

Matplotlib is used to draw graphs (plots) in Python

Used for:

Understanding data

Visualizing ML results

Debugging models

Reports

```
# pip install matplotlib
```

```
import matplotlib.pyplot as plt
```

```
#plt = shortcut name
```

```
#Almost all plots use plt
```

```
# Matplotlib na graph draw panna use panra Python library.  
# Numbers paakuradhu kashtam, graph paatha easy-aa puriyum.
```

```
import matplotlib.pyplot as plt
```

```
x = [1, 2, 3, 4, 6]  
y = [10, 20, 25, 30, 19]
```

```
plt.plot(x, y)  
plt.show()
```

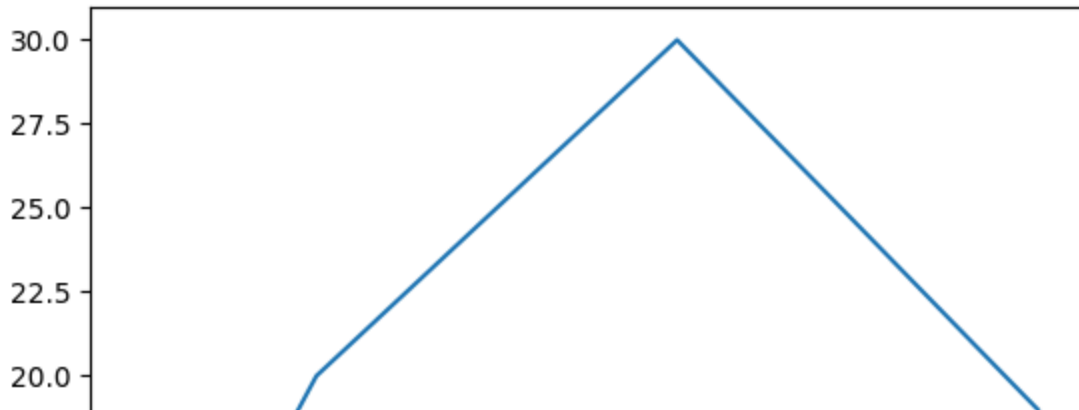
```
# x → horizontal axis
```

```
# y → vertical axis
```

```
# plot() → draw line
```

```
# show() → display graph
```

```
# x na keezha pogum line  
# y na mela pogum line  
# plt.plot() na line draw pannum  
# plt.show() illatti graph kaataadhu
```



```
x = [1, 2, 3, 4, 6, 5]
y = [30, 20, 35, 40, 19, 12]
```

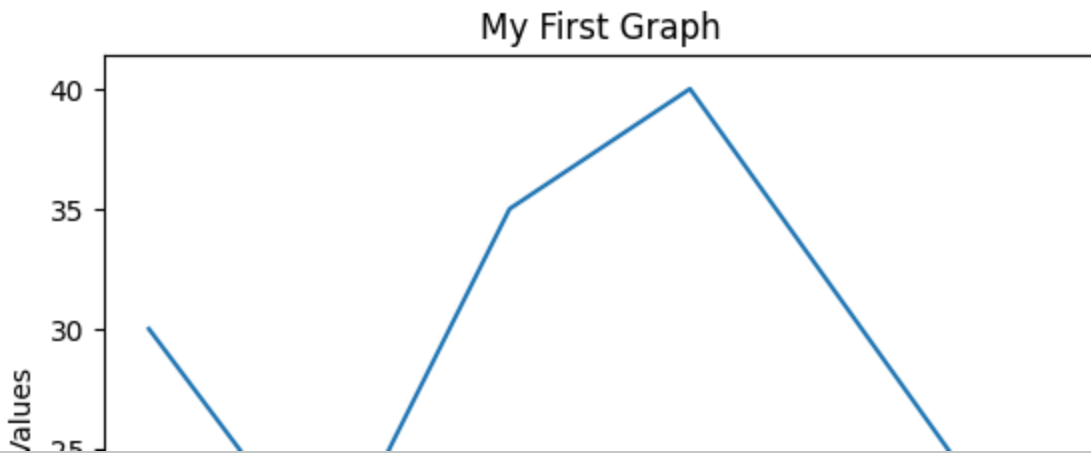
```
plt.plot(x, y)
plt.title("My First Graph")
plt.xlabel("X Values")
plt.ylabel("Y Values")
plt.show()
```

# title → graph oda peru

# xlabel → x axis peru

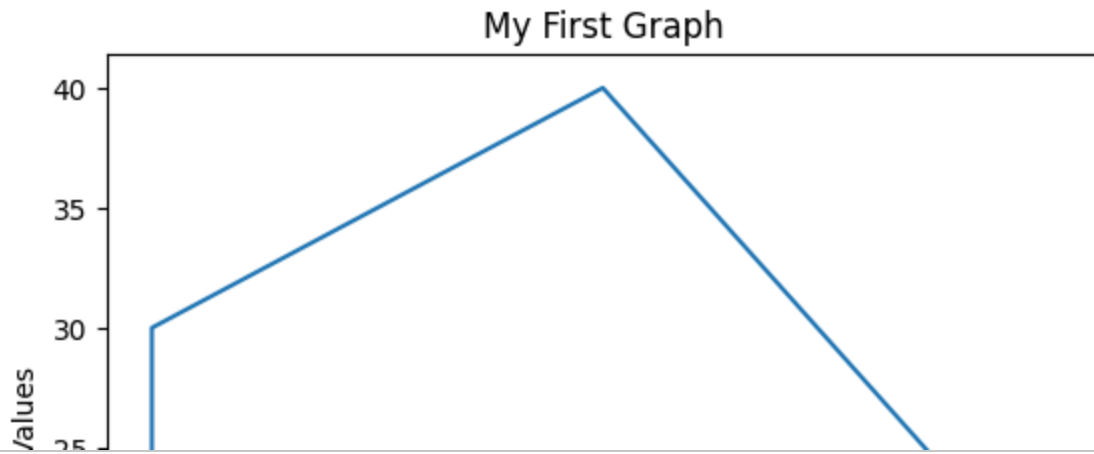
# ylabel → y axis peru

# Idhu interview + report la romba important



```
x = [3, 2, 2, 3, 4, 4]
y = [15, 20, 30, 40, 19, 12]
```

```
plt.plot(x, y)
plt.title("My First Graph")
plt.xlabel("X Values")
plt.ylabel("Y Values")
plt.show()
```

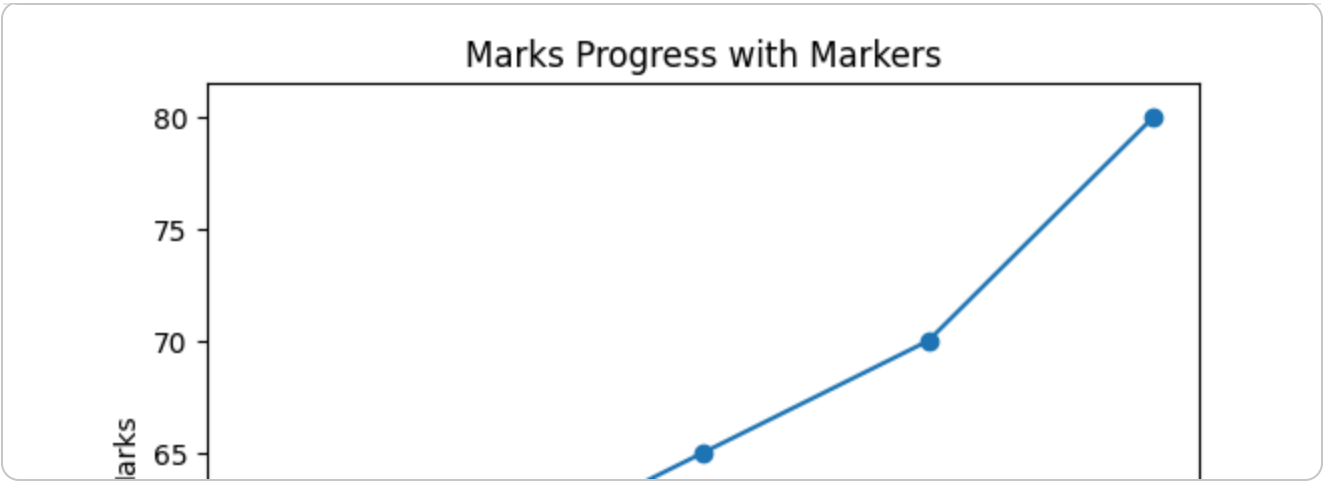


```
temp = np.array([30, 32, 31, 29, 28, 27, 26])
```

```
-----  
---  
NameError                                Traceback (most recent call  
last)  
/tmp/ipython-input-2773913153.py in <cell line: 0>()  
----> 1 temp = np.array([30, 32, 31, 29, 28, 27, 26])  
  
NameError: name 'np' is not defined
```

```
sales = np.array([2000, 2500, 3000, 2800, 3500, 4000])
```

```
import numpy as np  
import matplotlib.pyplot as plt  
  
marks = np.array([50, 60, 65, 70, 80])  
exams = np.arange(1, 6)  
  
plt.plot(exams, marks, marker='o')  
plt.title("Marks Progress with Markers")  
plt.xlabel("Exam")  
plt.ylabel("Marks")  
plt.show()
```

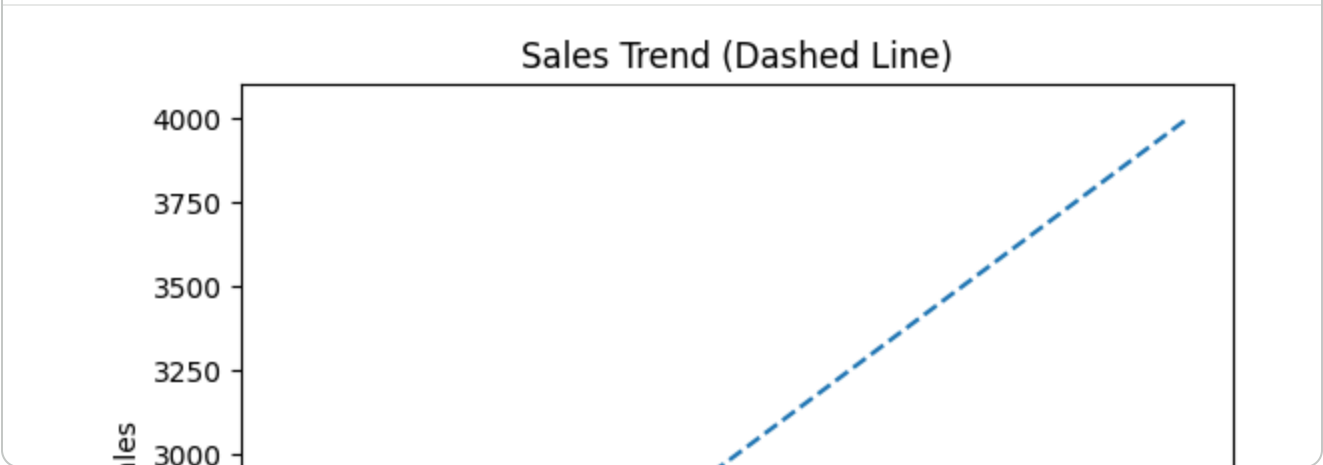


Marker	Meaning
--------	---------

'o'	Circle
's'	Square
'^'	Triangle
'*'	Star
'x'	Cross

```
# change line style
sales = np.array([2000, 2500, 3000, 3500, 4000])
months = np.arange(1, 6)

plt.plot(months, sales, linestyle='--')
plt.title("Sales Trend (Dashed Line)")
plt.xlabel("Month")
plt.ylabel("Sales")
plt.show()
```



Style	Code
-------	------

Solid	'-'
Dashed	'--'
Dotted	'...'
Dash-dot	'-...'

## > Bar plot

↳ 3 cells hidden

## ✓ Scatter plot

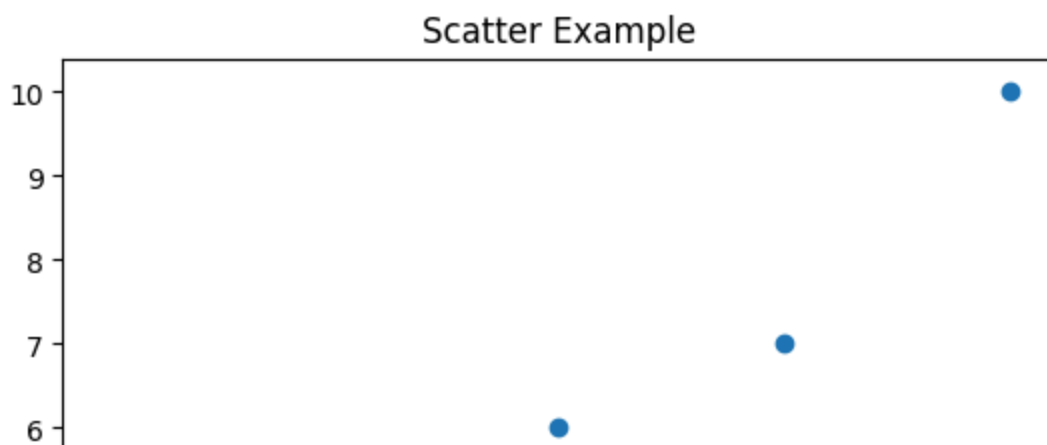
```
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 7, 10]

plt.scatter(x, y)
plt.title("Scatter Example")
plt.show()

#Relationship between variables

# ML data visualization

#nScatter plot la line illa, points mattum
# ML-la idha use pannu pattern irukka nu paapom
```

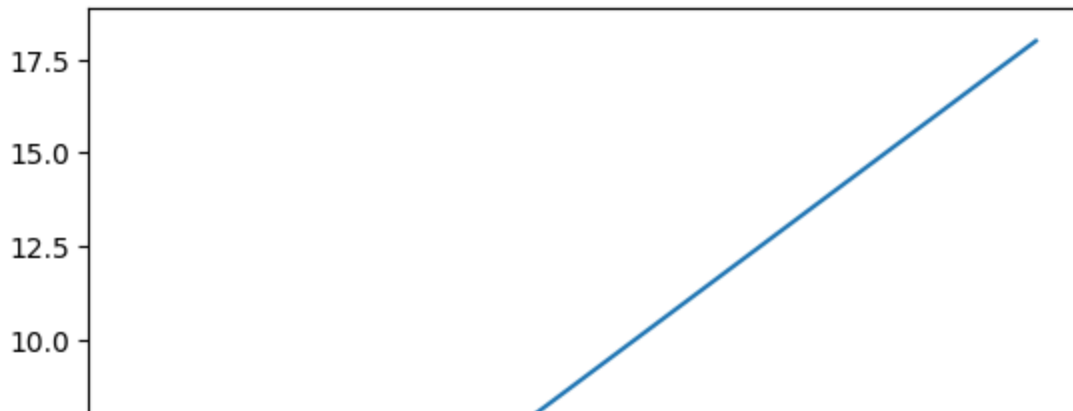


## ✓ Plot NumPy Array

```
import numpy as np

x = np.arange(0, 10) # already discussed in numpy colab notebook, oru vati par
y = x * 2

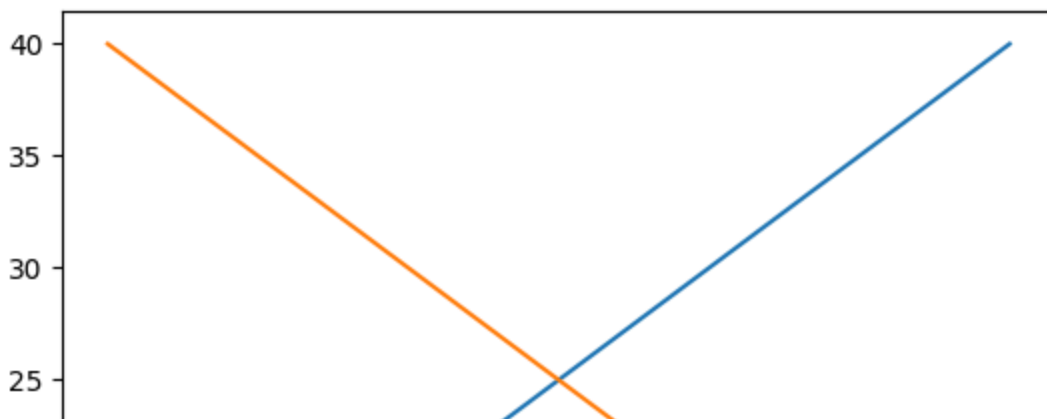
plt.plot(x, y)
plt.show()
```



## ✓ Multiple Lines in One Grap

```
x = [1, 2, 3, 4]

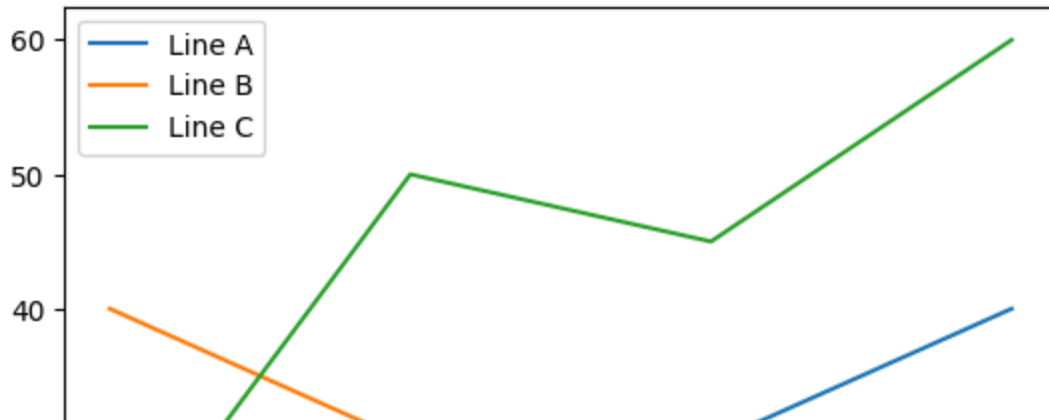
plt.plot(x, [10, 20, 30, 40])
plt.plot(x, [40, 30, 20, 10])
plt.show()
# Ore graph-la rendu line compare panna idhu use aagum
# Training vs Validation loss maari
```



```
plt.plot(x, [10, 20, 30, 40], label="Line A")
plt.plot(x, [40, 30, 20, 10], label="Line B")
```

```
plt.plot(x, [20, 50, 45, 60], label="Line C")
plt.legend()
plt.show()
```

```
# Legend illatti entha line edhu nu theriyadhu (Legend na inge label nu meaning)
# Legend na label kaatrum box
```

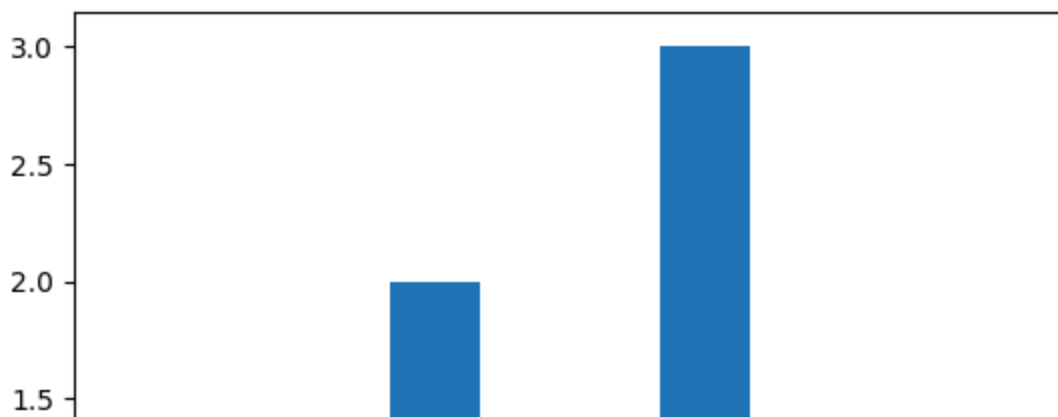


## ✓ Histogram (Data distribution)

```
data = [10, 20, 20, 30, 30, 30, 40]
```

```
plt.hist(data)
plt.show()
```

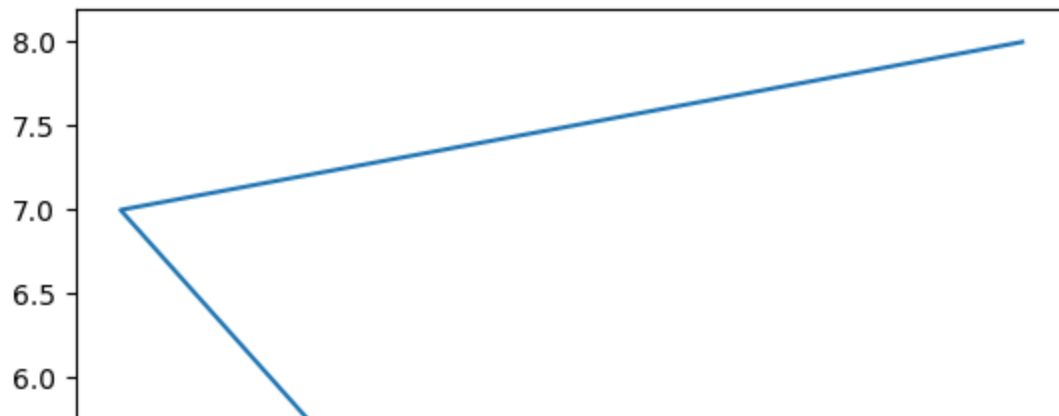
```
# Histogram na ethana vaati value repeat aagudhu nu kaatum
#Used for: Frequency Distribution, ML preprocessing
```



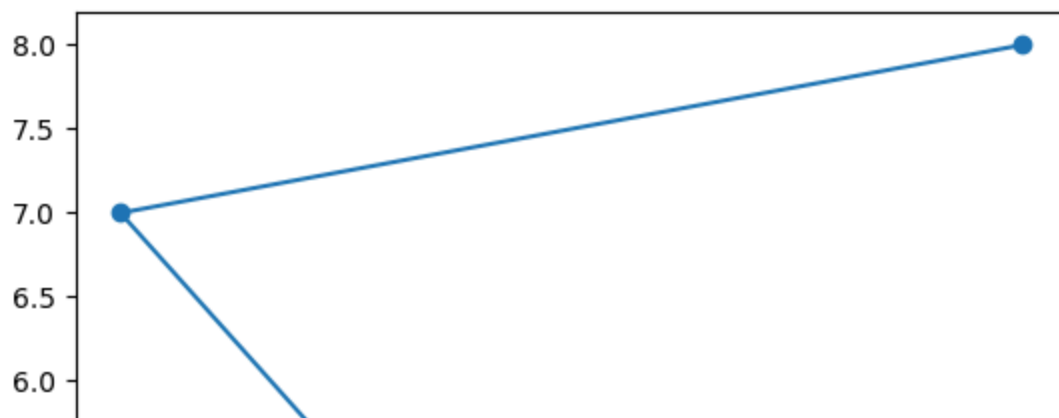
Double-click (or enter) to edit

## ✓ More deeper examples

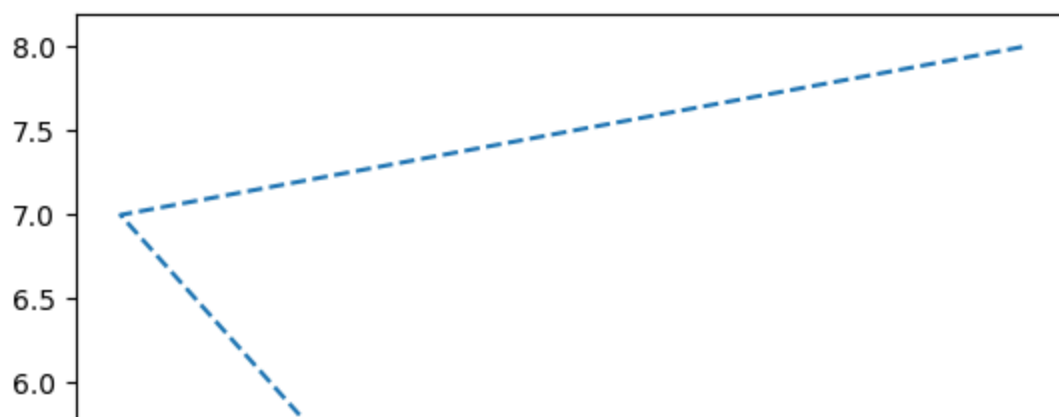
```
plt.plot([4, 2, 3], [8, 7, 4])  
plt.show()  
# Simple-aa points connect pannra line
```



```
plt.plot([4, 2, 3], [8, 7, 4], marker='o')  
plt.show()  
# Line oda points clear-aa kaatanum na marker use pannuvom
```



```
plt.plot([4, 2, 3], [8, 7, 4], linestyle='--')  
plt.show()  
#-- na dotted line  
# Model comparison la use aagum
```

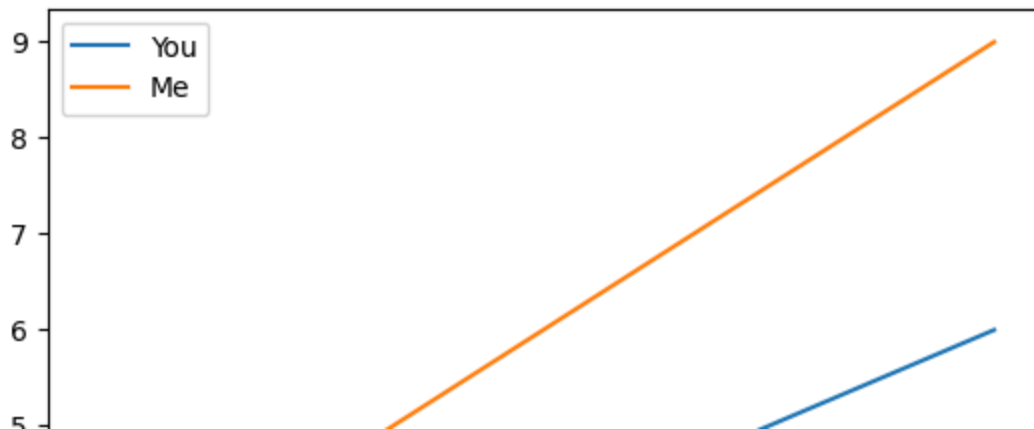


```
# multiple lines together
```



```
x = [1, 2, 3]

plt.plot(x, [2, 4, 6], label="You")
plt.plot(x, [3, 6, 9], label="Me")
plt.legend()
plt.show()
# lable name yedhu venalum vechikalam, eg: lineA, LineB...You , Me.....
```



```
# BAR CHART - plt.bar() (Varieties)
plt.bar(["Chennai", "Banglore", "Tirupati"], [10, 20, 15])
plt.show()
# Category compare panna bar chart best
```



```
# Horizontal bar

plt.barh(["Chennai", "Pondicherry", "New York"], [10, 20, 15])
plt.show()

# plt.barh use paniti , bar graph horizontal ah create pannalam
```

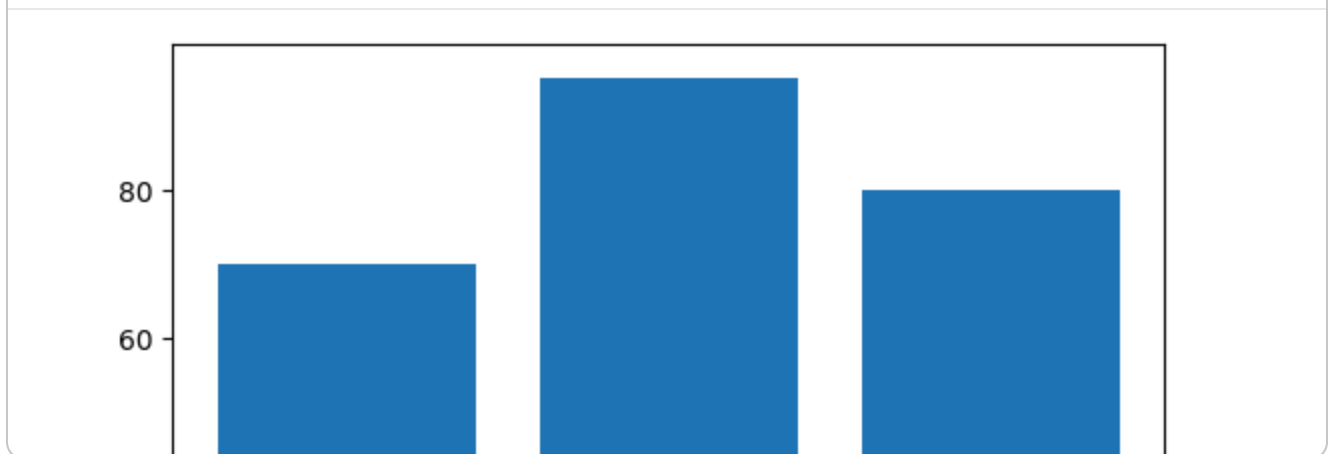


```
# task
student = ["Ramesh", "Ajith", "Vijay"]
marks = [70, 95, 80]

# vertical bar

plt.bar(student, marks)
plt.show()

# plot the same graph vertically
```

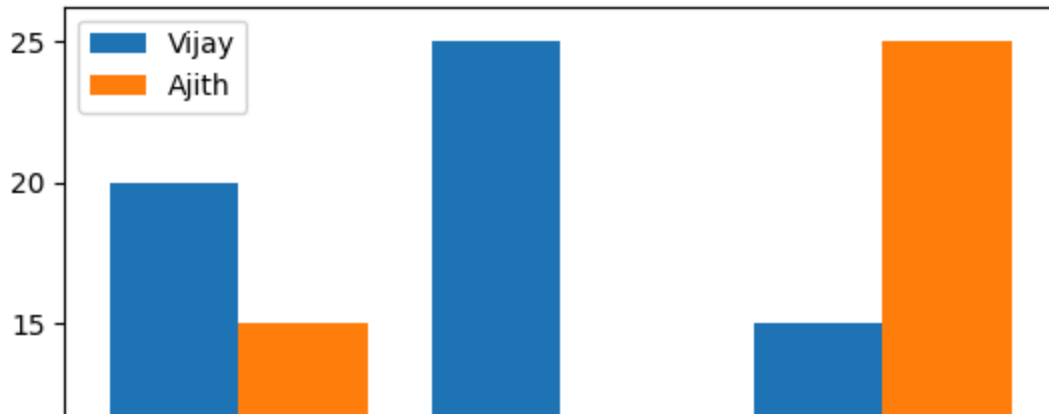


```
# multiple bars
```

```
import numpy as np

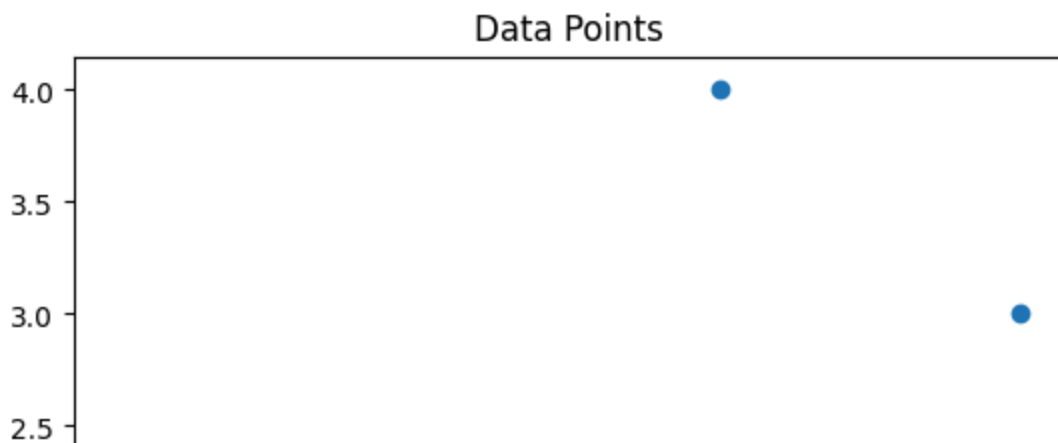
x = np.arange(3) # idhu namaku range of numbers 0, 1 , 2 kudukum , adha x axis

plt.bar(x - 0.2, [20, 25, 15], width=0.4, label="Vijay") # idhu oru color la
plt.bar(x + 0.2, [15, 10, 25], width=0.4, label="Ajith") # idhu vere color la
plt.legend()
plt.show()
```

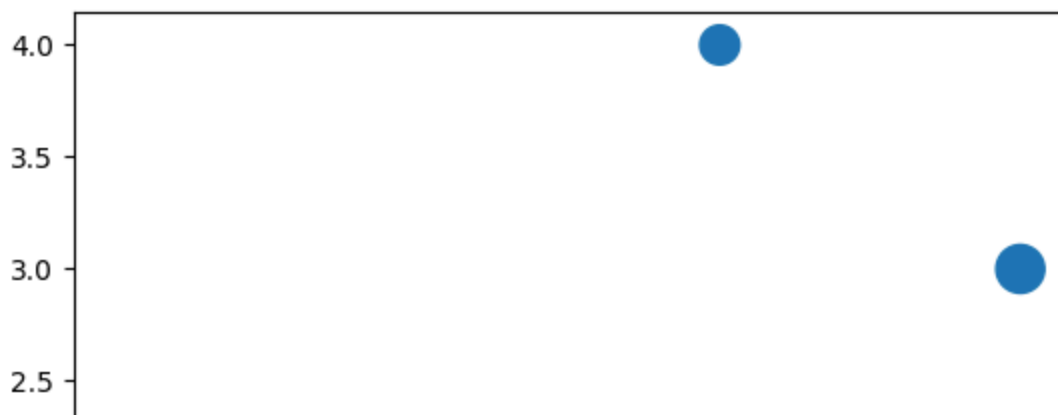


```
x = [1, 2, 3, 4]
y = [2, 1, 4, 3]

plt.scatter(x, y)
plt.title("Data Points")
plt.show()
```



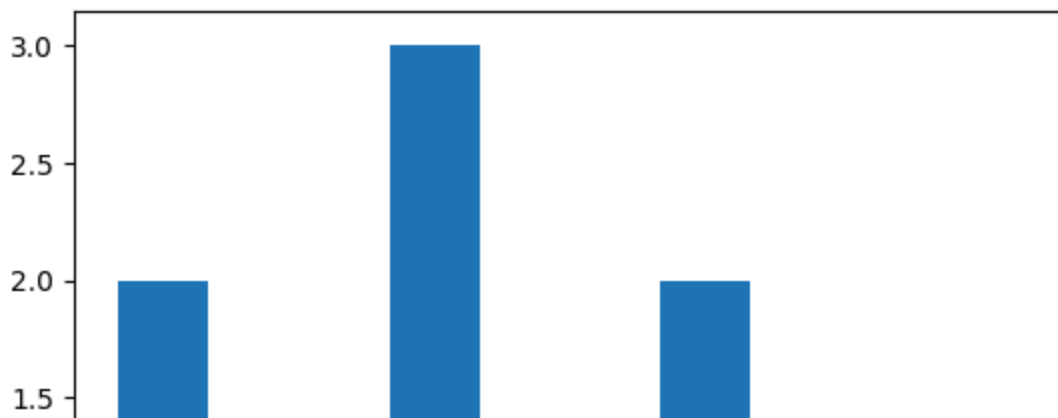
```
plt.scatter(x, y, s=[50, 100, 200, 300])
plt.show()
# s na point size
# Importance kaatanum na use pannuvom
```



```
# Histogram
data = [10, 20, 20, 20, 30, 30, 40, 10]
```

```
# inge 10 rendu vati , 20 rendu vati, 30 moonu vati 40 oru vati vandhirki, adha
plt.hist(data)
plt.show()

#Ethana vaati value repeat aagudhu nu distribution kaatum
```



```
plt.hist(data, bins=4)
# bins na verum , athana bars mattume namma use panananum , inge bins=4 so 4 bars
# meaning motha data andha 4 category groups ah mathi , oru oru group kulla y
plt.show()
```

```
# simple ah solanumna , inge data least value 10 , highest value 40 , so 10 to 40
# 40-10 is 30 , so difference 30 dha,, nama use panavendia categories (bins 4)
# so 30/4 = 7.5 , so oru oru bin , meaning oru oru bar 7.5 width la varum x axis
# next example pakalam
```



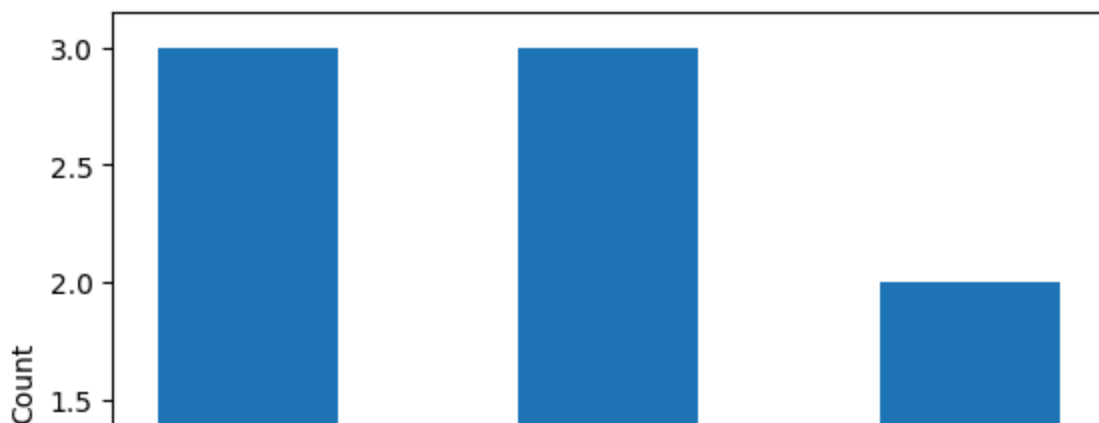
```
scores = [5, 6, 4, 1, 5, 7, 4, 9, 6, 7]
# inge pakalam , maximum value 9, minimum value 1, so 9-1 is 8 , so motha 8/bin
# 8 divided by 4 bins i.e 8/4 is 2
# so oru oru bin width 2 irukum , ipidi motha data 4 bins kulla divide ayindhi
# diagram pathingana first bin 1 landhu 3 varukum iriki meaning (3-1) is 2 width
# min 1 ,max 9 , so 1+2 is 3 ,so 1 to 3 oru bin (1 ku 3 ku nadula irukara data
# next 3 + 2(bin width) is 5 , so 3 to 5 oru bin (3 to 5 ku nadula 4 mattum 2
# 3 to 5 means , considering 3, 4 not considering 5(upperbound)
plt.hist(scores, bins=4)
plt.xlabel('Score')
```

```
plt.ylabel('Count')
plt.show()
```



```
scores = [45, 67, 34, 91, 55, 78, 40, 99, 63, 71]
# inge pakalam , maximum value 99, minimum value 34, so 99-34 is 65 , so motha
# 65 divided by 5 bins i.e 65/5 is 13
# so oru oru bin width 13 irukum , ipidi motha data 5 bins kulla divide ayindh

plt.hist(scores, bins=5)
plt.xlabel('Score')
plt.ylabel('Count')
plt.show()
```



Understanding Bins:

bins=5 → Groups data into 5 buckets (broad view)

bins=10 → Groups data into 10 buckets (balanced - recommended)

bins=20 → Groups data into 20 buckets (detailed view)

bins=[0, 50, 70, 90, 100] → Custom ranges (like grade boundaries)

```
scores = [45, 67, 82, 91, 55, 78, 88, 92, 63, 71]
```

```
plt.hist(scores, bins=5)
plt.xlabel('Score')
plt.ylabel('Count')
plt.show()
```



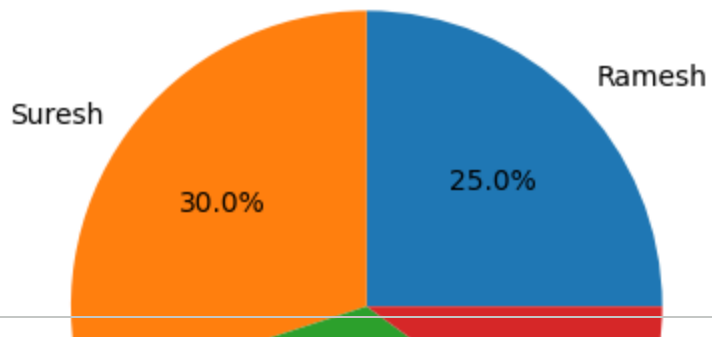
PIE chart

```
sizes = [25, 30, 35, 10]
labels = ["Ramesh", "Suresh", "Ganesh", "Vijay"]

plt.pie(sizes, labels=labels)
plt.show()
```



```
# same pie plot with percentages
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
# autopct na percentage numbers show panum
plt.show()
```



```
# Area plot  
plt.fill_between([1,2,3,4], [2,4,8,5])  
plt.show()  
# Line keezha area fill aagum  
# Growth visual-a kaatanum na use pannuvom
```

