In [1]:
```
import pandas as pd
data=[10,20,30,40,50,60]
df1=pd.DataFrame(data,columns=['Numbers'])
df1
```

Out[1]:

	Numbers
0	10
1	20
2	30
3	40
4	50
5	60

In [3]:
```
data={'color': ['red','blue','green','blue']}#DICTONARY DATA TYPE
df=pd.DataFrame(data)
```

In [5]:
```
df
```

Out[5]:

	color
0	red
1	blue
2	green
3	blue

In [7]:
```
data=[['tom',10], ['jay',15],['juli',14]]
dfx=pd.DataFrame(data,columns=['Name','age'])
dfx
```

Out[7]:

	Name	age
0	tom	10
1	jay	15
2	juli	14

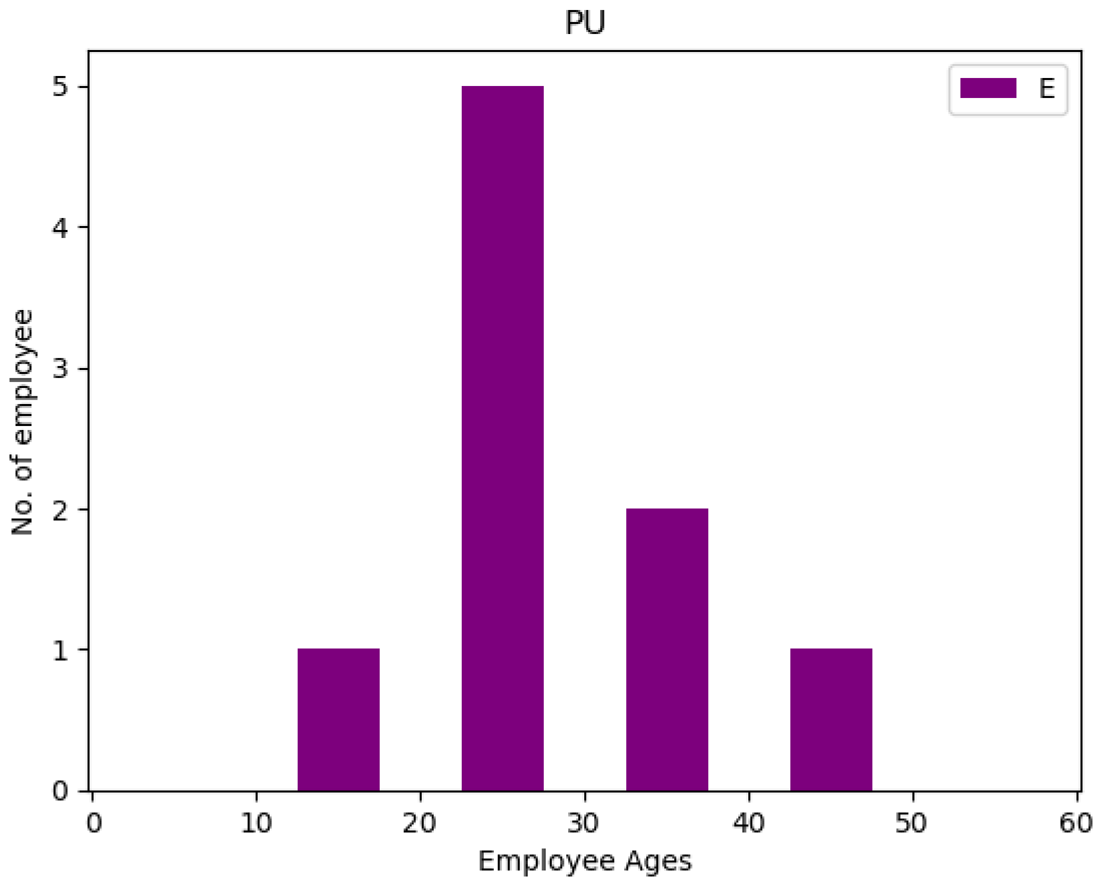
In [9]:
```
df_encoded=pd.get_dummies(df,columns=["color"])
df_encoded
```

Out[9]:

	color_blue	color_green	color_red
0	False	False	True
1	True	False	False
2	False	True	False
3	True	False	False

In [11]:

```
import matplotlib.pyplot as plt
emp_age=[12,43,24,23,32,32,22,24,22,64]
bins=[0,10,20,30,40,50,60]
plt.hist(emp_age, bins, histtype='bar', rwidth=0.5, color="purple")
plt.xlabel("Employee Ages")
plt.ylabel("No. of employee")
plt.title("PU")
plt.legend("EMPLOYEE")
plt.show()
```

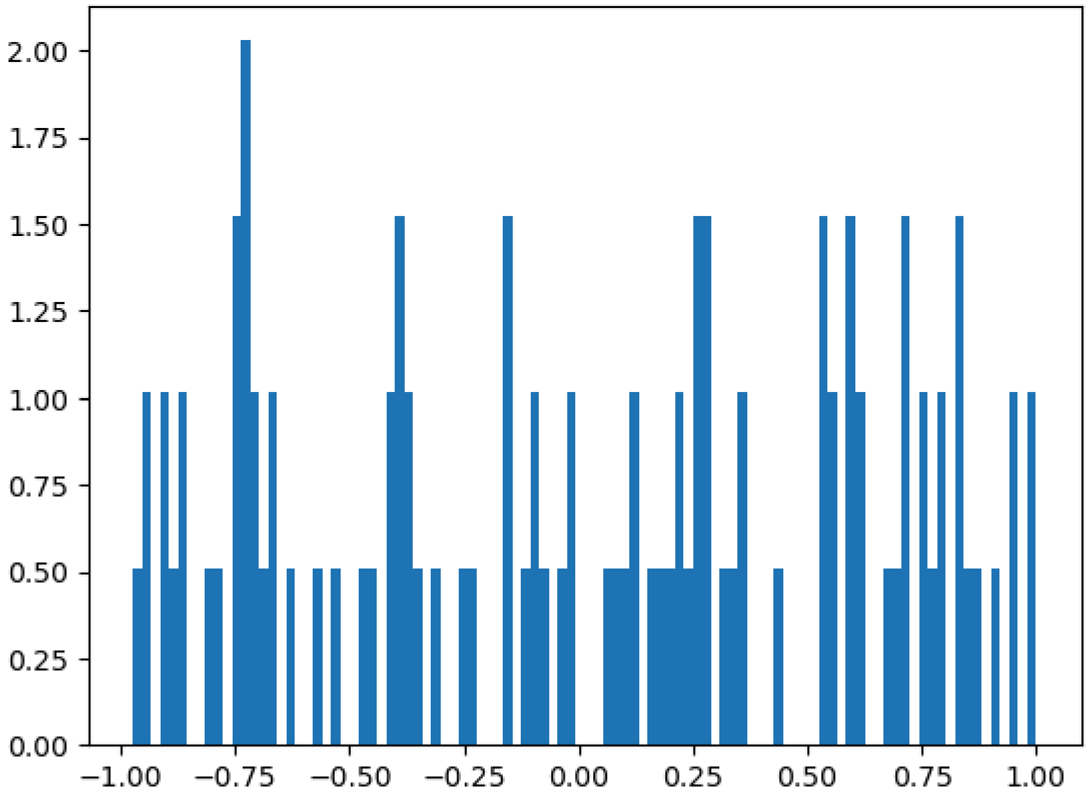


In [13]:

```
import matplotlib.pyplot as plt
import numpy as np
dist_uni=np.random.uniform(-1,1,100)
dist_uni
```

```
Out[13]: array([-0.72183504, -0.55990682,  0.21629023,  0.12735724,  0.99334715,
                -0.53699345, -0.71708397,  0.77239753, -0.23003726,  0.8547013 ,
                 0.2033258 ,  0.58934182,  0.9148132 , -0.80350387,  0.61987869,
                 0.7005186 ,  0.78468735, -0.38532845,  0.21089677,  0.25192182,
                 0.2599792 , -0.01177755,  0.54877757, -0.93856775,  0.34381627,
                -0.97296623, -0.6620186 ,  0.6127079 ,  0.70816264, -0.24825528,
                 0.23557206,  0.67444814,  0.58729926,  0.36498897,  0.83887121,
                -0.37038337, -0.75262228, -0.11565851,  0.71198202, -0.86139848,
                -0.88349055, -0.90912956, -0.74440935, -0.39686877,  0.87222865,
                -0.71015483, -0.73943028,  0.535524  ,  0.53976587,  0.9990366 ,
                -0.41438087, -0.73487107,  0.32604482,  0.53287293, -0.71730629,
                -0.15131996,  0.05818538, -0.90431273, -0.41228498, -0.93448364,
                 0.79496645,  0.28064619,  0.94679536, -0.10051384,  0.56390493,
                 0.94804534, -0.38100228, -0.06999232, -0.10398811, -0.31575523,
                 0.83128697, -0.66049656, -0.01858452, -0.15792318,  0.12056611,
                 0.35600753,  0.75534713,  0.44153917,  0.078342  ,  0.09193191,
                 0.26441814, -0.6311181 ,  0.27907641, -0.86889708, -0.38246506,
                -0.47916563, -0.15658313, -0.03772857,  0.75704518,  0.83378684,
                 0.16026762, -0.71198839,  0.17835913,  0.59243597, -0.45076195,
                 0.71367139, -0.34511454, -0.69503653, -0.78176489,  0.27564991])
```

```
In [15]: plt.hist(dist_uni, bins = 100, density = True)
         plt.show()
```

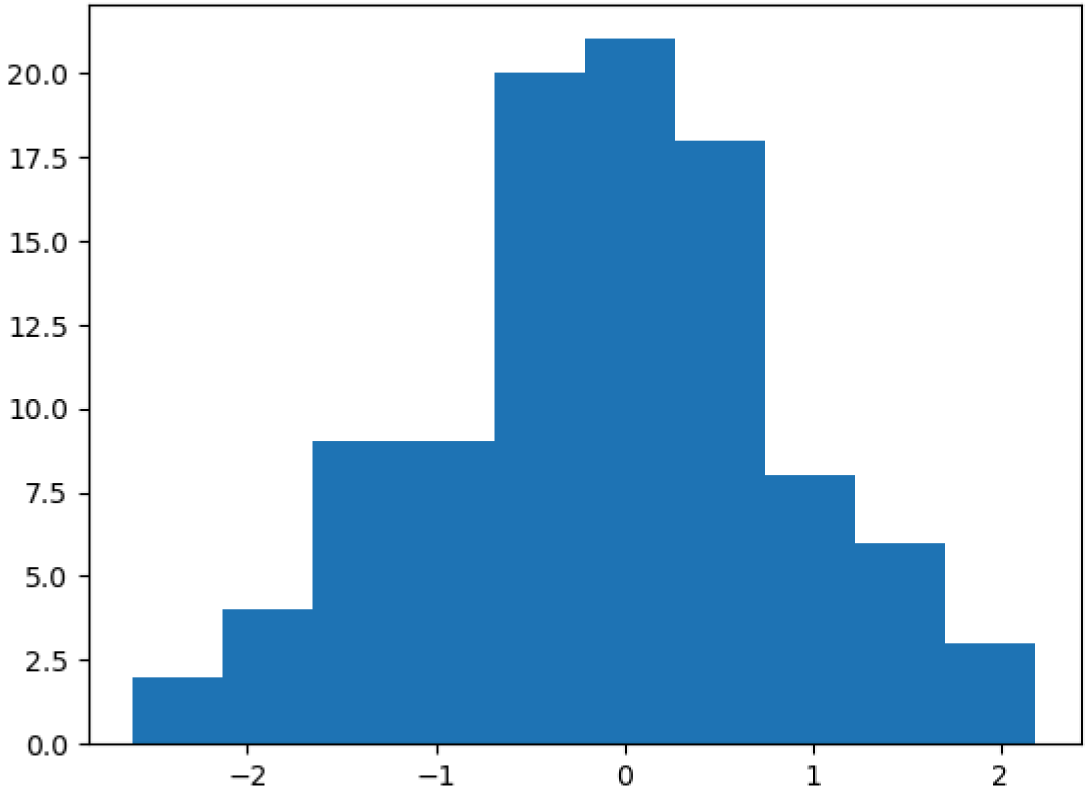


```
In [17]: dist_norm= np.random.normal(0.0, 1.0, 100)
         print(dist_norm)
```

```
[ 0.9275576  -0.44129745  0.07269806 -0.19545225 -1.61837331 -0.35392839
 -0.22742182  0.48650644 -1.66927086 -2.10690993  0.70660904 -0.78165194
  1.10924392 -0.58902716 -0.05147363 -2.60802948 -1.3382064  -0.03304735
 -0.7145957  -1.66305576  1.42589445  2.1849952   0.69150102  1.2904151
 -0.62324841 -0.1064719  -0.01438334 -1.25320023 -0.44385658  0.29019754
  0.36012994 -0.14150384  1.34502584 -0.73849774  0.63982886 -0.15051992
 -0.02117693  2.04619044  2.00091719  0.61064211  0.86719047  0.62057234
  0.42542422  0.61141679  0.74295863 -0.15085713  0.46300882  0.38577836
  0.55346704 -0.08811228 -0.87199853  0.31755283 -0.30544602 -0.99000303
  1.35384853  0.80740873  0.98060749 -0.54517848 -1.4449631  -0.43259159
 -0.45806597 -0.32045144  0.35930157  0.89079883 -0.8753588   0.63009418
 -1.20853268 -0.68642972 -2.36266491  0.06126315 -1.68387433 -1.09885964
 -0.43034168 -1.20906422 -0.6599875  -0.67535835 -1.59255827 -1.40325069
 -0.2815267  -0.39717126 -0.36330551  0.15456708 -0.88109399 -1.2136703
  0.1006959   0.3099726   0.74715345  0.17050091  0.17668267 -0.78086624
 -0.45812349 -0.00773726  0.10134638  1.59212372  1.07011228  0.22535958
  1.42015329  0.0409541  -0.09065781 -0.24196255]
```

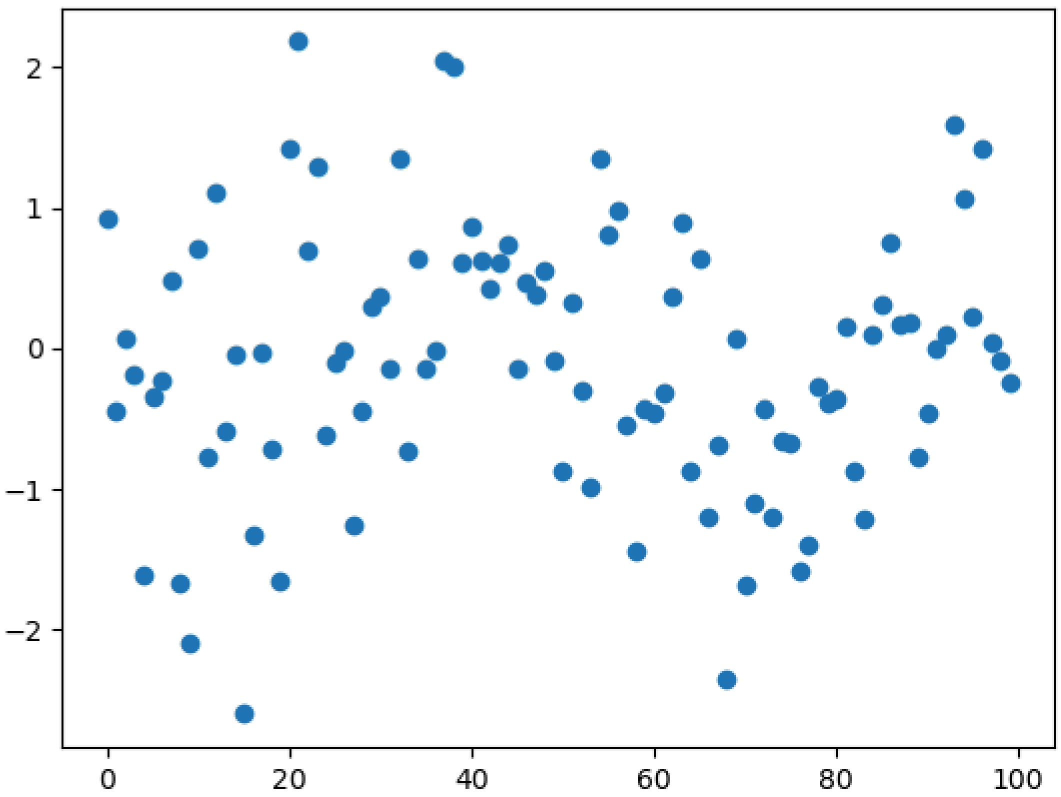
In [19]:

```
plt.hist(dist_norm)
plt.show()
```



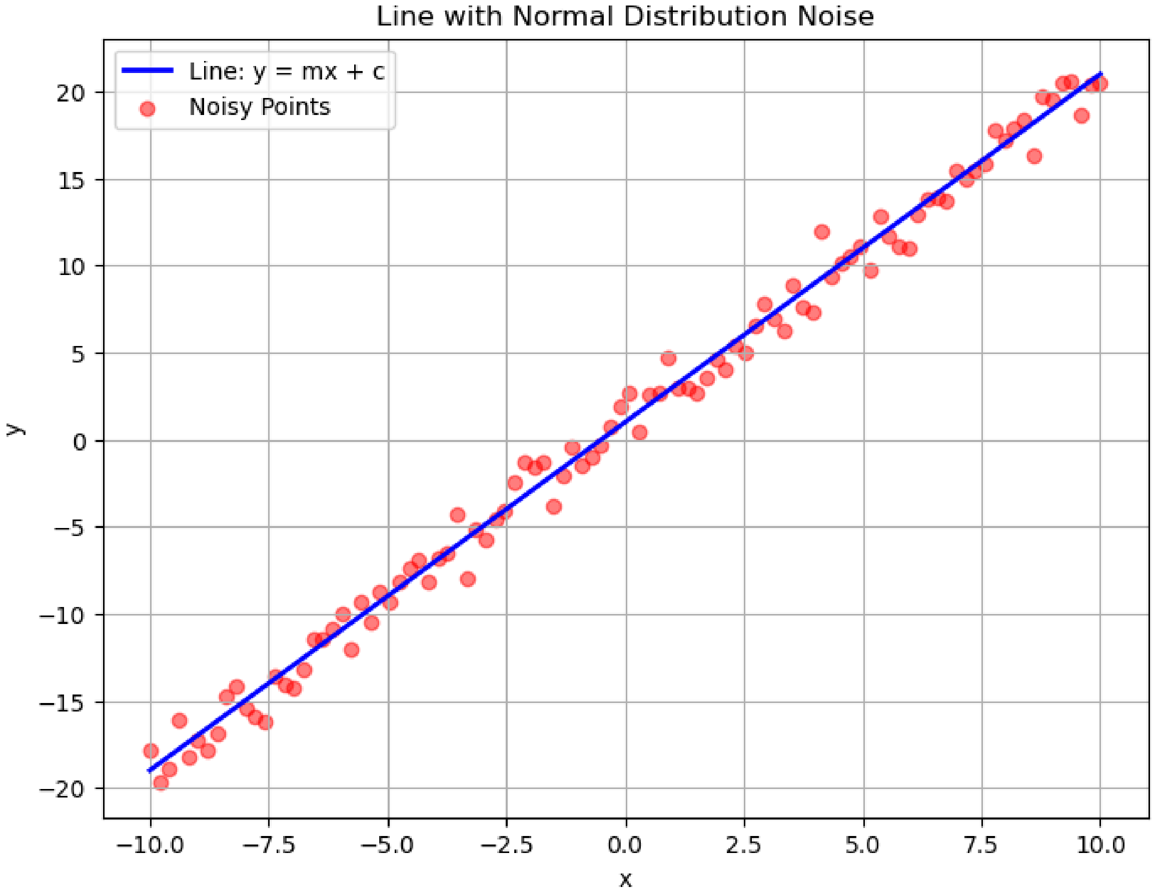
In [21]:

```
x=np.arange(100)
y=dist_norm
plt.scatter(x,y)
plt.show()
```



In [23]:

```
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt

# Parameters for the line equation y = mx + c
m = 2  # slope
c = 1  # intercept

# Parameters for the normal distribution noise
mean = 0  # mean of normal distribution
std_dev = 1  # standard deviation of normal distribution

# Generate x values (e.g., from -10 to 10)
x_values = np.linspace(-10, 10, 100)

# Generate the corresponding y values for the line y = mx + c
y_values_line = m * x_values + c

# Generate noise based on normal distribution and add to y values
noise = np.random.normal(mean, std_dev, size=x_values.shape)
y_values_with_noise = y_values_line + noise

# Plotting
plt.figure(figsize=(8, 6))
plt.plot(x_values, y_values_line, label='Line: y = mx + c', color='blue', linewi
plt.scatter(x_values, y_values_with_noise, color='red', label='Noisy Points', al
plt.title('Line with Normal Distribution Noise')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.grid(True)
plt.show()
```


Line with Normal Distribution Noise
Line: y = mx + c
Noisy Points
20
15
10
5
0
−5
−10
−15
−20
y
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
x