

- [Saving and loading notebooks in GitHub](#)
- [Interactive forms](#)
- [Interactive widgets](#)

## Working with data

- [Loading data: Drive, Sheets and Google Cloud Storage](#)
- [Charts: visualising data](#)
- [Getting started with BigQuery](#)

## Machine learning crash course

These are a few of the notebooks from Google's online machine learning course. See the [full course website](#) for more.

- [Intro to Pandas DataFrame](#)
- [Linear regression with tf.keras using synthetic data](#)

## Using accelerated hardware

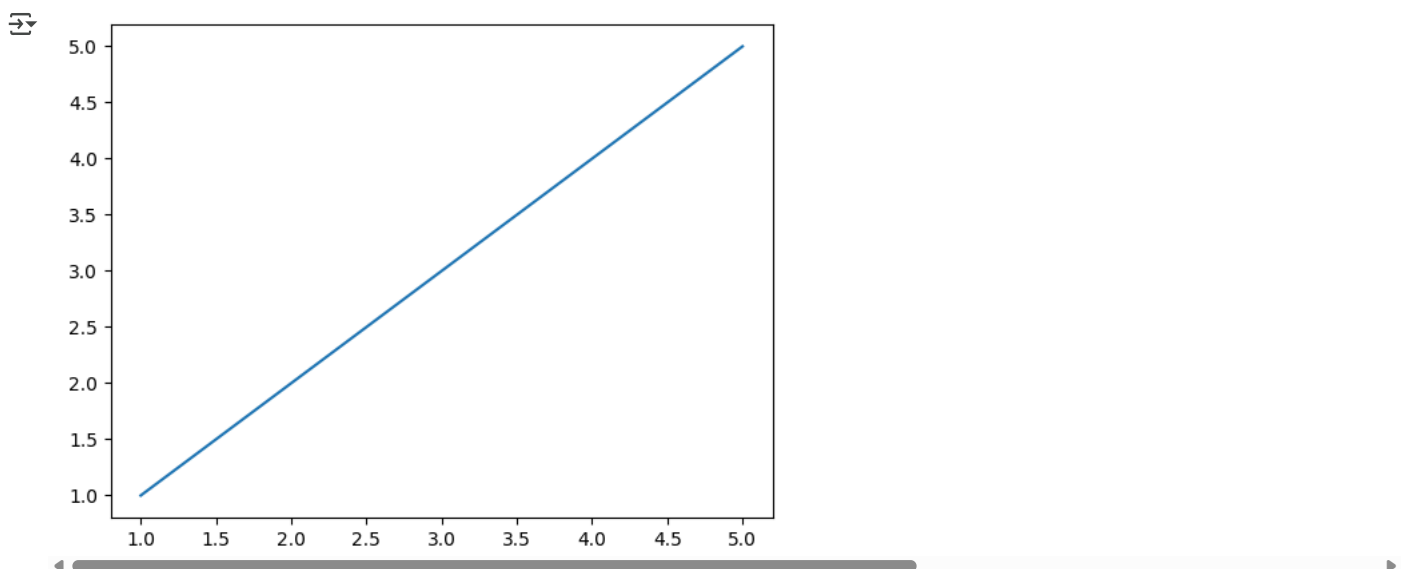
- [TensorFlow with GPUs](#)
- [TensorFlow with TPUs](#)

## ✓ Featured examples

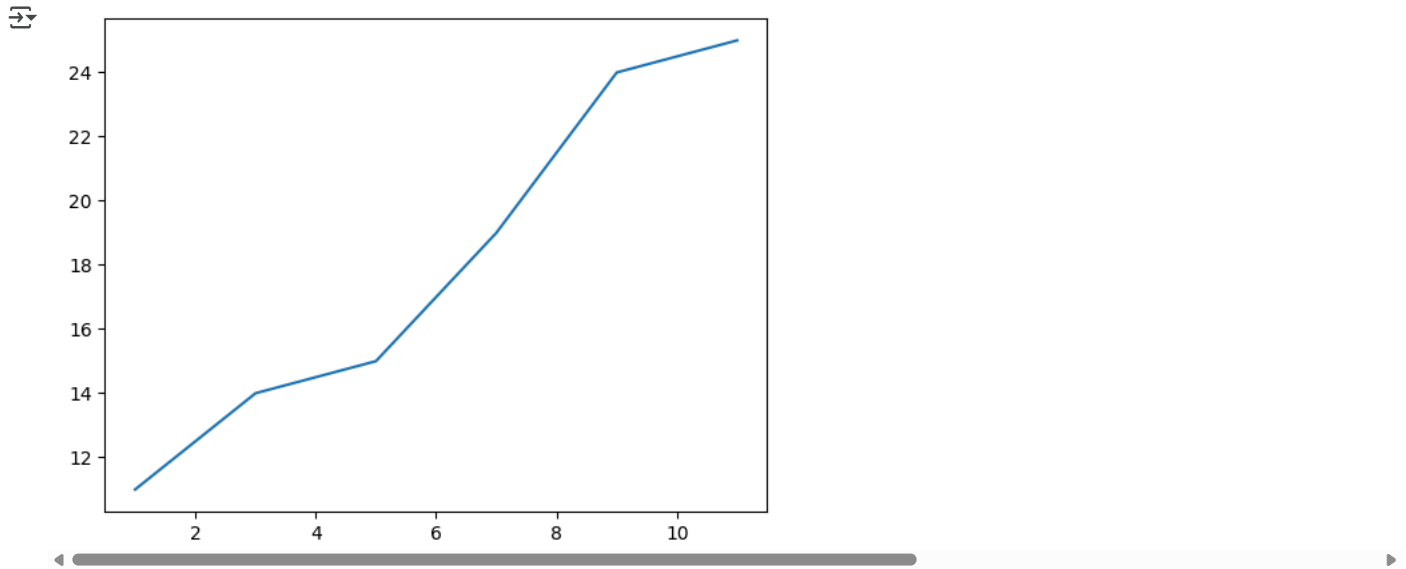
- [NeMo voice swap](#): Use Nvidia NeMo conversational AI toolkit to swap a voice in an audio fragment with a computer-generated one.
- [Retraining an Image Classifier](#): Build a Keras model on top of a pre-trained image classifier to distinguish flowers.
- [Text Classification](#): Classify IMDB film reviews as either *positive* or *negative*.
- [Style Transfer](#): Use deep learning to transfer style between images.
- [Multilingual Universal Sentence Encoder Q&A](#): Use a machine-learning model to answer questions from the SQuAD dataset.
- [Video Interpolation](#): Predict what happened in a video between the first and the last frame.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

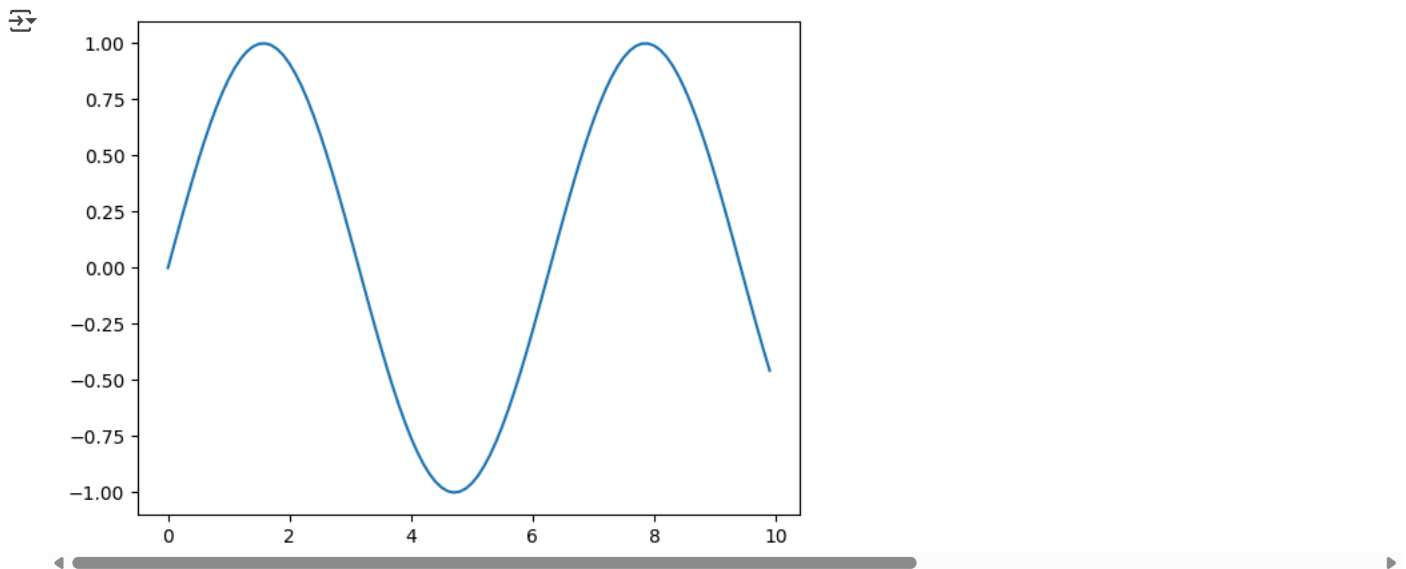
```
x=[1,2,3,4,5]
y=x
plt.plot(x,y)
plt.show()
```



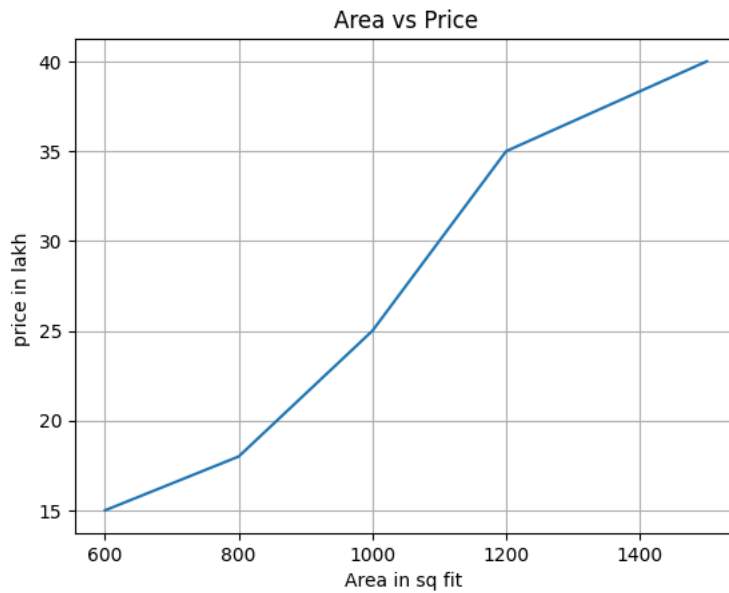
```
x=[1,3,5,7,9,11]
y=[11,14,15,19,24,25]
plt.plot(x,y)
plt.show()
```



```
x=np.arange(0,10,0.1)
y=np.sin(x)
plt.plot(x,y)
plt.show()
```

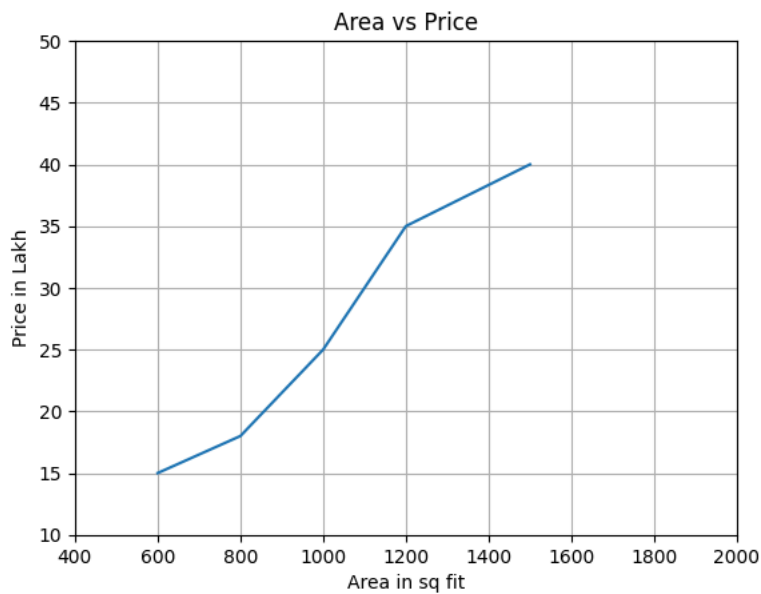


```
area=[600,800,1000,1200,1500]
price=[15,18,25,35,40]
plt.plot(area,price)
plt.title("Area vs Price")
plt.xlabel("Area in sq fit ")
plt.ylabel("price in lakh ")
plt.grid(True)
plt.show()
```

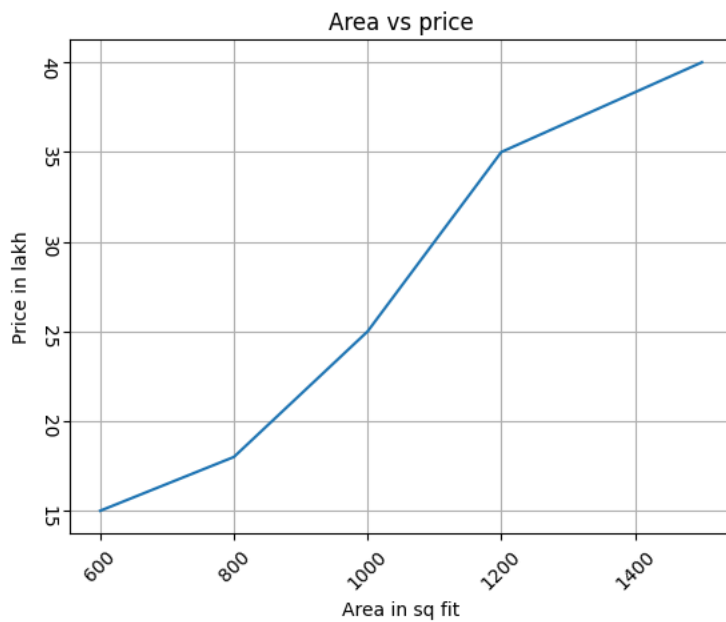


#xlim and ylim to define x and y range

```
plt.plot(area, price)
plt.title("Area vs Price")
plt.xlabel("Area in sq fit")
plt.ylabel("Price in Lakh")
plt.xlim((400,2000))
plt.ylim((10,50))
plt.grid(True)
plt.show()
```



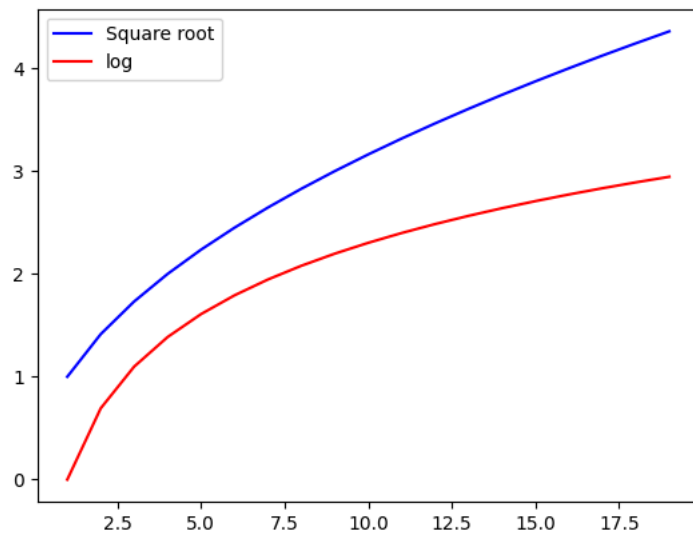
```
plt.plot(area, price)
plt.title("Area vs price")
plt.xlabel("Area in sq fit")
plt.ylabel("Price in lakh")
plt.xticks(rotation=45)
plt.yticks(rotation=270)
plt.grid(True)
plt.show()
```



#Multiple Plots

```
x=np.arange(1,20)
y1=np.sqrt(x)
y2=np.log(x)
```

```
plt.plot(x,y1, color="blue")
plt.plot(x,y2, color="red")
plt.legend(labels=["Square root", "log"])
plt.show()
```

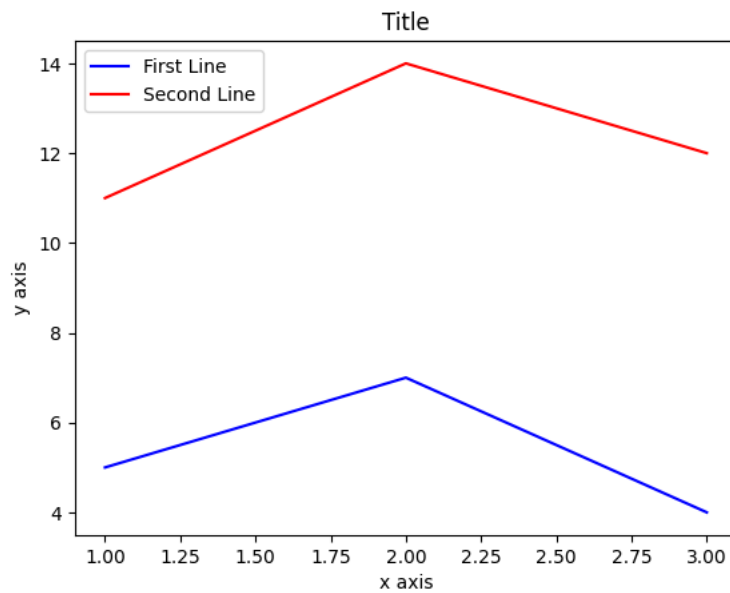


```
x=[1,2,3]
y=[5,7,4]
```

```
x1=[1,2,3]
y1=[11,14,12]
```

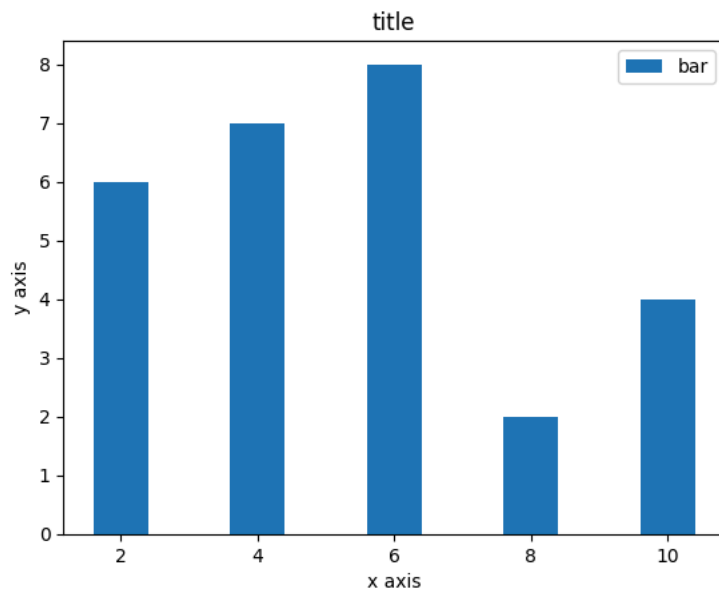
```
plt.plot(x,y, label="First Line", color="blue")
plt.plot(x1,y1, label="Second Line", color="red")
```

```
plt.title("Title")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.legend()
plt.show()
```



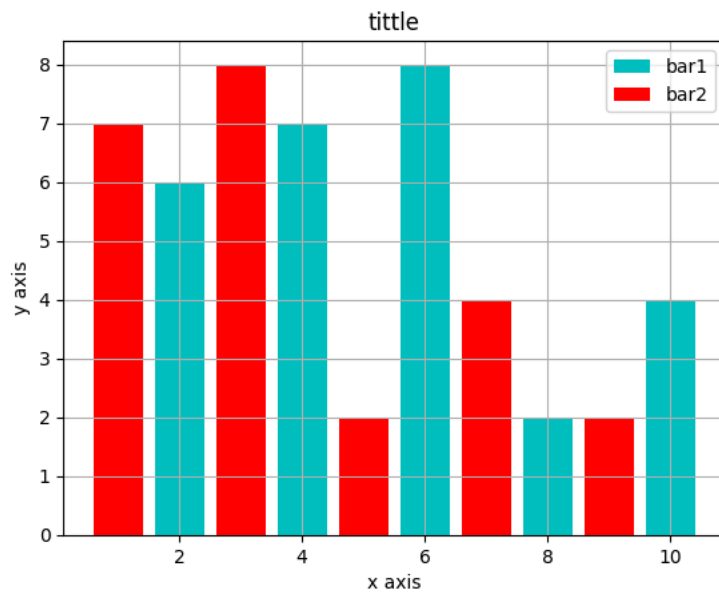
#Bra chat & Histogram

```
x,y=[2,4,6,8,10],[6,7,8,2,4]
plt.bar(x,y, label="bar")
plt.title("title")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.legend()
plt.show()
```



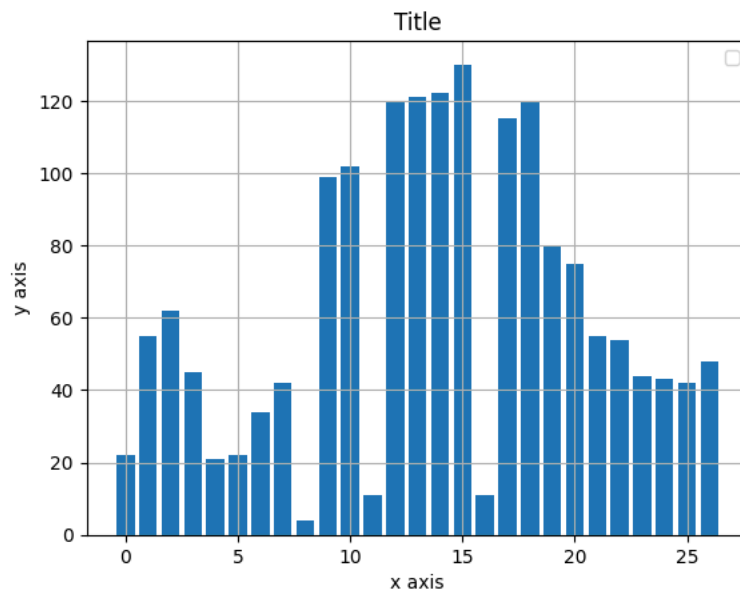
```
x,y=[2,4,6,8,10], [6,7,8,2,4]
x1,y1=[1,3,5,7,9],[7,8,2,4,2]
```

```
plt.bar(x,y,label="bar1", color="c")
plt.bar(x1,y1, label="bar2", color="r")
plt.title("tittle")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.grid(True)
plt.legend()
plt.show()
```



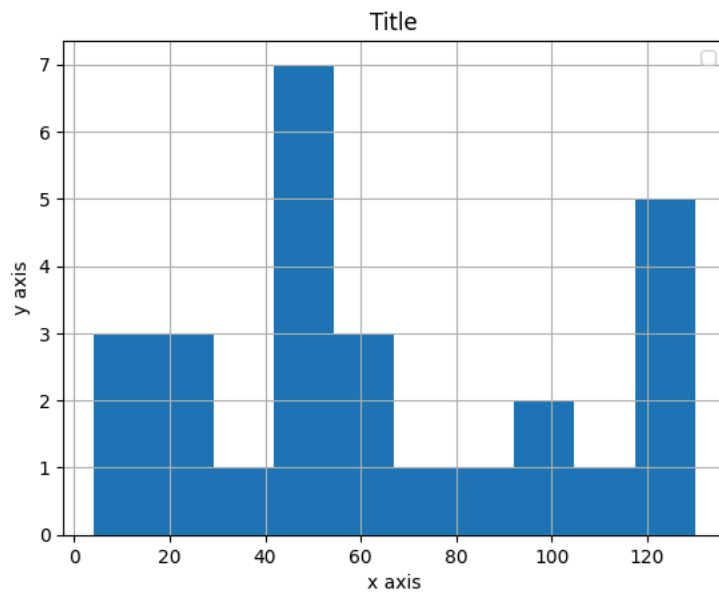
```
population_ages=[22,55,62,45,21,22,34,42,4,99,102,11,120,121,122,130,11,115,120,80,75,55,54,44,43,42,48]
```

```
ids=[x for x in range(len(population_ages))]
plt.bar(ids,population_ages)
plt.title("Title")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.grid(True)
plt.legend()
plt.show()
```



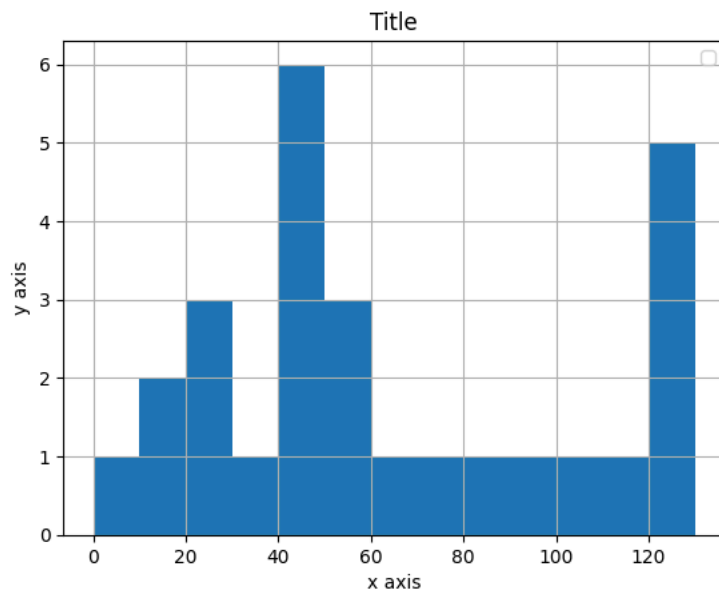
```
population_ages=[22,55,62,45,21,22,34,42,4,99,102,11,120,121,122,130,11,115,120,80,75,55,54,44,43,42,48]
```

```
plt.hist(population_ages)
plt.title("Title")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.grid(True)
plt.legend()
plt.show()
```



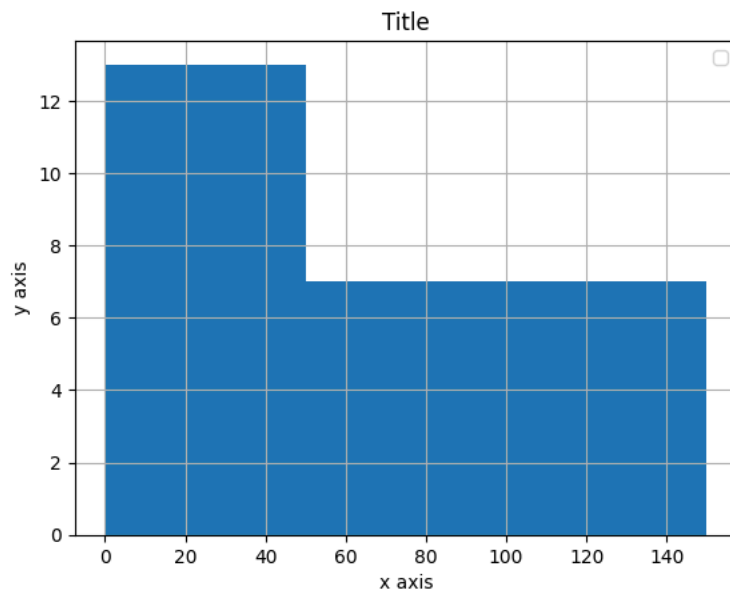
```
population_ages=[22,55,62,45,21,22,34,42,4,99,102,11,120,121,122,130,11,115,120,80,75,55,54,44,43,42,48]
```

```
bins=[0,10,20,30,40,50,60,70,80,90,100,110,120,130]
plt.hist(population_ages,bins)
plt.title("Title")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.grid(True)
plt.legend()
plt.show()
```



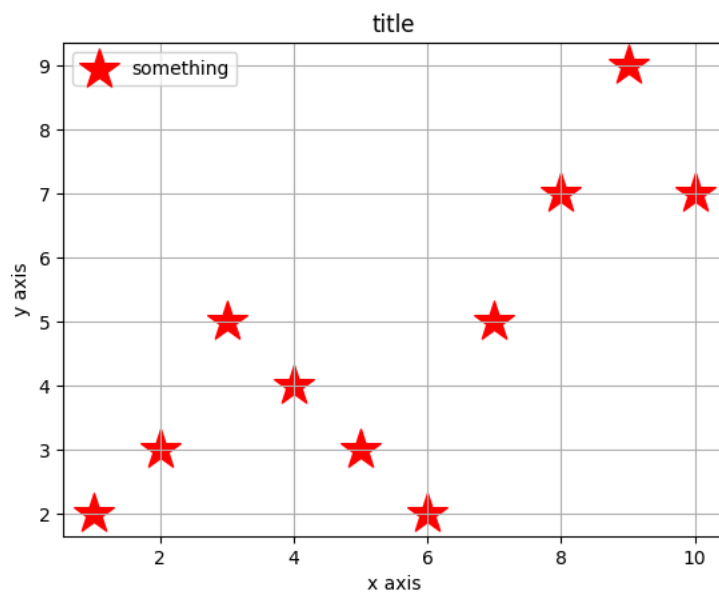
```
population_ages=[22,55,62,45,21,22,34,42,4,99,102,11,120,121,122,130,11,115,120,80,75,55,54,44,43,42,48]
```

```
bins=[0,50,100,150]
plt.hist(population_ages,bins)
plt.title("Title")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.grid(True)
plt.legend()
plt.show()
```



#Scatter plot

```
x=[1,2,3,4,5,6,7,8,9,10]
y=[2,3,5,4,3,2,5,7,9,7]
plt.scatter(x,y, label="something", color="r", marker="*", s=500)
plt.title("title")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.grid(True)
plt.legend()
plt.show()
```



#pie plot

```
slices=[7,2,2,13] # 24 hrs
activites=["sleep", "eating", "gye", "working"]
colos=["c", "m", "r", "b"]

plt.pie(slices, labels=activites, colors=colos)
plt.title("my pie chart")
plt.show()
```





my pie chart

