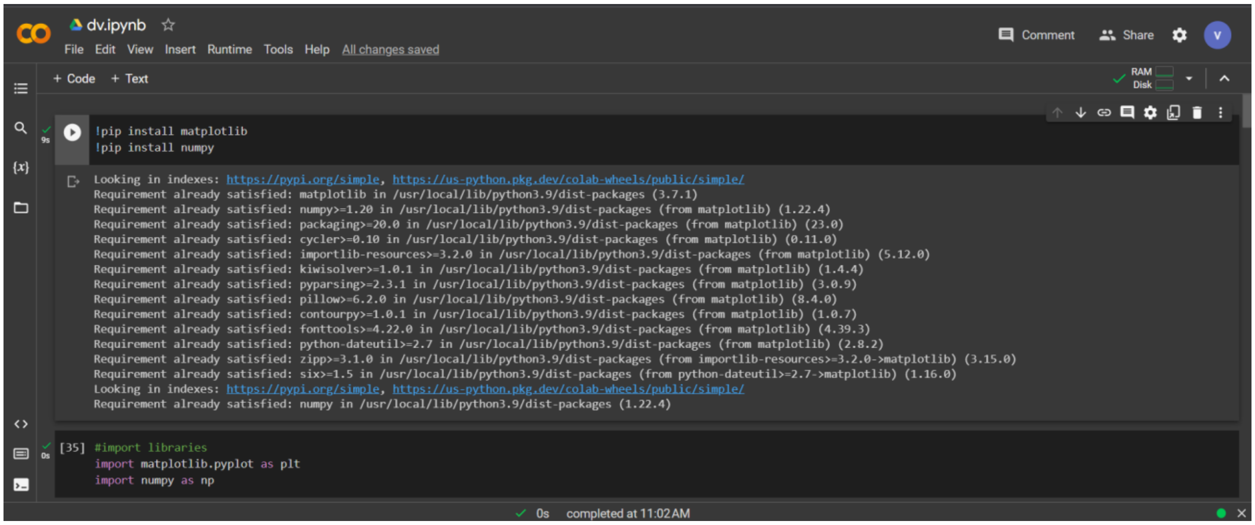
Big Data and Data Science

ICE 9

Task 1

Installed the libraries matplotlib and numpy in the google colab notebook. Imported pyplot from matplotlib as plt and numpy as np.



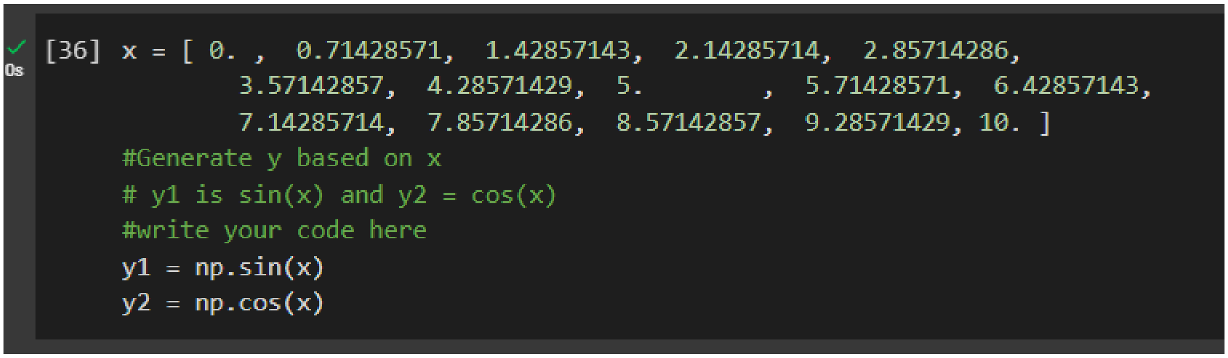
```
!pip install matplotlib
!pip install numpy
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: matplotlib in /usr/local/lib/python3.9/dist-packages (3.7.1)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.9/dist-packages (from matplotlib) (1.22.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib) (23.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.9/dist-packages (from matplotlib) (0.11.0)
Requirement already satisfied: importlib-resources>=3.2.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib) (5.12.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib) (1.4.4)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib) (3.0.9)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib) (8.4.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib) (1.0.7)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib) (4.39.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.9/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/python3.9/dist-packages (from importlib-resources>=3.2.0->matplotlib) (3.15.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages (1.22.4)
```

```
[35] #import libraries
     import matplotlib.pyplot as plt
     import numpy as np
```

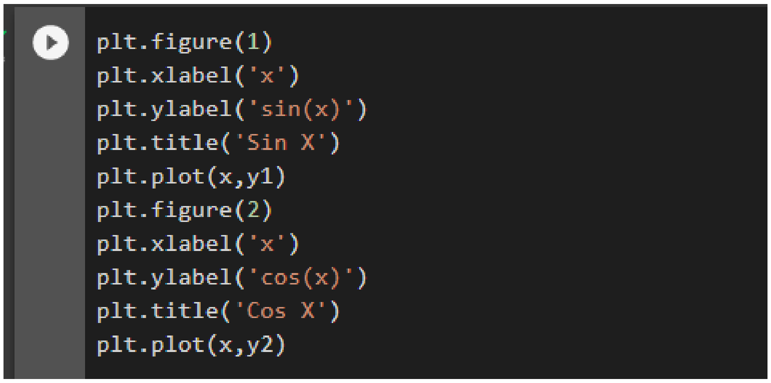
A)

Computed sin values for the given x values using numpy and assigned the result to variable y1. Similarly, computed cosine value for the given x values using numpy.



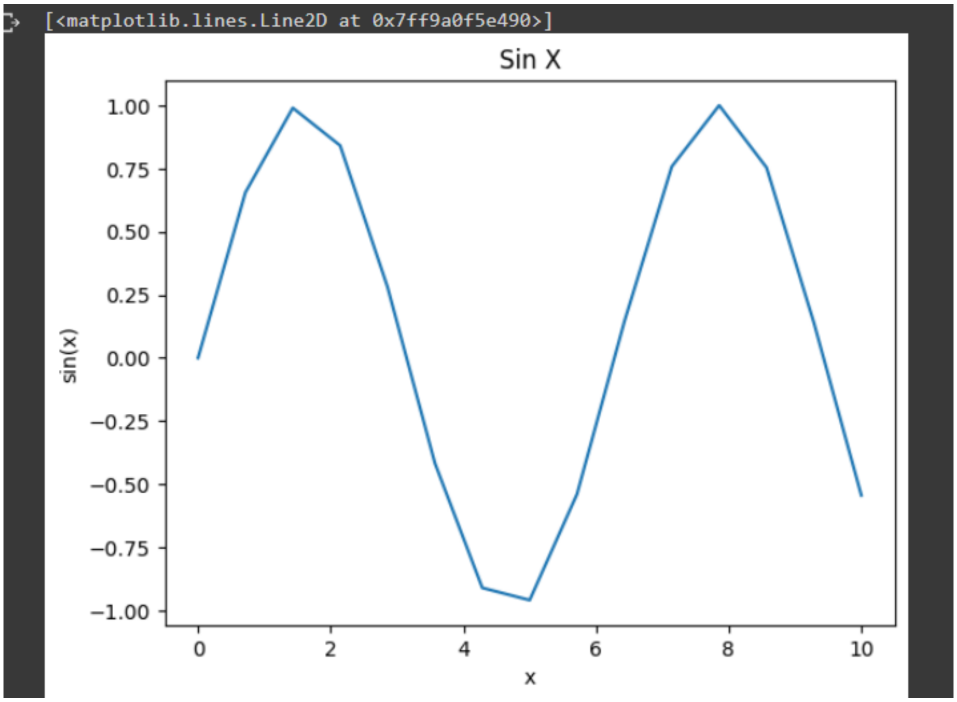
```
[36] x = [ 0.  ,  0.71428571,  1.42857143,  2.14285714,  2.85714286,
            3.57142857,  4.28571429,  5.        ,  5.71428571,  6.42857143,
            7.14285714,  7.85714286,  8.57142857,  9.28571429, 10. ]
     #Generate y based on x
     # y1 is sin(x) and y2 = cos(x)
     #write your code here
     y1 = np.sin(x)
     y2 = np.cos(x)
```

Created plots for sin(x) in figure 1 and provided title of the plot, x label and y label. And created plot for cos(x) with the title, x label and y label. The plot sin x is created with the values of x and is stored in the variable y1, and the plot of cos x is created with the values of x and is assigned to variable y2.



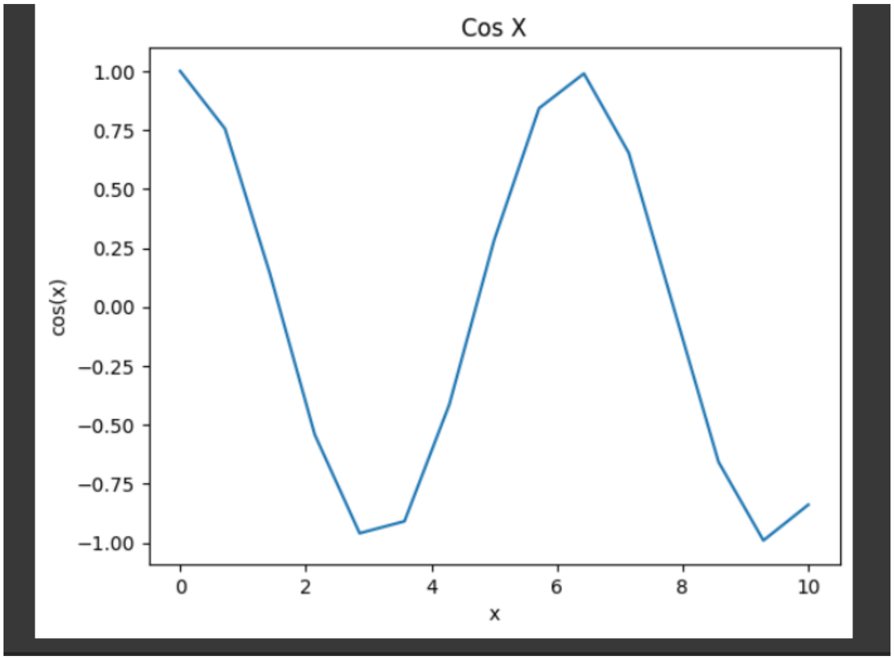
```
plt.figure(1)
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.title('Sin X')
plt.plot(x,y1)
plt.figure(2)
plt.xlabel('x')
plt.ylabel('cos(x)')
plt.title('Cos X')
plt.plot(x,y2)
```

The above figure shows the code for the plots and their labels.



```
[<matplotlib.lines.Line2D at 0x7ff9a0f5e490>]
```

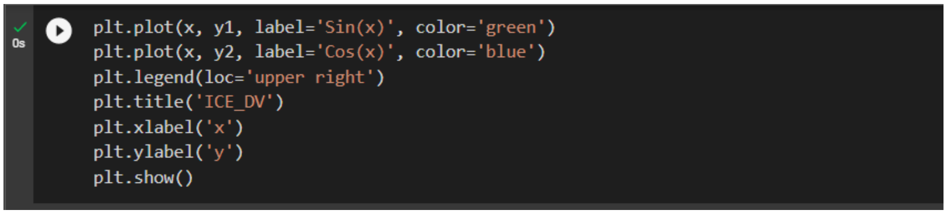
The above figure is the plot for sin x.



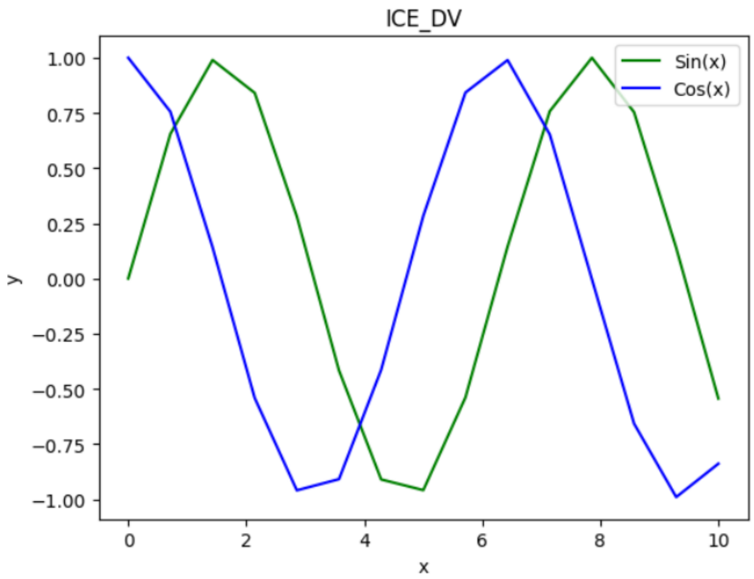
The above figure is the plot for cos x.

B & C)

In the below figure, the code for plotting the sin x and cos x in a single plot has been written. The sin x curve is in green color and the cos x curve is in blue color. The title of the plot is set as ICE_DEV and the x and y axis are labeled as x and y respectively. The legend for the plot is positioned on the top right of the plot. Legend has the information for the name of the curve and the indication for how it is identified.



```
plt.plot(x, y1, label='Sin(x)', color='green')
plt.plot(x, y2, label='Cos(x)', color='blue')
plt.legend(loc='upper right')
plt.title('ICE_DV')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



Result of the plot using above code to view sin x and cos x in a single plot.

D)

Strengths of line graph are:

- It is easy to read the data from a line graph that is built to measure the trends over a time period.
- Even a small change in the trend can be easily identified using line graph compared to other graphs.
- They are less complex and easy to read and understand.
- More than one trend or one graph can be plotted on the same axis, and this helps in comparing 2 trends that are of same kind.

Weakness of line graphs are:

- A line graph with more than 3 lines in it is hard to read and understand. It may create confusion instead of clarity.
- A line graph with huge data and huge fluctuations in the data is hard to read and analyze.
- This kind of representation can be done only for the numerical data.
- A smooth line graph cannot be obtained if the data is not appropriate.
- Plot with decimal value can be a challenging task.

E)

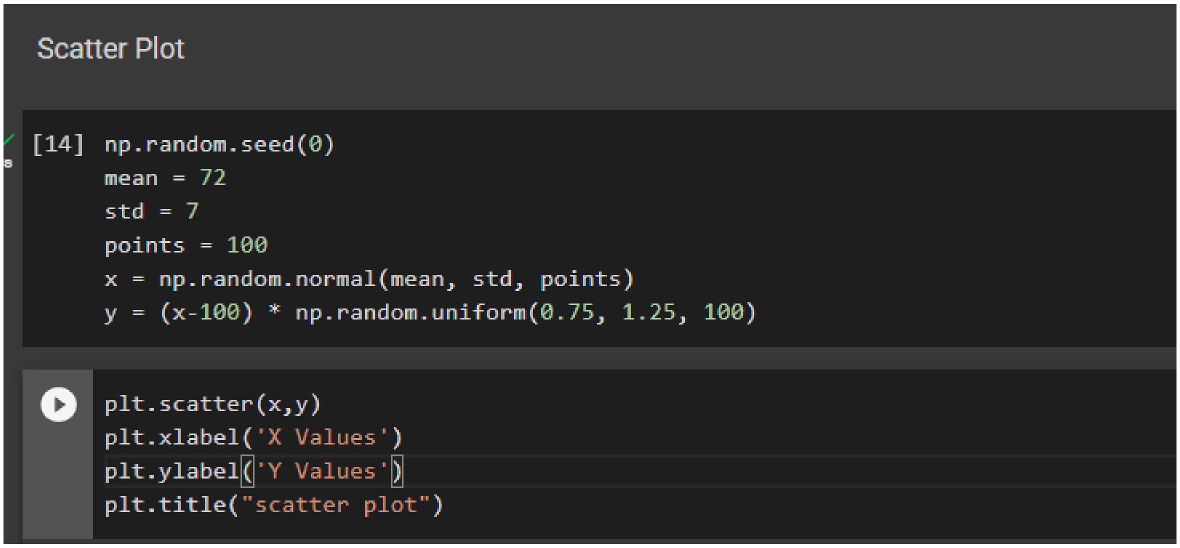
From the plot, the point where the sin x and cos x are meeting can be identified easily. As we are using 2 different colors for 2 different functions, it is easy to understand and differentiate one graph with the another. The peak and minimum values can be identified for both functions in just a glance.

This plot can be used by the owner of a store to analyze the sales of the store over a period. The other application is in the field of cricket to the flow of score over a period of few overs. This will help the players, viewers and coaches to identify the trend of the score where it was going fast and where it was going slow.

Task 2

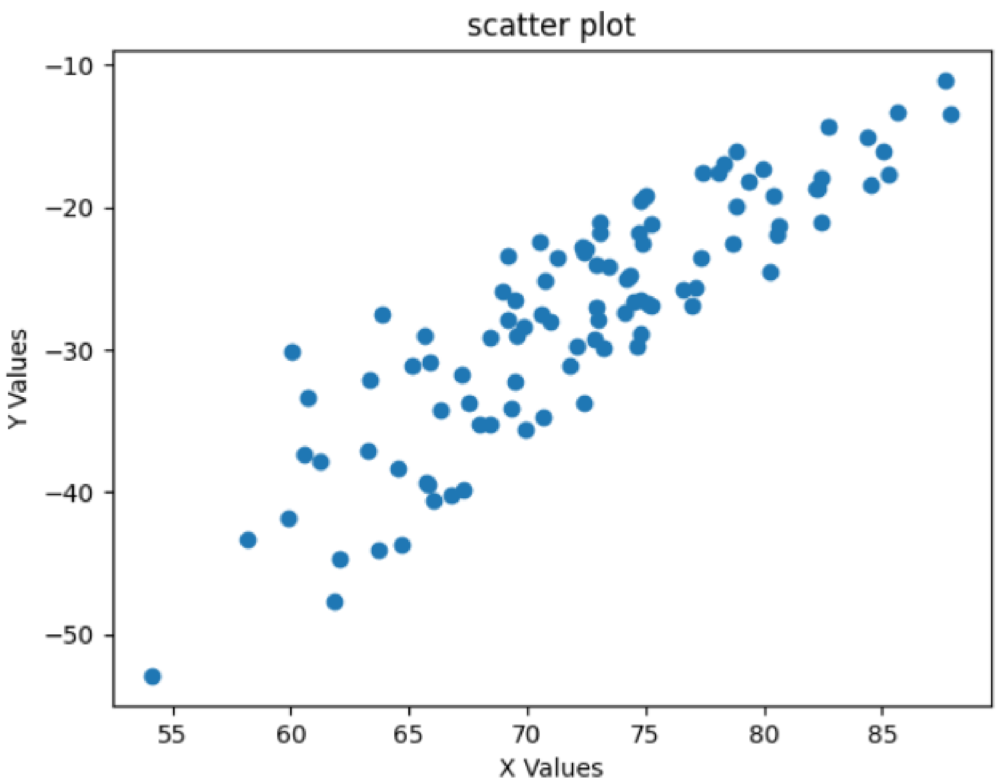
A, C & D)

Created a scatter plot for the given values of mean, standard deviation and for 100 points. Generated 100 random points for mean 72 and standard deviation 7. Created scatter plot between variables x and y with x axis as x label and y axis as y label. The title of the plot is kept as scatter plot.



```
np.random.seed(0)
mean = 72
std = 7
points = 100
x = np.random.normal(mean, std, points)
y = (x-100) * np.random.uniform(0.75, 1.25, 100)
```

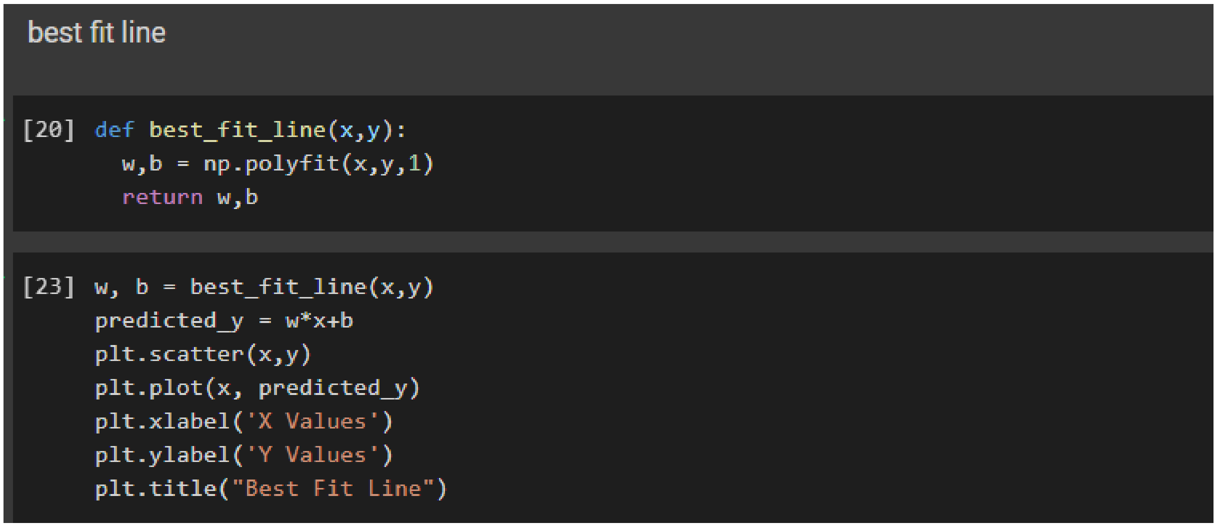
```
plt.scatter(x,y)
plt.xlabel('X Values')
plt.ylabel('Y Values')
plt.title("scatter plot")
```



Above is the plot for scatter plot

B) Best Fit Line

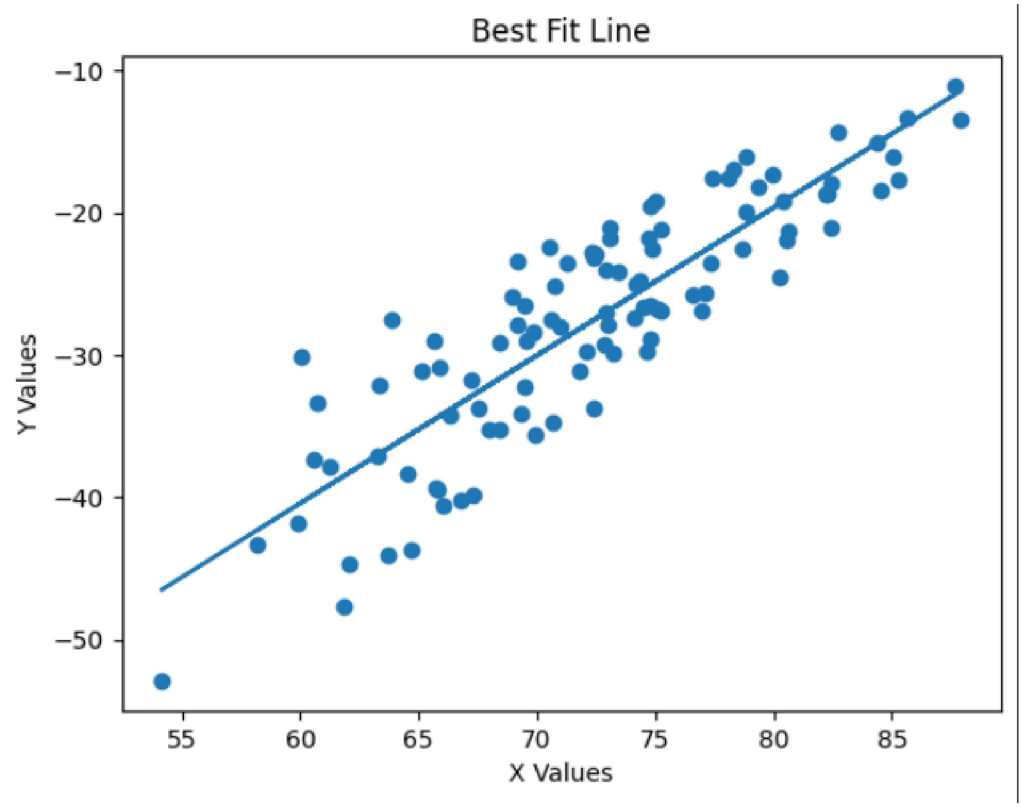
Created function named best fit line that calculates the poly fit for the points in the scatter plot and generates slope and the constant for the points. A predicted line equation is generated using the slope and constant values for x. The plot for x and predicted values has been plotted on the graph below the screenshot the code for best fit line. The axis are labeled and the graph is titled.



best fit line

```
[20] def best_fit_line(x,y):
       w,b = np.polyfit(x,y,1)
       return w,b
```

```
[23] w, b = best_fit_line(x,y)
     predicted_y = w*x+b
     plt.scatter(x,y)
     plt.plot(x, predicted_y)
     plt.xlabel('X Values')
     plt.ylabel('Y Values')
     plt.title("Best Fit Line")
```



Above is the plot for the best fit line.

E)

Strengths of scatter plot:

- Correlation and clustering effects between 2 variables can be easily interpreted.
- Scatter plots can work with any continuous scale data.
- Easy to plot.
- Outliers can be easily identified and a analyst can ignore them while analyzing the plots.

Weaknesses of scatter plot:

- The correlation can be identified but the degree of correlation cannot be obtained.
- This plot is useful only for a small number of data points.
- The relationship for more than 2 variables cannot be shown.

F)

Correlation between 2 variables can be easily identified using this plot. The correlation is positive if the plots follow an upward trend, then it is known as positive correlation and if it follows downward trend, then it is known as negative correlation.

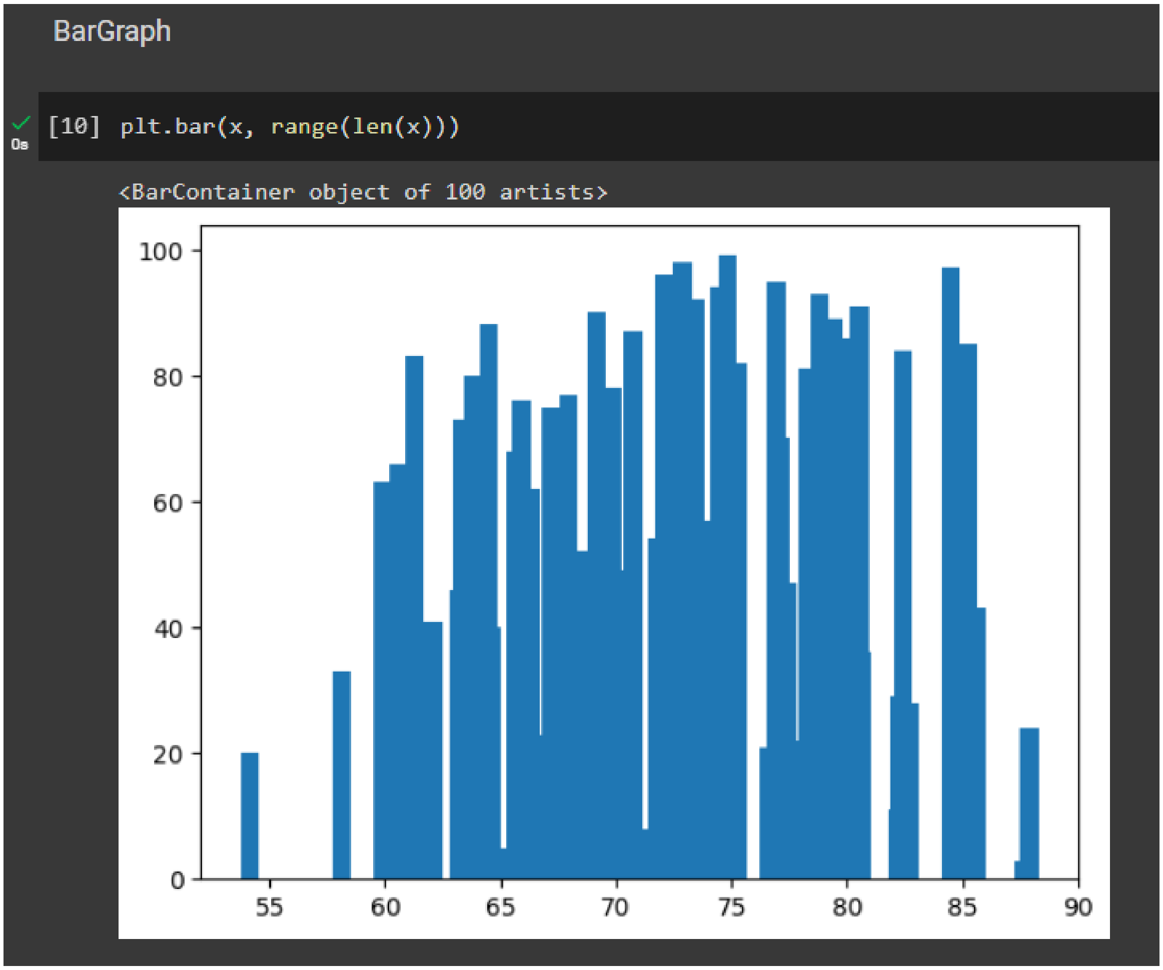
Scatter plot can be used in industries to draw the correlation between temperature and quality of the machine. This helps them to analyze the performance of the machines and helps in scheduling maintenance.

In the medical field, a scatter plot can be used to draw the correlation between patients' age and the kind of diseases that can attract a particular age group.

Task 3

A)

The bar function in matplotlib library can be used to plot a bar graph. The graph is plotted for the variable x which is already used in the question 1.



```
[10] plt.bar(x, range(len(x)))
```

```
<BarContainer object of 100 artists>
```

B)

The bars in the plot are going high over from left to right. Though there are a few bars that are going downwards, most of them are growing. Therefore, the plot is following an upward trend.

C)

Strengths of bar graphs:

- A bar graph is easy to analyze. Though the person who is not familiar with analyzing the data, He/ she will be able to understand the graph.
- Bar graphs can be used for both numerical data and categorical data.

- Comparison between different datasets can be made using bar graphs by giving different color to different categories.
- Can be used to analyze the trends of the data.

Weakness of bar graphs:

- Bar graphs are not suitable for continuous data as the bars only represent a single value are categories.
- The bar graph is limited to a small number of categories. It is hard to fit and analyze large number of categories in one plot.
- Bar graphs only provide comparisons between categories but not absolute values.

D)

Bar graphs are used for categorical data. Bar graphs can be used both vertically and horizontally. This can be used to compare values of different categories of categories.

They are widely used in sports analytics. A sports department of university can use bar graphs to compare medals won by different sports or medals won by each department for a single sport. Another scenario can be the number of medals won by a department in the past 15 years.

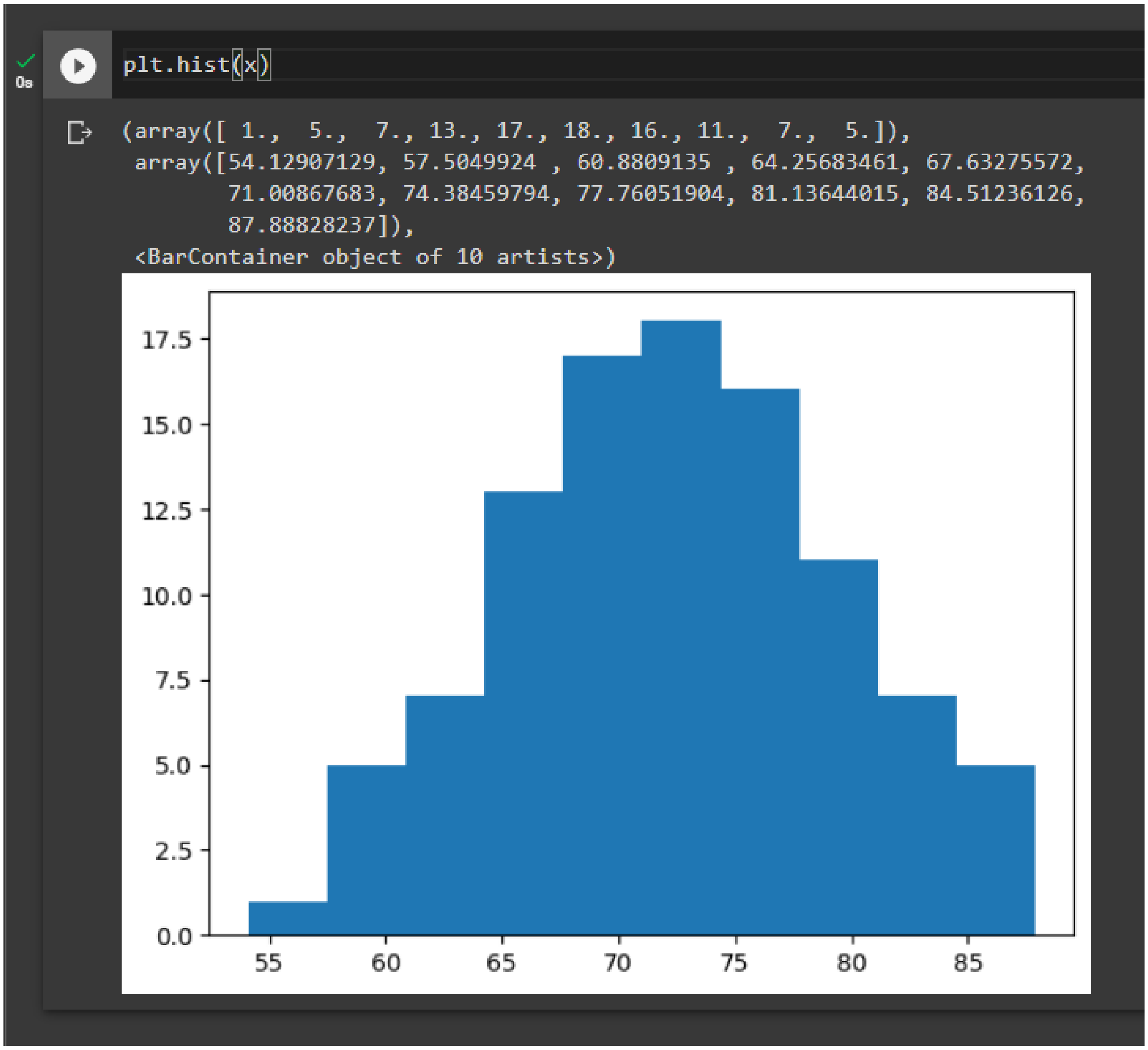
Bar graphs can be used in the field of medical sciences to show the amount of ingredients or proportions of chemicals used in a drug.

They can also be used in the field of finance to study the amount of revenue generated by different businesses in an industry.

Task 4

A)

The hist function in matplotlib library can be used to plot a histogram. The graph is plotted for the variable x which is already used in the previous question.



```
plt.hist(x)
```

```
(array([ 1.,  5.,  7., 13., 17., 18., 16., 11.,  7.,  5.]),
 array([54.12907129, 57.5049924 , 60.8809135 , 64.25683461, 67.63275572,
        71.00867683, 74.38459794, 77.76051904, 81.13644015, 84.51236126,
        87.88828237]),
 <BarContainer object of 10 artists>)
```

B)

Histogram is close to bell shaped. That means the data is following normal distribution. The data is not skewed but close to symmetric. As the graph is close to symmetric, we can say that the graph is in normal distribution. The maximum value in the graph is at a -25 on x axis and above 17.5 on y axis. The left half of the graph are following the increasing pattern and after -25 on x axis the data is following a downward trend.

C)

Strengths of histograms:

- Easy to read and understand the graph.
- Easy to compare data of different categories.
- The range of histograms is very large, it will be easy to analyze the data with huge range gaps.

Weakness of histograms:

- It is difficult to extract the exact value of a point in the graph in case of a wide range of data.
- In the case of a greater number of categories, it is hard to draw comparisons between selected categories.
- The precise information of the graph cannot be extracted unless plotted with frequency distribution.

D)

Histogram charts are most effective. They may be used to determine the distribution's form, such as measures of central tendency (mean, median, mode), skewness, and kurtosis. Histograms are very important when working with huge datasets since they allow for fast and efficient data processing.

Histograms can be used in the field of stock market to analyze the distribution of investments and returns or price of the stocks.

In medical field, histograms can be used to show the blood sugar levels, cholesterol levels and various other metrics and that helps to compare patients data with the ideal values of that corresponding category.