There are essentially two ways to use Matplotlib: 1) Explicitly create figures and axes, and call methods on them (the 'object - oriented 'style,(o-o style)

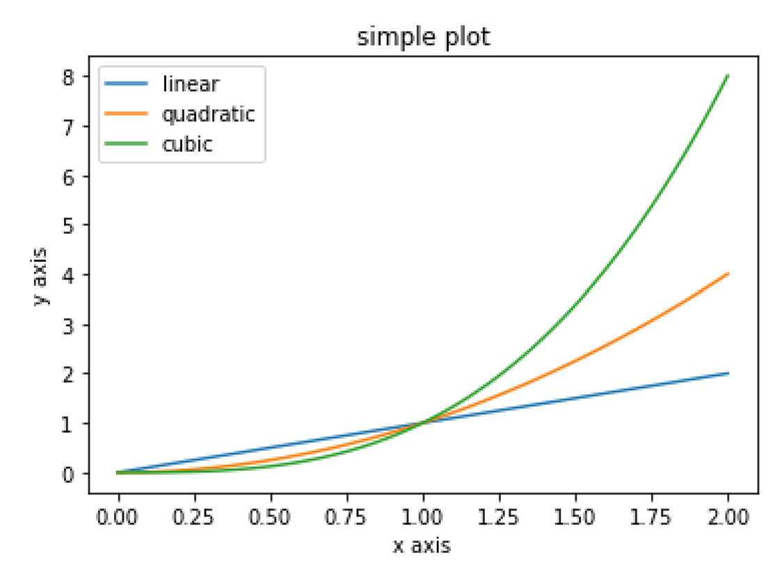
```
2) Rely on pyplot to automatically create and manage the figues and axes,
and use pyplot functions for plotting
```

In [49]:
```python
# object - oriented style
```

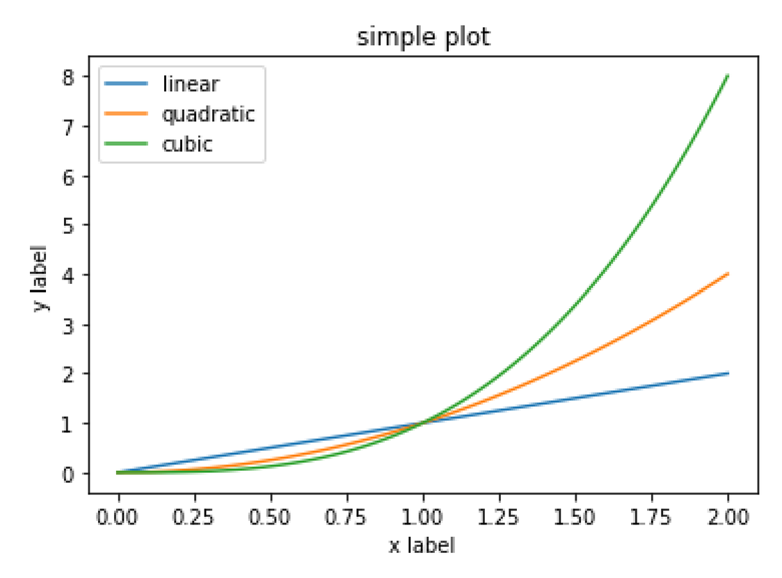
In [1]:
```python
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```python
x = np.linspace(0,2,100) # to generate sample data

fig, ax = plt.subplots() #Create a figure and an axes
ax.plot(x,x,label='linear') # x-axis = x, y-axis = x
ax.plot(x,x**2,label='quadratic')
ax.plot(x,x**3,label='cubic')
ax.set_xlabel('x axis')
ax.set_ylabel('y axis')
ax.set_title('simple plot')
ax.legend()
```

Out[2]: <matplotlib.legend.Legend at 0x1c748e936d8>

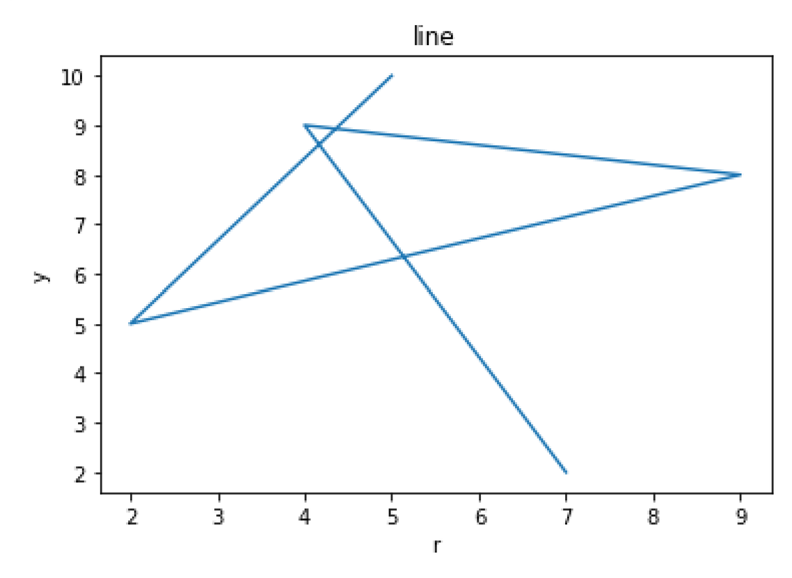


In [3]:
```python
## pyplot functions
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0,2,100)

plt.plot(x,x,label='linear')
plt.plot(x,x**2,label='quadratic')
plt.plot(x,x**3,label='cubic')
plt.xlabel('x label')
plt.ylabel('y label')
plt.title("simple plot")
plt.legend()
plt.show()
```



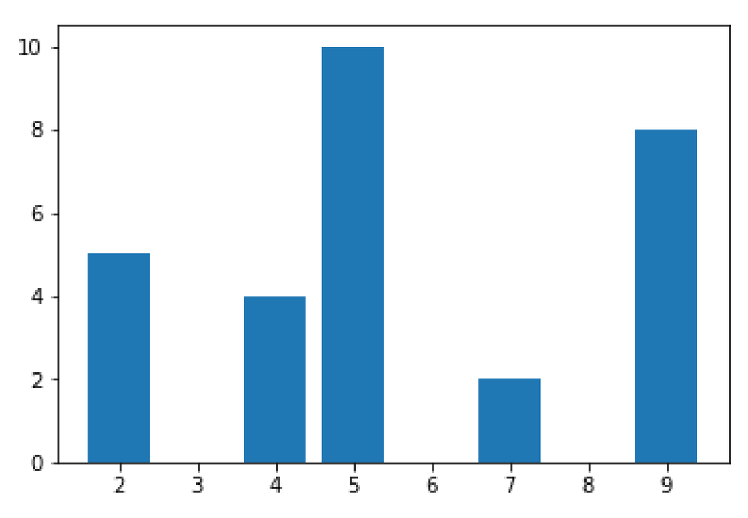
Line Plot

In [5]:
```python
import matplotlib.pyplot as plt

# x- axis values
x = [5,2,9,4,7]

# y - axis values
y = [10,5,8,9,2]

# function to plot
plt.plot(x,y) # plot give line graph
plt.title('line')
plt.xlabel('r')
plt.ylabel('y')

# function to show the plot
plt.show()
```



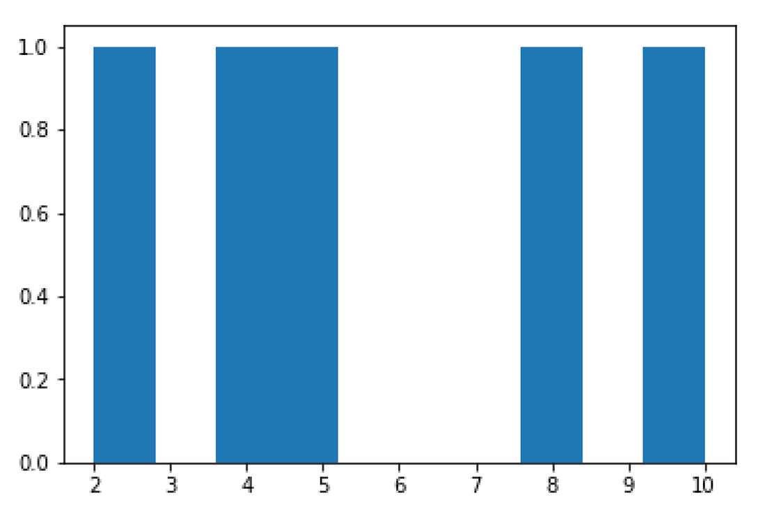
Bar plot

In [6]:
```python
x = [5,2,9,4,7]
y = [10,5,8,4,2]
plt.bar(x,y)
plt.show()
```



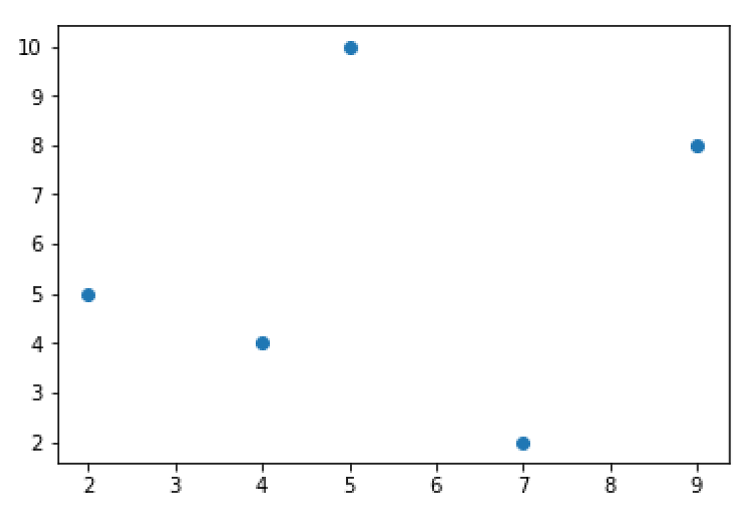
Histogram

In [7]:
```python
# y - axis values
y = [10,5,8,4,2]

plt.hist(y)
plt.show()
```

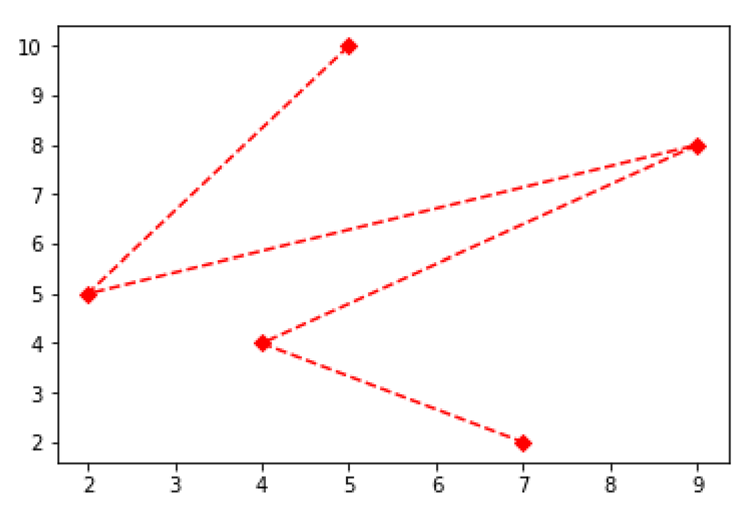


scatter plot

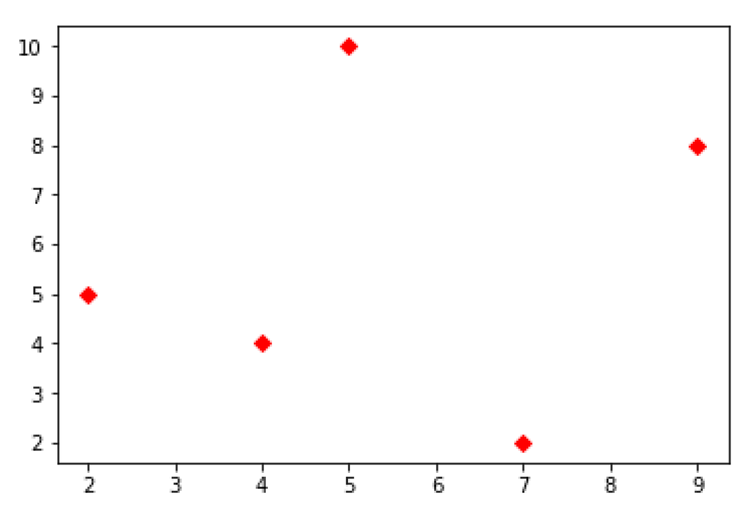
In [8]:
```python
x = [5,2,9,4,7]
y = [10,5,8,4,2]

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



In [10]:
```python
x = [5,2,9,4,7]
y = [10,5,8,4,2]

plt.plot(x,y, marker = 'D', color = 'r', linestyle = '--')
plt.show()
```

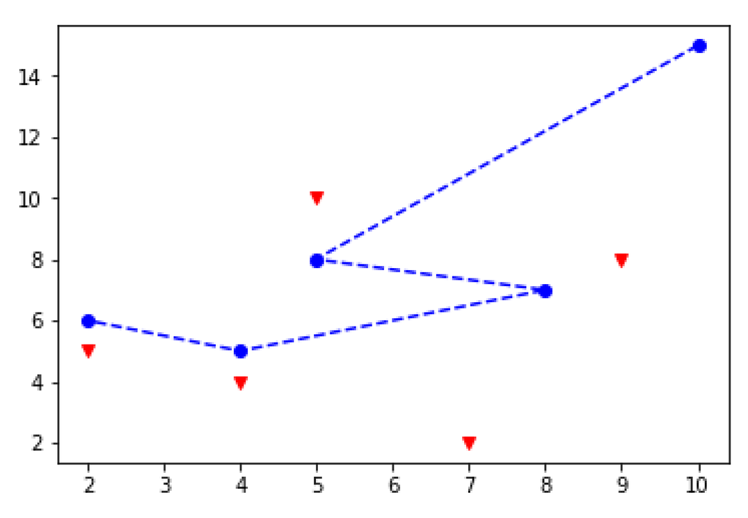


In [12]:
```python
x = [5,2,9,4,7]
y = [10,5,8,4,2]

plt.plot(x,y, marker = 'D', color = 'r', linestyle = "None")  # line plot
plt.show()
```

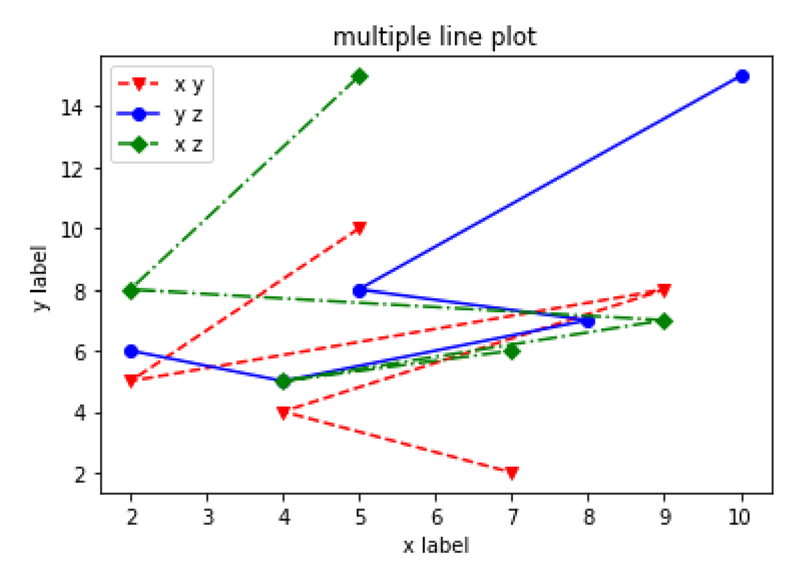


multiple line plots

In [13]:
```python
x = [5,2,9,4,7]
y = [10,5,8,4,2]
z = [15,8,7,5,6]

plt.plot(x,y,marker='v',color='r',linestyle="None")
plt.plot(y,z,marker='o',color='b',linestyle='--')

plt.show()
```

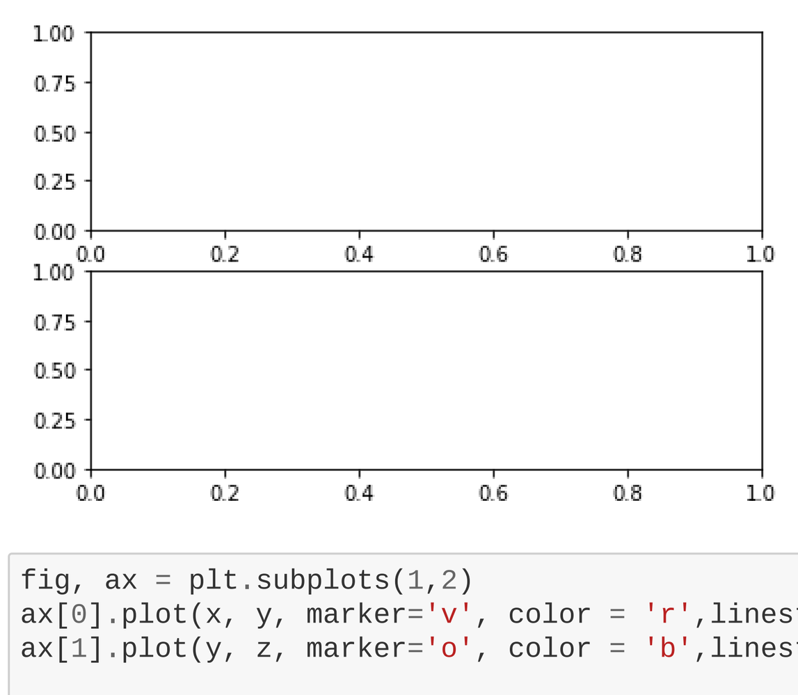


In [17]:
```python
plt.plot(x,y,marker='v',color='r',linestyle='--',label='x y')
plt.plot(y,z,marker='o',color='b',label='y z')
plt.plot(x,z,marker='D',color='g',linestyle='-.',label='x z')

plt.xlabel('x label')
plt.ylabel('y label')
plt.title('multiple line plot')
plt.legend()
plt.show()
```



In [19]:
```python
#plt.subplots()
fig, ax = plt.subplots(2,1)
plt.subplots
```

Out[19]: <function matplotlib.pyplot.subplots(nrows=1, ncols=1, sharex=False, sharey=False, squeeze=True, subplot_kw=None, gridspec_kw=None, **fig_kw)>



In [21]:
```python
fig, ax = plt.subplots(1,2)
ax[0].plot(x, y, marker='v', color = 'r',linestyle='--')
ax[1].plot(y, z, marker='o', color = 'b',linestyle='--')

ax[0].set_xlabel('x label')
ax[1].set_ylabel('y label')
ax[1].set_title('simple')

plt.show()
```