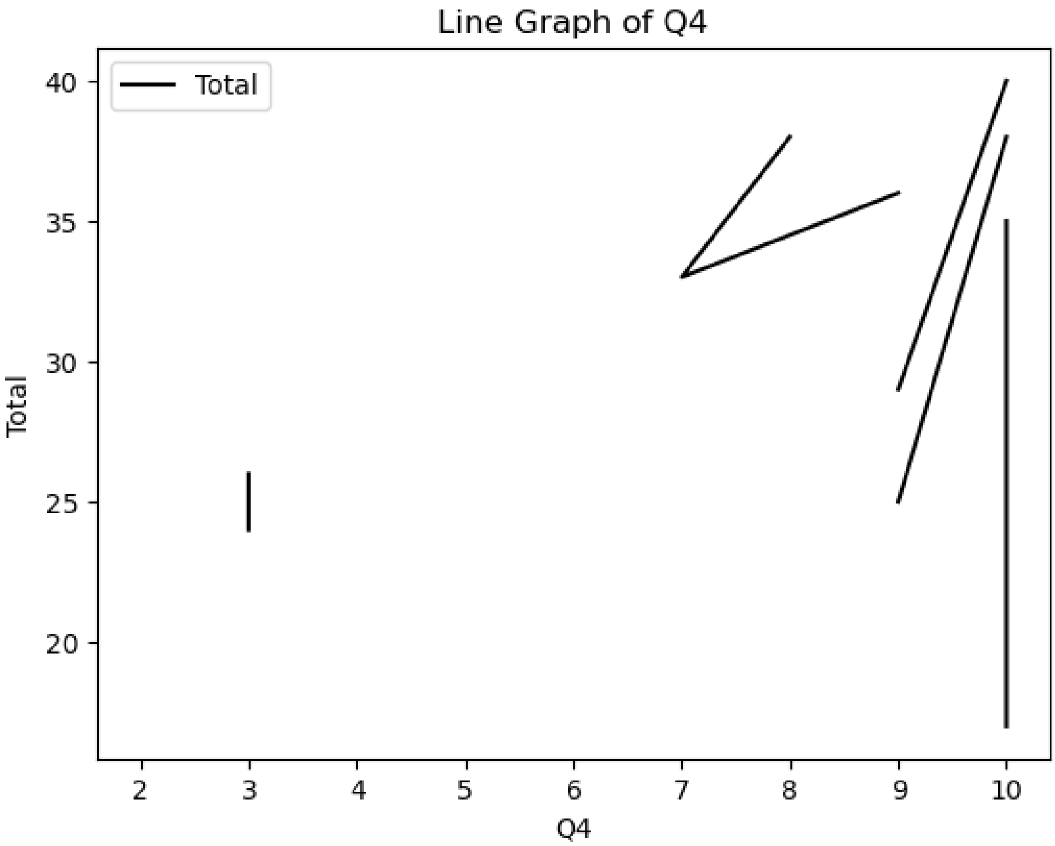
```
[39]: a.plot.line(x='Q3',y='Total',color='red')
      plt.title("Line Graph of Q3")
      plt.ylabel("Total")
```

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



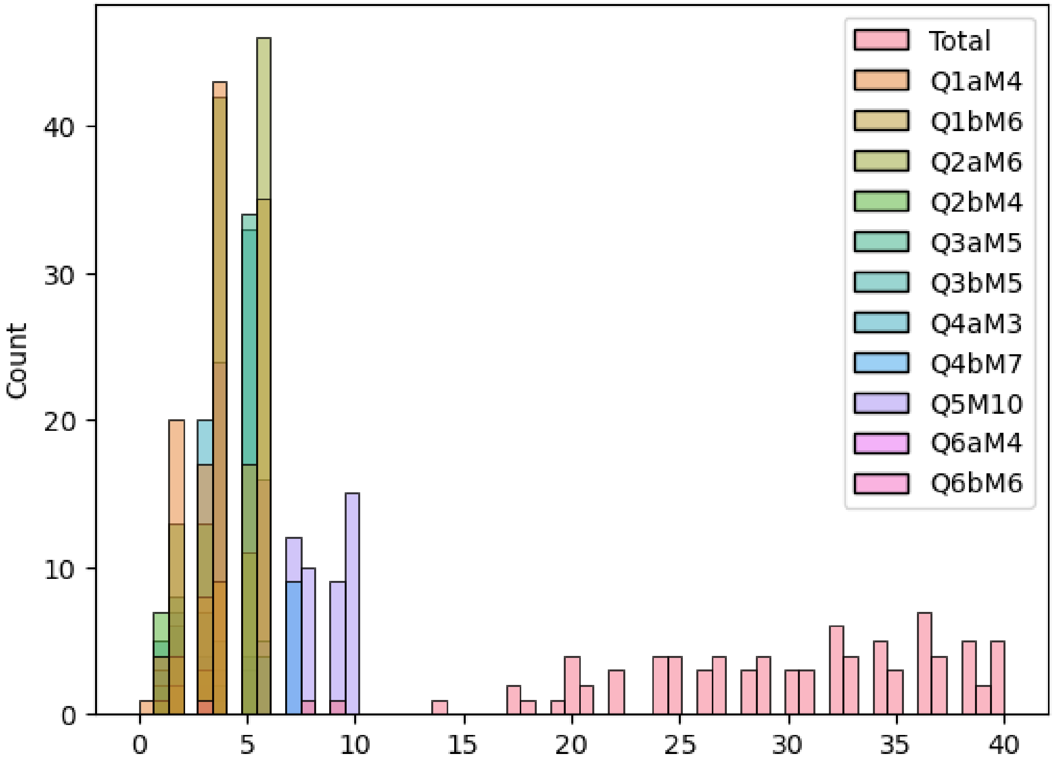
```
[41]: a.plot.line(x='Q4',y='Total',color='black')
      plt.title("Line Graph of Q4")
      plt.ylabel("Total")
```

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



```
[46]: import pandas as pd
      import seaborn as sns
      data = pd.read_csv("class_marks.csv")
      sns.histplot(data)
```

```
[46]: <Axes: ylabel='Count'>
```



[47]:

```
import pandas as pd

file_path = "class_marks.csv"
df = pd.read_csv(file_path)
df.fillna(0, inplace=True)


max_marks = 50


df['Percentage'] = (df['Total'] / max_marks) * 50


df['Percentage'] = df['Percentage'].round(2)

def assign_grade(percentage):
    if percentage >= 40:
        return 'A'
    elif percentage >= 30:
        return 'B'
    elif percentage >= 20:
        return 'C'
    elif percentage >= 100:
        return 'D'
    else:
        return 'F'


df['Grade'] = df['Percentage'].apply(assign_grade)


print(df)
```

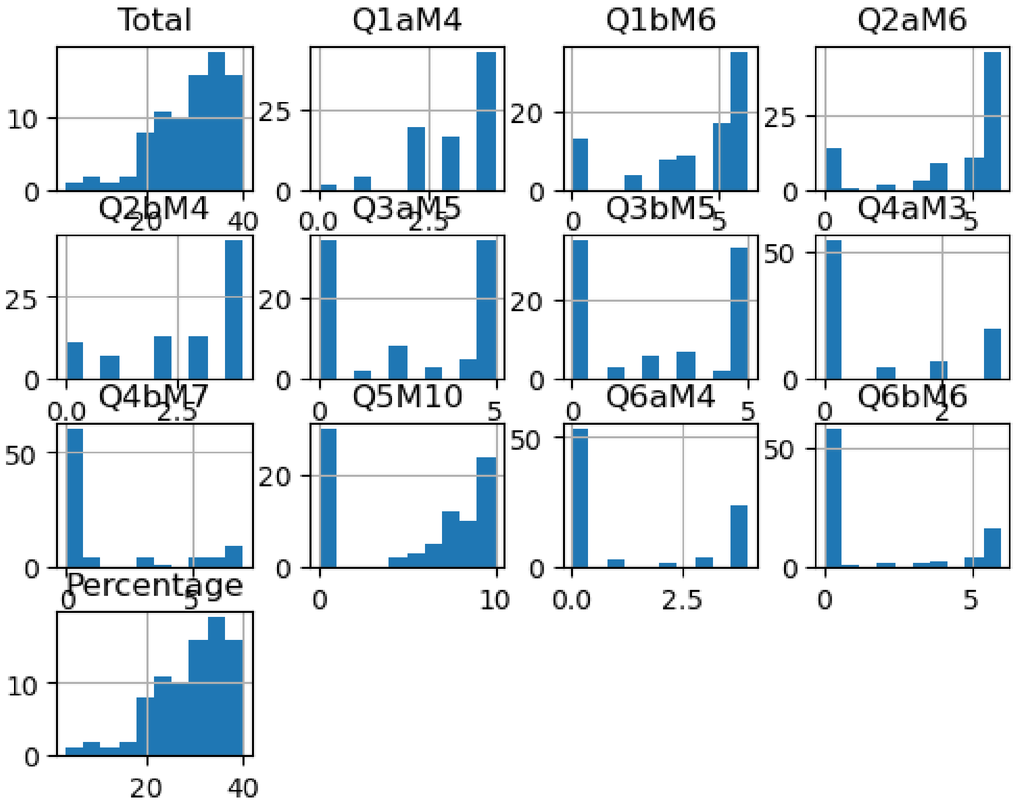
```
    Total  Q1aM4  Q1bM6  Q2aM6  Q2bM4  Q3aM5  Q3bM5  Q4aM3  Q4bM7  Q5M10  \
0      37    4.0    5.0    6.0    4.0    2.0    1.0    0.0    5.0    8.0
1      32    4.0    3.0    4.0    3.0    0.0    0.0    3.0    6.0    9.0
2      33    4.0    5.0    5.0    1.0    5.0    5.0    0.0    0.0    8.0
3      24    4.0    6.0    6.0    3.0    2.0    2.0    0.0    0.0    0.0
4      36    3.0    6.0    4.0    4.0    5.0    4.0    0.0    0.0   10.0
..    …    …    …    …    …    …    …    …    …    …
81     32    3.0    6.0    3.0    4.0    5.0    3.0    0.0    0.0    0.0
82     27    2.0    2.0    5.0    3.0    0.0    0.0    0.0    0.0    7.0
83     37    4.0    6.0    6.0    2.0    0.0    0.0    0.0    0.0    9.0
84     28    4.0    0.0    5.0    4.0    5.0    4.0    0.0    0.0    6.0
85     29    4.0    6.0    0.0    0.0    0.0    0.0    3.0    5.0    7.0

    Q6aM4  Q6bM6  Percentage Grade
0     4.0    6.0        37.0     B
1     0.0    0.0        32.0     B
2     0.0    0.0        33.0     B
3     2.0    0.0        24.0     C
4     0.0    0.0        36.0     B
..    …    …         …   …
81    4.0    6.0        32.0     B
82    3.0    5.0        27.0     C
83    4.0    6.0        37.0     B
84    0.0    0.0        28.0     C
85    1.0    4.0        29.0     C

[86 rows x 14 columns]
```

[50]:

```
df.hist()
```

[50]:

```
array([[<Axes: title={'center': 'Total'}>,
        <Axes: title={'center': 'Q1aM4'}>,
        <Axes: title={'center': 'Q1bM6'}>,
        <Axes: title={'center': 'Q2aM6'}>],
       [<Axes: title={'center': 'Q2bM4'}>,
        <Axes: title={'center': 'Q3aM5'}>,
        <Axes: title={'center': 'Q3bM5'}>,
        <Axes: title={'center': 'Q4aM3'}>],
       [<Axes: title={'center': 'Q4bM7'}>,
        <Axes: title={'center': 'Q5M10'}>,
        <Axes: title={'center': 'Q6aM4'}>,
        <Axes: title={'center': 'Q6bM6'}>],
       [<Axes: title={'center': 'Percentage'}>, <Axes: >, <Axes: >,
        <Axes: >]], dtype=object)
```



4 The dataset reveals student performance in an exam, with scores ranging from 3 to 40 and an average of 29.36. Most students scored between 25 and 36, but some struggled significantly. Certain questions, like Q4bM7 and Q6bM6, were skipped by many, suggesting they were difficult or less prioritized. Widely attempted questions, such as Q1aM4 and Q2aM6, had higher average scores, indicating they were easier. The data highlights performance gaps, suggesting the need for targeted academic support. Understanding these trends can help educators improve teaching strategies and exam design.

[ ]:

```

```