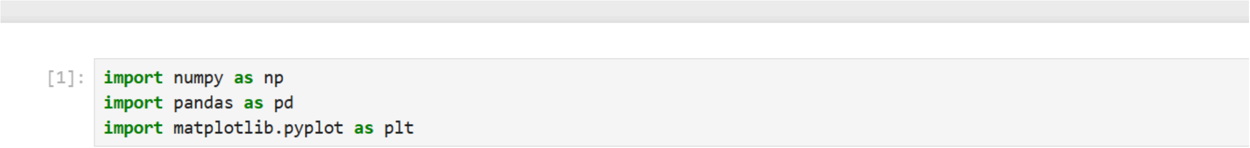
# Task 4

Submitted by  : Ghulam Mustafa

**Basic Data analysis and EDA in python** 🙂

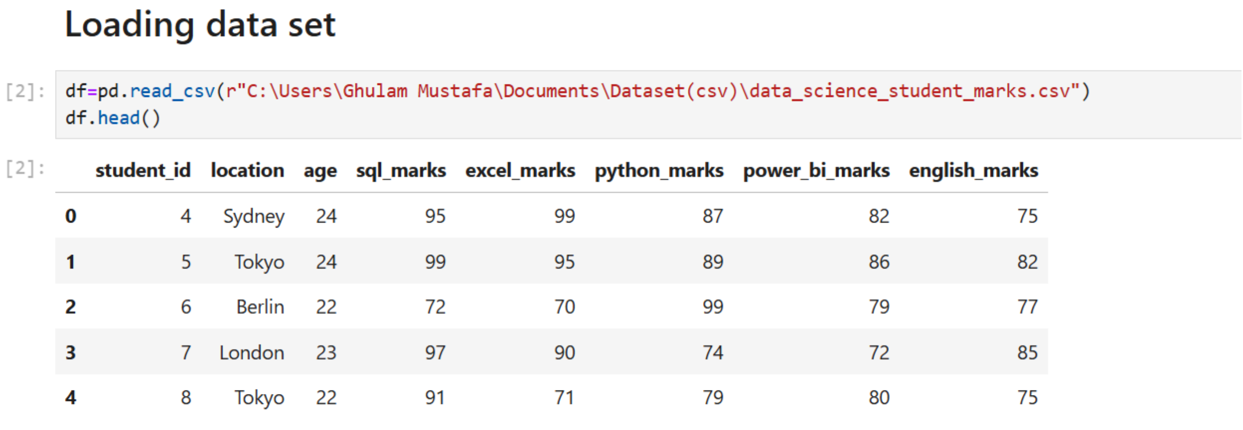
**Data set overview** : I downloaded this data set from Kaggle , it includes the marks students of Data science department the data set has 8 columns and 5 hundred rows .

1.Importing libraries 👍



```
[1]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
```

2. Loading data in Pandas using pd.read_csv()



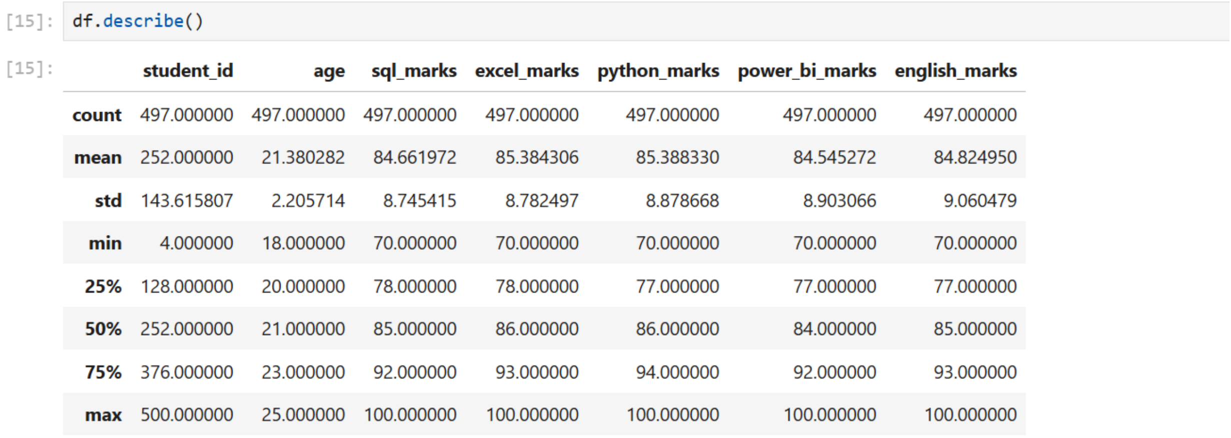
## Loading data set

```
[2]: df=pd.read_csv(r"C:\Users\Ghulam Mustafa\Documents\Dataset(csv)\data_science_student_marks.csv")
     df.head()
```

[2]:

| | student_id | location | age | sql_marks | excel_marks | python_marks | power_bi_marks | english_marks |
|---|---|---|---|---|---|---|---|---|
| **0** | 4 | Sydney | 24 | 95 | 99 | 87 | 82 | 75 |
| **1** | 5 | Tokyo | 24 | 99 | 95 | 89 | 86 | 82 |
| **2** | 6 | Berlin | 22 | 72 | 70 | 99 | 79 | 77 |
| **3** | 7 | London | 23 | 97 | 90 | 74 | 72 | 85 |
| **4** | 8 | Tokyo | 22 | 91 | 71 | 79 | 80 | 75 |

3. Taking Statistical overview of data

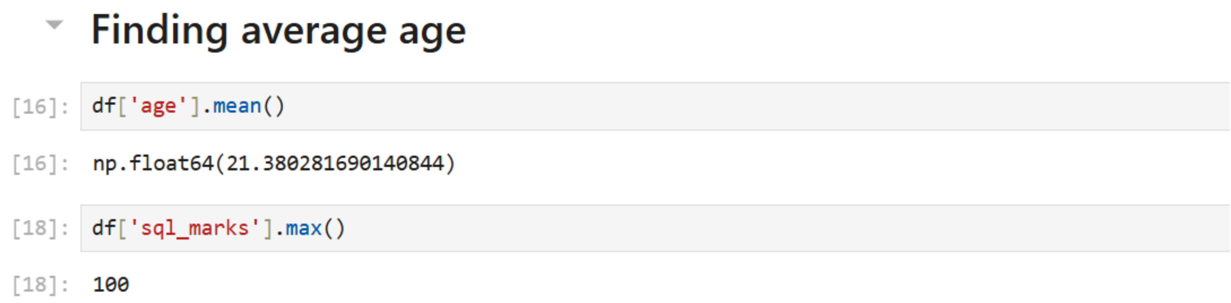


```
[15]: df.describe()
```

[15]:

| | student_id | age | sql_marks | excel_marks | python_marks | power_bi_marks | english_marks |
|---|---|---|---|---|---|---|---|
| **count** | 497.000000 | 497.000000 | 497.000000 | 497.000000 | 497.000000 | 497.000000 | 497.000000 |
| **mean** | 252.000000 | 21.380282 | 84.661972 | 85.384306 | 85.388330 | 84.545272 | 84.824950 |
| **std** | 143.615807 | 2.205714 | 8.745415 | 8.782497 | 8.878668 | 8.903066 | 9.060479 |
| **min** | 4.000000 | 18.000000 | 70.000000 | 70.000000 | 70.000000 | 70.000000 | 70.000000 |
| **25%** | 128.000000 | 20.000000 | 78.000000 | 78.000000 | 77.000000 | 77.000000 | 77.000000 |
| **50%** | 252.000000 | 21.000000 | 85.000000 | 86.000000 | 86.000000 | 84.000000 | 85.000000 |
| **75%** | 376.000000 | 23.000000 | 92.000000 | 93.000000 | 94.000000 | 92.000000 | 93.000000 |
| **max** | 500.000000 | 25.000000 | 100.000000 | 100.000000 | 100.000000 | 100.000000 | 100.000000 |

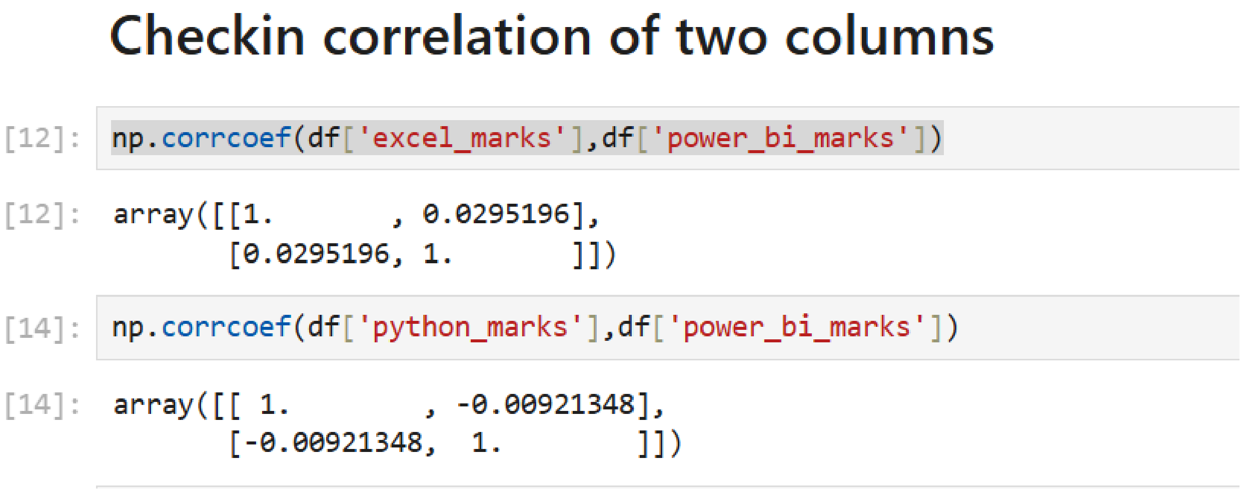
4.Find Avg of Age and Mac marks in SQL subject 👍



## Finding average age

```
[16]: df['age'].mean()

[16]: np.float64(21.380281690140844)

[18]: df['sql_marks'].max()

[18]: 100
```

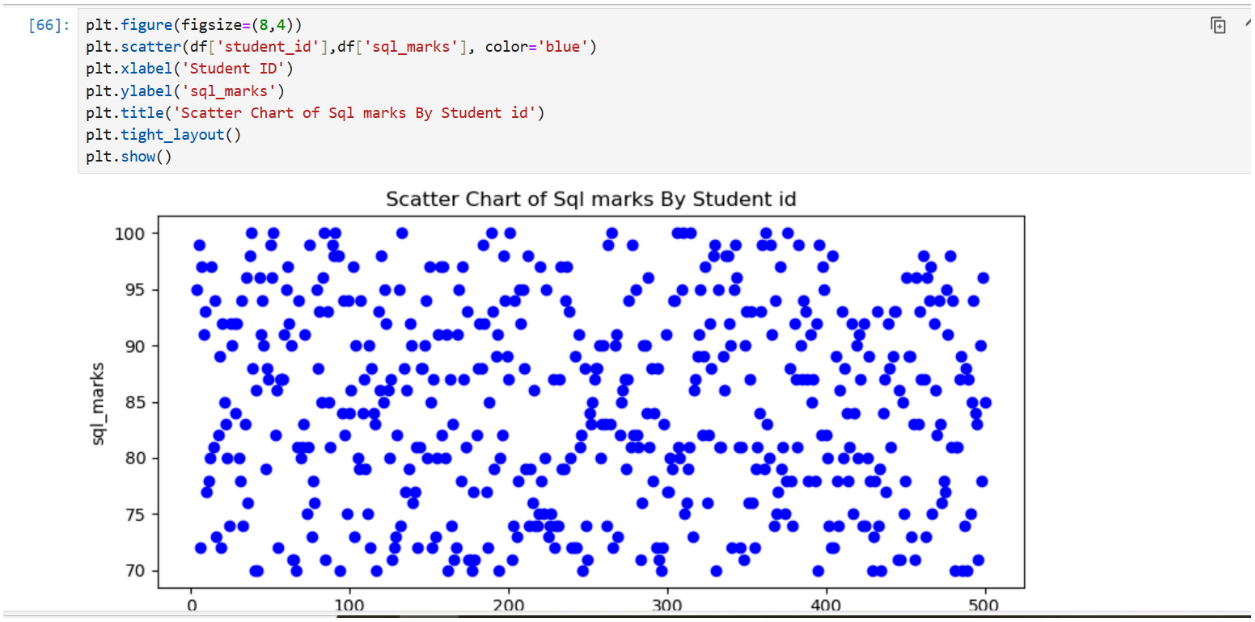
5. Finding correlation between columns 😀



## Checkin correlation of two columns

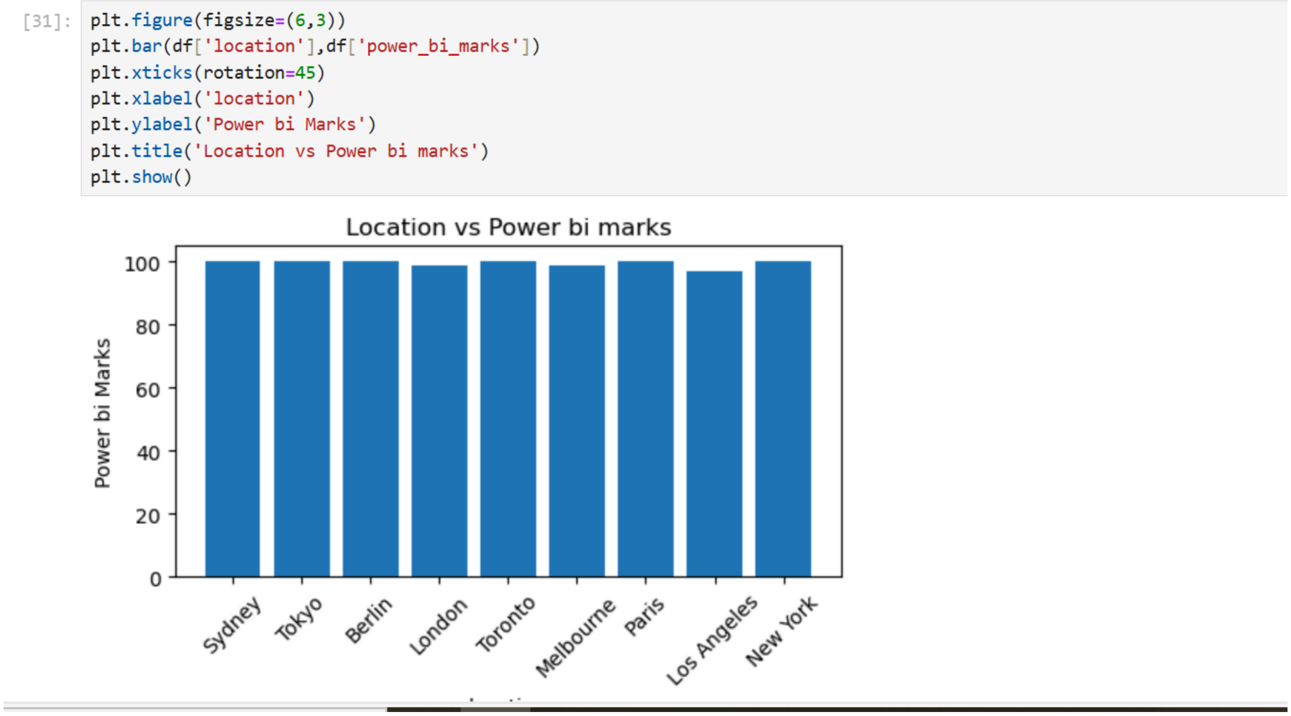
```
[12]: np.corrcoef(df['excel_marks'],df['power_bi_marks'])

[12]: array([[1.       , 0.0295196],
              [0.0295196, 1.       ]])

[14]: np.corrcoef(df['python_marks'],df['power_bi_marks'])

[14]: array([[ 1.        , -0.00921348],
              [-0.00921348,  1.        ]])
```

## 6. Scatter plot of Student id and sql marks



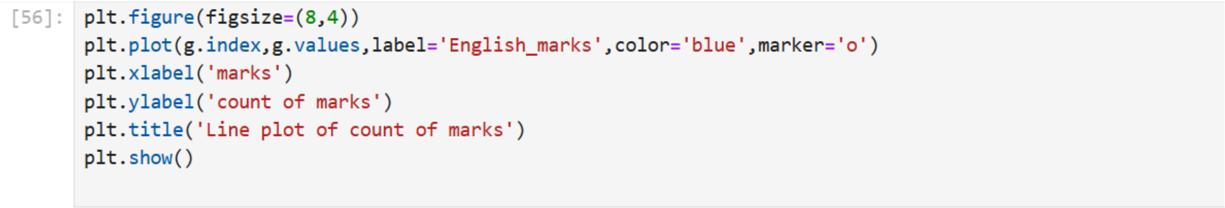
```
[66]: plt.figure(figsize=(8,4))
      plt.scatter(df['student_id'],df['sql_marks'], color='blue')
      plt.xlabel('Student ID')
      plt.ylabel('sql_marks')
      plt.title('Scatter Chart of Sql marks By Student id')
      plt.tight_layout()
      plt.show()
```

## 7. Bar Chart showing Location wise marks

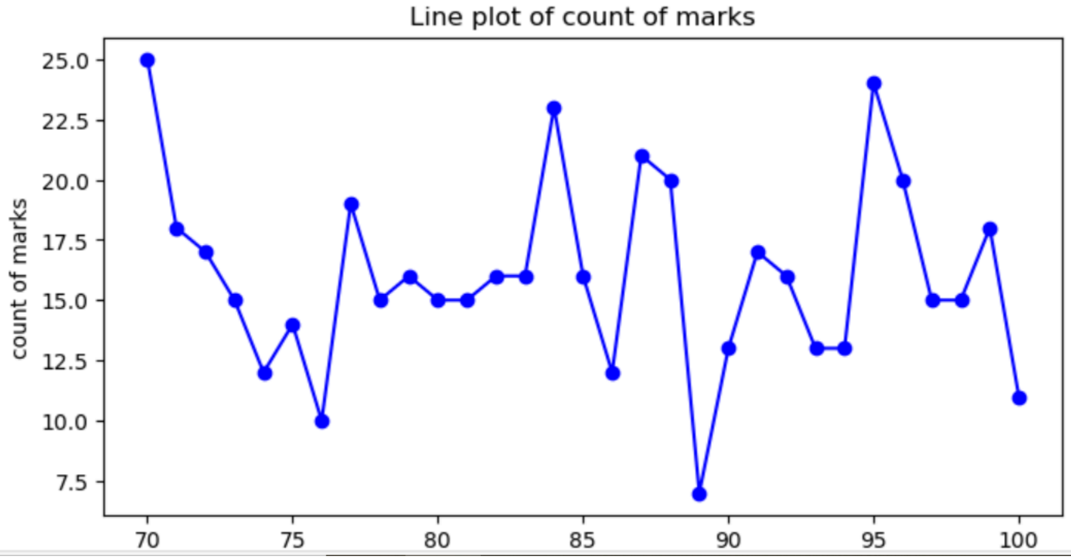


```
[31]: plt.figure(figsize=(6,3))
      plt.bar(df['location'],df['power_bi_marks'])
      plt.xticks(rotation=45)
      plt.xlabel('location')
      plt.ylabel('Power bi Marks')
      plt.title('Location vs Power bi marks')
      plt.show()
```

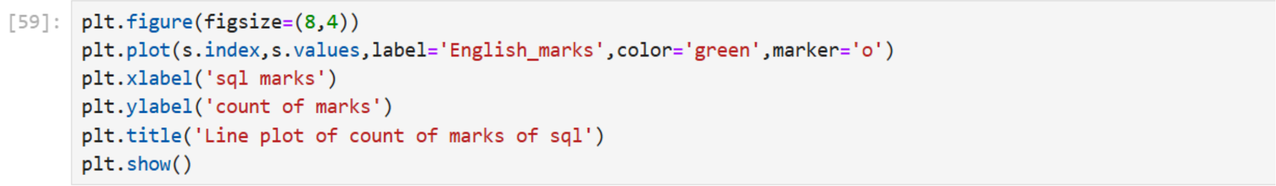
8. Line plot of English marks and their count



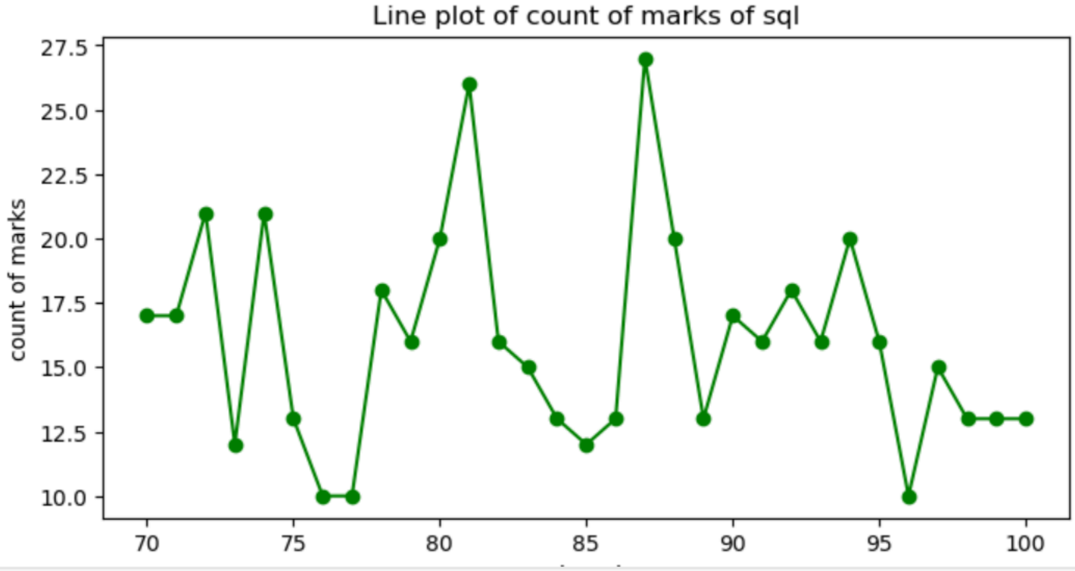
```
[56]: plt.figure(figsize=(8,4))
      plt.plot(g.index,g.values,label='English_marks',color='blue',marker='o')
      plt.xlabel('marks')
      plt.ylabel('count of marks')
      plt.title('Line plot of count of marks')
      plt.show()
```



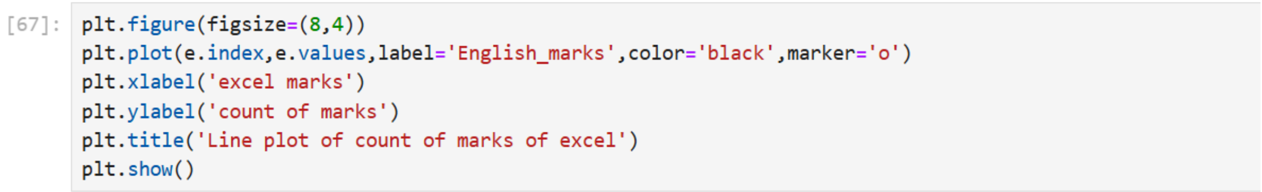
9.Line plot of SQL  marks and their count



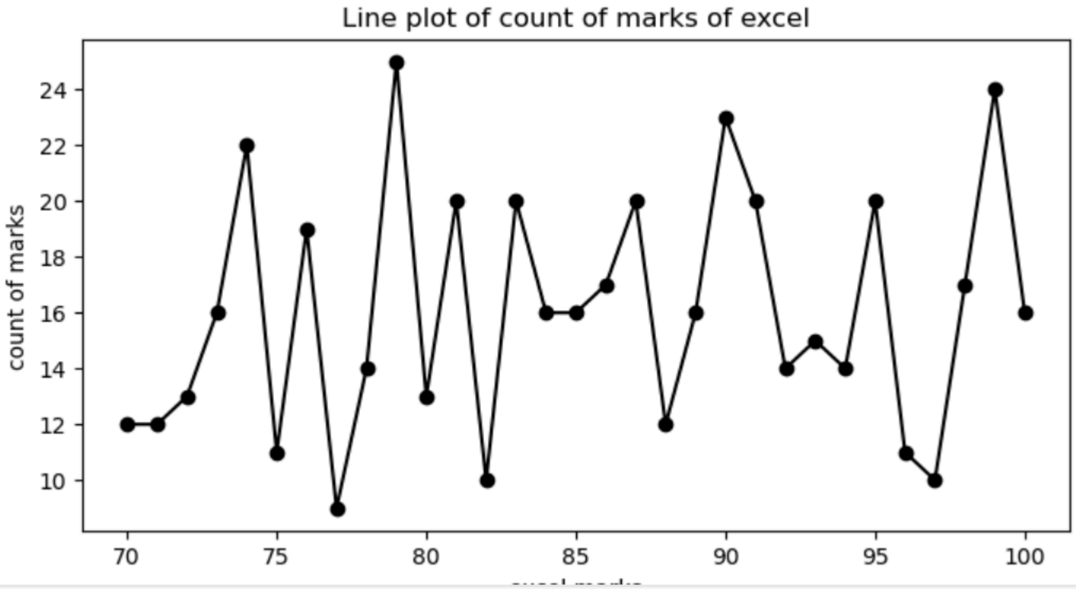
```
[59]: plt.figure(figsize=(8,4))
      plt.plot(s.index,s.values,label='English_marks',color='green',marker='o')
      plt.xlabel('sql marks')
      plt.ylabel('count of marks')
      plt.title('Line plot of count of marks of sql')
      plt.show()
```



## 9.Line plot of Excel  marks and their count 👏



```
[67]: plt.figure(figsize=(8,4))
      plt.plot(e.index,e.values,label='English_marks',color='black',marker='o')
      plt.xlabel('excel marks')
      plt.ylabel('count of marks')
      plt.title('Line plot of count of marks of excel')
      plt.show()
```


Line plot of count of marks of excel
count of marks
excel marks