

In [2]:

```
import numpy as np
```

line-plot

In [3]:

```
from matplotlib import pyplot as plt
```

In [4]:

```
print(dir(plt))
```

```
['Annotation', 'Arrow', 'Artist', 'AutoLocator', 'Axes', 'Button', 'Circle',
'Figure', 'FigureCanvasBase', 'FixedFormatter', 'FixedLocator', 'FormatStrFo
rmatter', 'Formatter', 'FuncFormatter', 'GridSpec', 'IndexLocator', 'Line2
D', 'LinearLocator', 'Locator', 'LogFormatter', 'LogFormatterExponent', 'Log
FormatterMathtext', 'LogLocator', 'MaxNLocator', 'MultipleLocator', 'Normali
ze', 'NullFormatter', 'NullLocator', 'Number', 'PolarAxes', 'Polygon', 'Rect
angle', 'ScalarFormatter', 'Slider', 'Subplot', 'SubplotTool', 'Text', 'Tick
Helper', 'Widget', '_INSTALL_FIG_OBSERVER', '_IP_REGISTERED', '__builtins_
__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__packag
e__', '__spec__', '_auto_draw_if_interactive', '_backend_mod', '_get_running
_interactive_framework', '_interactive_bk', '_log', '_pylab_helpers', '_set
p', '_setup_pyplot_info_docstrings', '_show', 'acorr', 'angle_spectrum', 'an
notate', 'arrow', 'autoscale', 'autumn', 'axes', 'axhline', 'axhspan', 'axi
s', 'axvline', 'axvspan', 'bar', 'barbs', 'barh', 'bone', 'box', 'boxplot',
'broken_barh', 'cbook', 'cla', 'clabel', 'clf', 'clim', 'close', 'cm', 'cohe
re', 'colorbar', 'colormaps', 'connect', 'contour', 'contourf', 'cool', 'cop
per', 'csd', 'cycler', 'dedent', 'delaxes', 'deprecated', 'disconnect', 'doc
string', 'draw', 'draw_all', 'draw_if_interactive', 'errorbar', 'eventplot',
'figaspect', 'figimage', 'figlegend', 'fignum_exists', 'figtext', 'figure',
'fill', 'fill_between', 'fill_betweenx', 'findobj', 'flag', 'functools', 'gc
a', 'gcf', 'gci', 'get', 'get_backend', 'get_cmap', 'get_current_fig_manage
r', 'get_figlabels', 'get_fignums', 'get_plot_commands', 'get_scale_docs',
'get_scale_names', 'getp', 'ginput', 'gray', 'grid', 'hexbin', 'hist', 'hist
2d', 'hlines', 'hot', 'hsv', 'importlib', 'imread', 'imsave', 'imshow', 'inf
erno', 'inspect', 'install_repl_displayhook', 'interactive', 'ioff', 'ion',
'isinteractive', 'jet', 'legend', 'locator_params', 'logging', 'loglog', 'ma
gma', 'magnitude_spectrum', 'margins', 'matplotlib', 'matshow', 'minorticks_
off', 'minorticks_on', 'mlab', 'new_figure_manager', 'nipy_spectral', 'np',
'pause', 'pcolor', 'pcolormesh', 'phase_spectrum', 'pie', 'pink', 'plasma',
'plot', 'plot_date', 'plotfile', 'plotting', 'polar', 'prism', 'psd', 'pylab
_setup', 'quiver', 'quiverkey', 'rc', 'rcParams', 'rcParamsDefault', 'rcPara
msOrig', 'rc_context', 'rcdefaults', 'rcsetup', 're', 'register_cmap', 'rgri
ds', 'savefig', 'sca', 'scatter', 'sci', 'semilogx', 'semilogy', 'set_cmap',
'set_loglevel', 'setp', 'show', 'silent_list', 'specgram', 'spring', 'spy',
'stackplot', 'stem', 'step', 'streamplot', 'style', 'subplot', 'subplot2gri
d', 'subplot_tool', 'subplots', 'subplots_adjust', 'summer', 'suptitle', 'sw
itch_backend', 'sys', 'table', 'text', 'thetagrids', 'tick_params', 'ticklab
el_format', 'tight_layout', 'time', 'title', 'tricontour', 'tricontourf', 't
ripcolor', 'tripplot', 'twinx', 'twiny', 'uninstall_repl_displayhook', 'violi
nplot', 'viridis', 'vlines', 'waitforbuttonpress', 'warn_deprecated', 'winte
r', 'xcorr', 'xkcd', 'xlabel', 'xlim', 'xscale', 'xticks', 'ylabel', 'ylim',
'yscale', 'yticks']
```

In [5]:

```
x=np.arange(1,11)  
x
```

Out[5]:

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [6]:

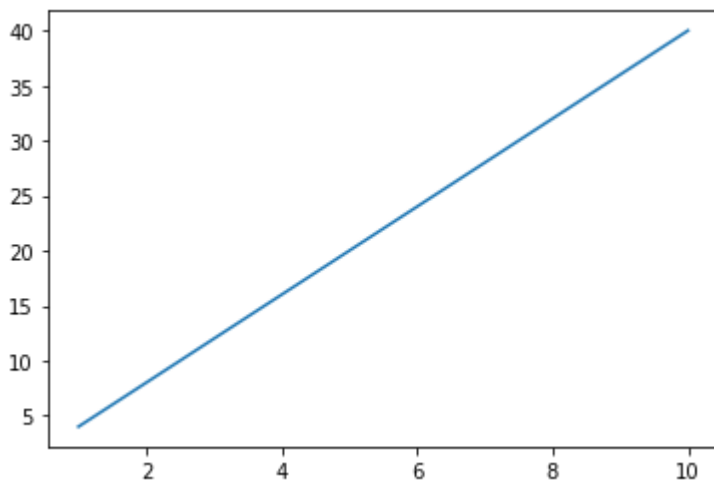
```
y=x*4  
y
```

Out[6]:

```
array([ 4,  8, 12, 16, 20, 24, 28, 32, 36, 40])
```

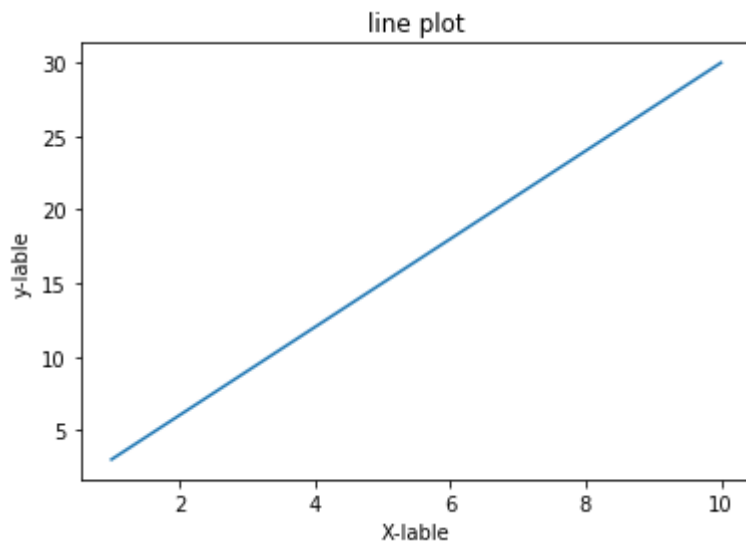
In [7]:

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



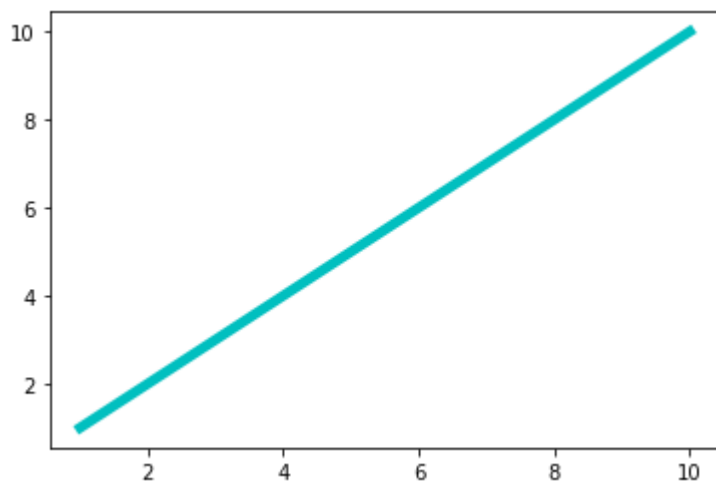
In [9]:

```
plt.plot(x,y)
plt.title("line plot")
plt.xlabel("X-lable")
plt.ylabel("y-lable")
plt.show()
```



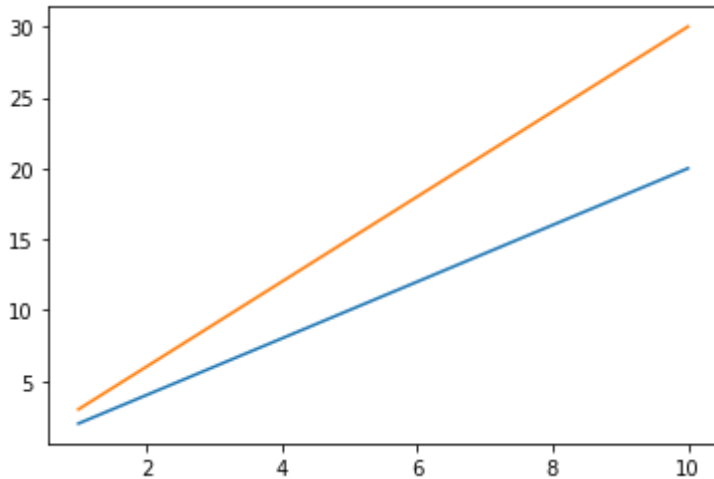
In [9]:

```
plt.plot(x,x,color='c',linestyle='-',linewidth=5)
plt.show()
```



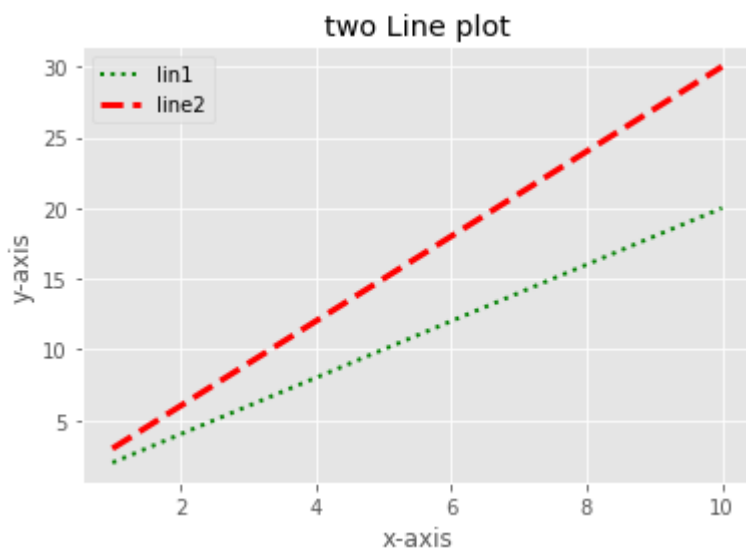
In [8]:

```
x=np.arange(1,11)
y1=2*x
y2=3*x
plt.plot(x,y1)
plt.plot(x,y2)
plt.show()
```



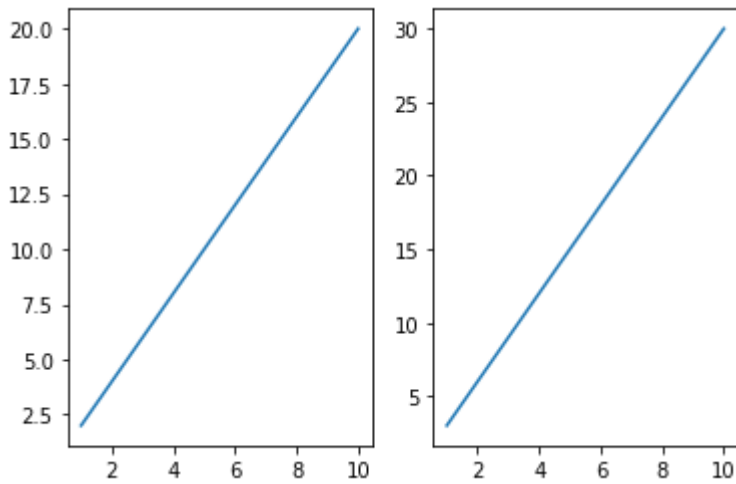
In [13]:

```
plt.plot(x,y1,color='g',linestyle=':',linewidth=2,label='lin1')
plt.plot(x,y2,color='r',linestyle='--',linewidth=3,label='line2')
plt.title("two Line plot")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.legend() #include Legend Line 1 and Line 2
plt.grid(True)
plt.show()
```



In [21]:

```
x=np.arange(1,11)
y1=2*x
y2=3*x
plt.subplot(1,2,1)
plt.plot(x,y1)
plt.subplot(1,2,2)
plt.plot(x,y2)
plt.show()
```



In [24]:

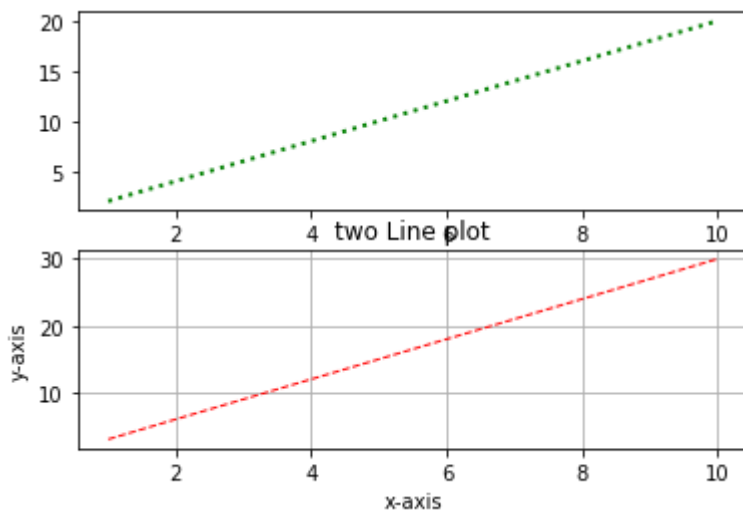
```

x=np.arange(1,11)
y1=2*x
y2=3*x
plt.subplot(2,1,1)
plt.plot(x,y1,color='g',linestyle=':',linewidth=2)

plt.subplot(2,1,2)
plt.plot(x,y2,color='r',linestyle='--',linewidth=1)
plt.title("two Line plot")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.grid(True)

plt.show()

```



Bar Plot

In [25]:

```
student={"BOB":87,"Julia":76,"Anne":99,"Matt":45}
```

In [26]:

```
name=list(student.keys())
name
```

Out[26]:

```
['BOB', 'Julia', 'Anne', 'Matt']
```

In [27]:

```
marks=list(student.values())
marks
```

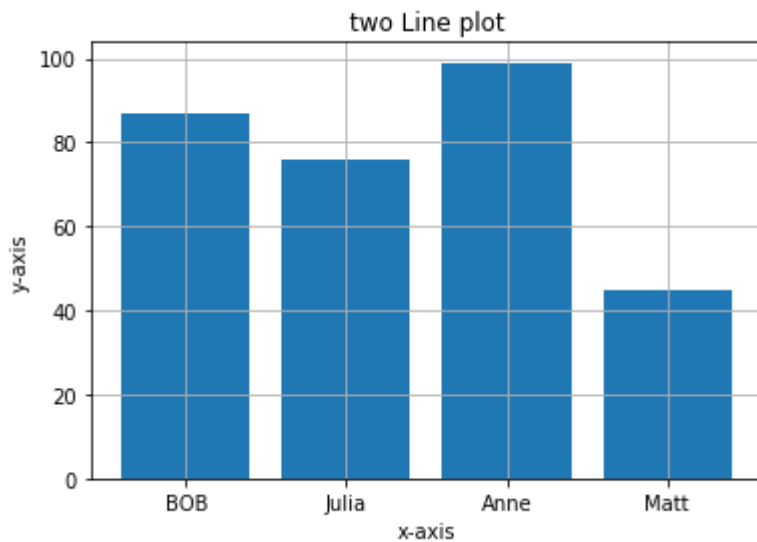
Out[27]:

```
[87, 76, 99, 45]
```

In [32]:

```
plt.bar(name,marks)
plt.title("two Line plot")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.grid(True)

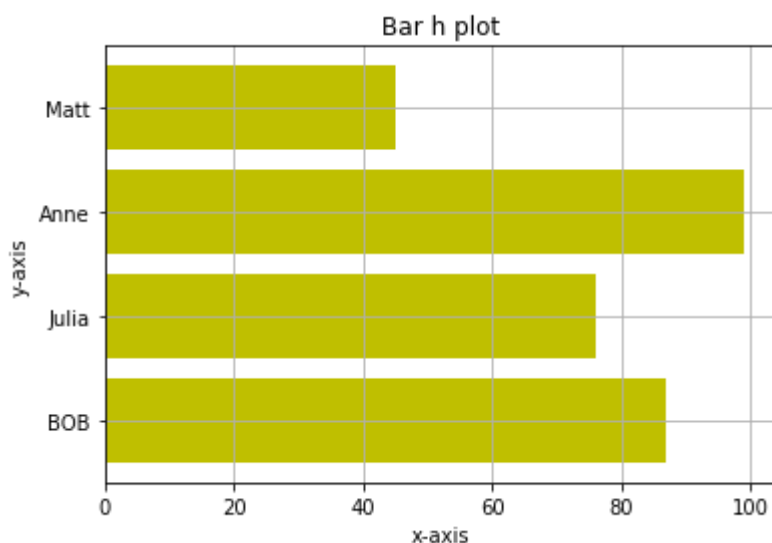
plt.show()
```



In [35]:

```
plt.barh(name,marks,color='y')
plt.title("Bar h plot")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.grid(True)

plt.show()
```



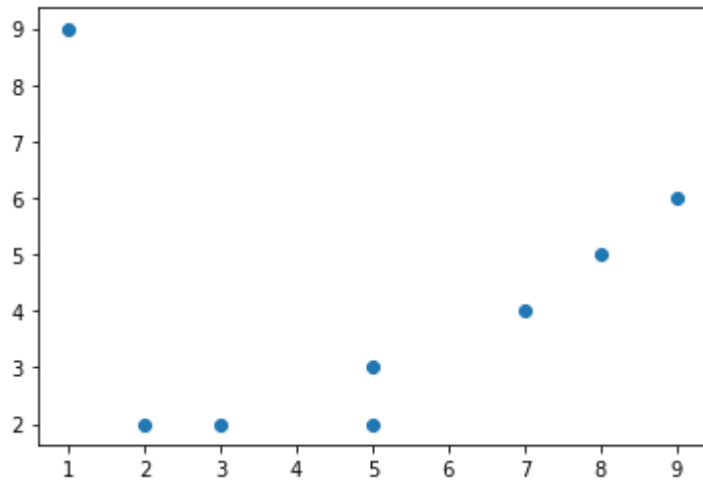
Scatter plot

In [36]:

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

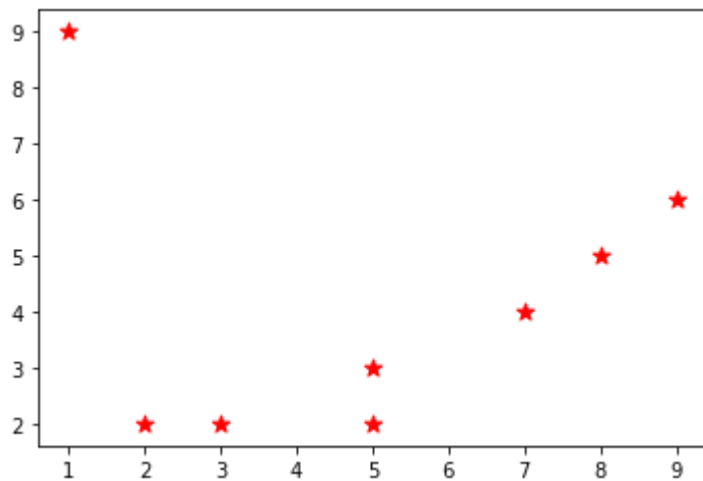
In [37]:

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



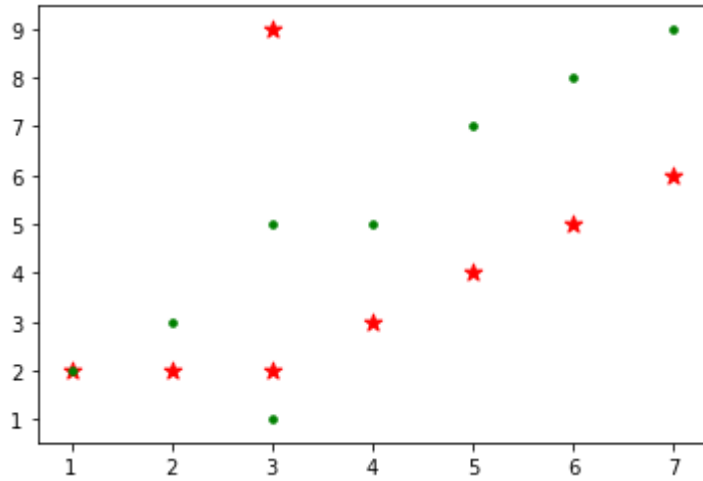
In [41]:

```
plt.scatter(x,y,marker='*',c='r',s=80)  
plt.show()
```



In [43]:

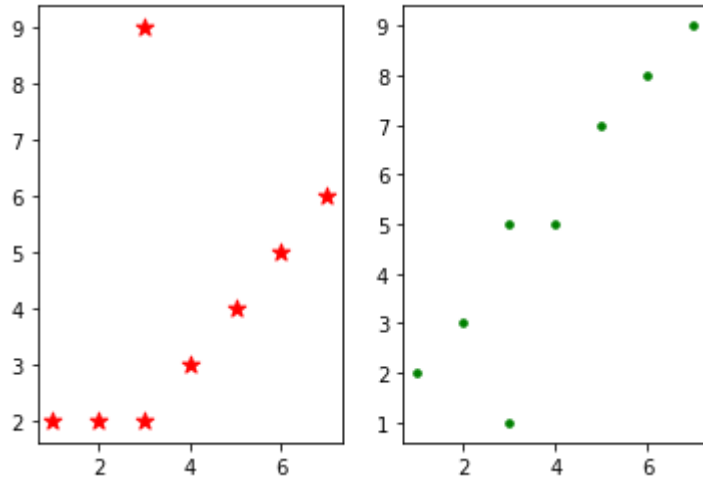
```
a=[1,2,3,4,5,6,7,3]
x=[2,3,5,5,7,8,9,1]
y=[2,2,2,3,4,5,6,9]
plt.scatter(a,y,marker='*',c='r',s=80)
plt.scatter(a,x,marker='.',c='g',s=50)
plt.show()
```



In [45]:

```
plt.subplot(1,2,1)
plt.scatter(a,y,marker='*',c='r',s=80)

plt.subplot(1,2,2)
plt.scatter(a,x,marker='.',c='g',s=50)
plt.show()
```



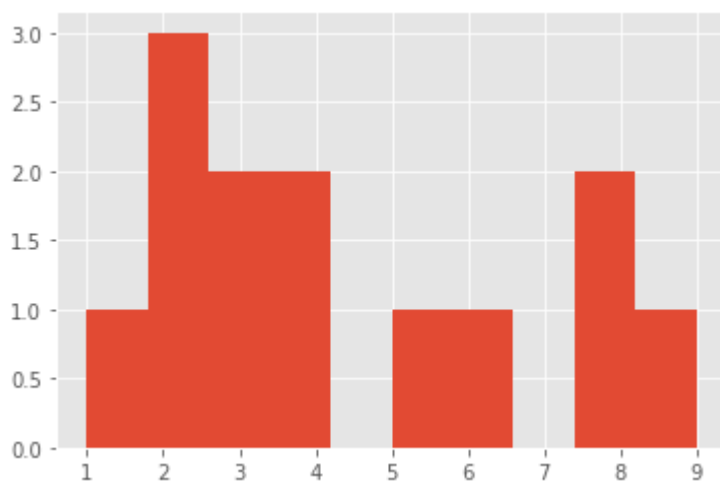
Histogram plot

In [46]:

```
#bar-catogorical value .... histogram-numerical value
```

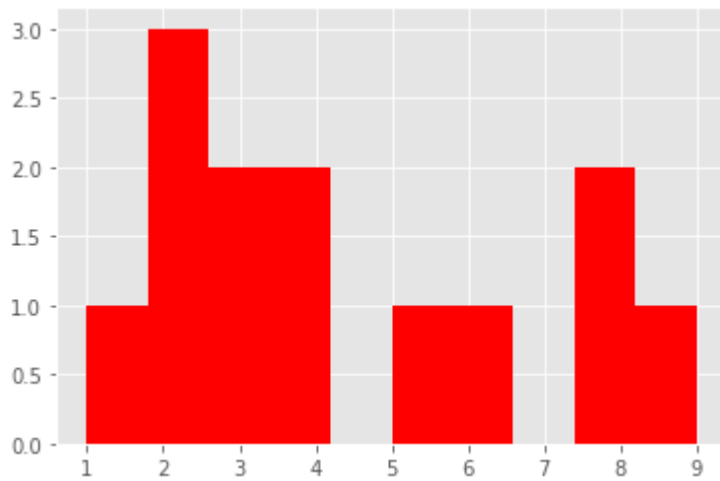
In [15]:

```
data=[1,2,3,4,6,9,5,3,2,8,4,8,2]
plt.hist(data)
plt.show()
```



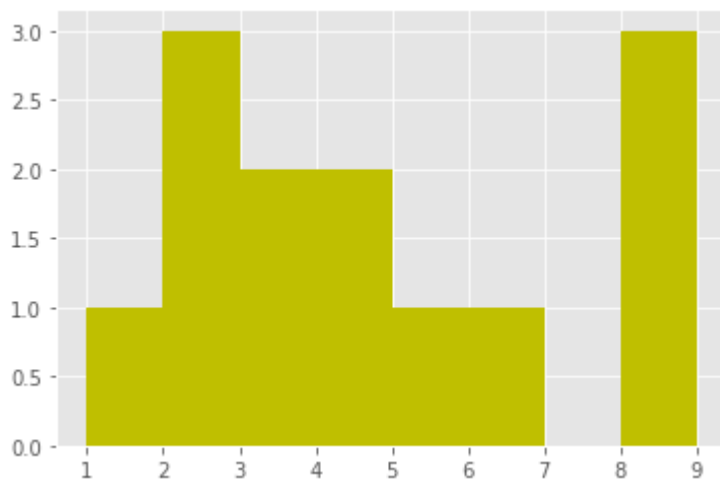
In [16]:

```
plt.hist(data,color='r')  
plt.show()
```



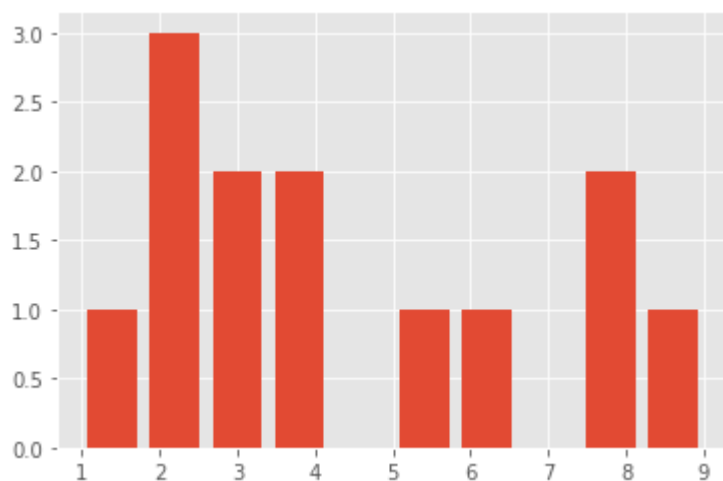
In [19]:

```
plt.hist(data,color='y',bins=8)  
plt.show()
```



In [23]:

```
plt.hist(data,histtype='bar',rwidth=0.8) #width off bars
plt.show()
```



In [5]:

```
import pandas as pd
from sklearn.datasets import load_iris
dataset=load_iris()
df=pd.DataFrame(dataset['data'],columns=dataset['feature_names'])
df['target']=dataset['target']
df.head(5)
```

Out[5]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [8]:

```
plt.hist(df['sepal_lenght'],bins=30) #name error
plt.show()
```

```
-----
KeyError                                Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    2645         try:
-> 2646             return self._engine.get_loc(key)
    2647         except KeyError:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()
```

KeyError: 'sepal_lenght'

During handling of the above exception, another exception occurred:

```
KeyError                                Traceback (most recent call last)
<ipython-input-8-f8f0b3ac24cd> in <module>
----> 1 plt.hist(df['sepal_lenght'],bins=30) #name error
      2 plt.show()

~\anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
    2798         if self.columns.nlevels > 1:
    2799             return self._getitem_multilevel(key)
-> 2800         indexer = self.columns.get_loc(key)
    2801         if is_integer(indexer):
    2802             indexer = [indexer]

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    2646         return self._engine.get_loc(key)
    2647         except KeyError:
-> 2648         return self._engine.get_loc(self._maybe_cast_indexer(key))
    2649         indexer = self.get_indexer([key], method=method, tolerance=tolerance)
    2650         if indexer.ndim > 1 or indexer.size > 1:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'sepal_lenght'
```

In [96]:

```

from sklearn.datasets import load_wine
import pandas as pd
dataset=load_wine()
df=pd.DataFrame(dataset['data'],columns=dataset['feature_names'])
df['target']=dataset['target']
df.head()

```

Out[96]:

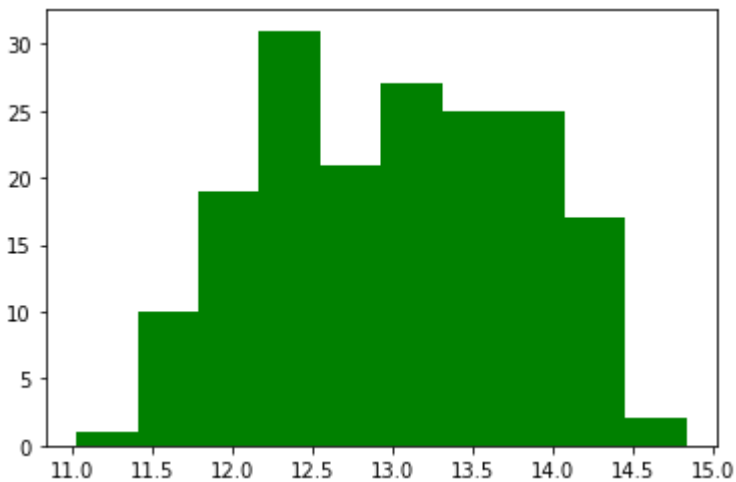
	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavan
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	

In [97]:

```

plt.hist(df['alcohol'],color='g')
plt.show()

```



box-plot

In [100]:

```

#fi number summery
#gives minimum,max, 25 ,50,75 percenticle value

```

In [103]:

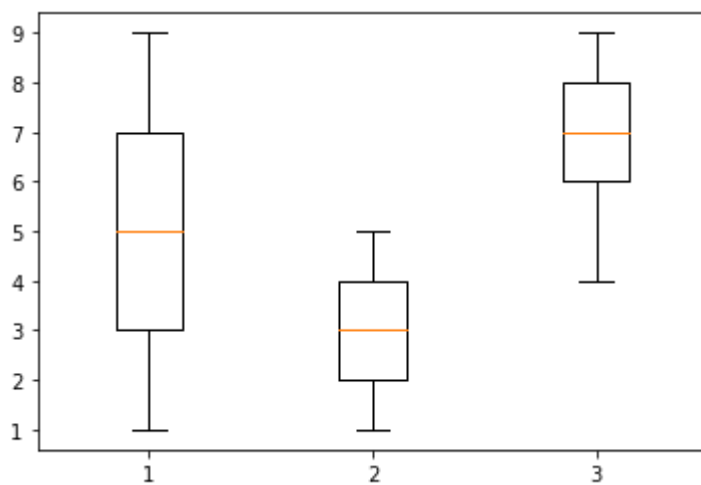
```
one = [1,2,3,4,5,6,7,8,9]
two = [1,2,3,4,5,4,3,2,1]
three = [6,7,8,9,8,7,6,5,4]
data=list([one,two,three])
data
```

Out[103]:

```
[[1, 2, 3, 4, 5, 6, 7, 8, 9],
 [1, 2, 3, 4, 5, 4, 3, 2, 1],
 [6, 7, 8, 9, 8, 7, 6, 5, 4]]
```

In [102]:

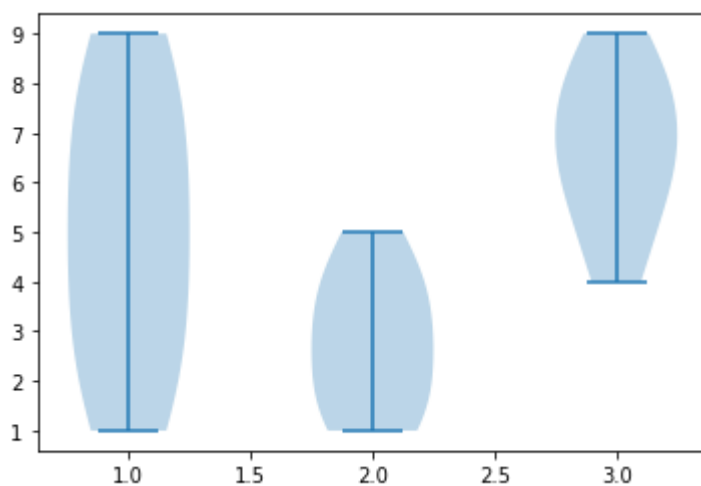
```
plt.boxplot(data)
plt.show()
```



violin plot

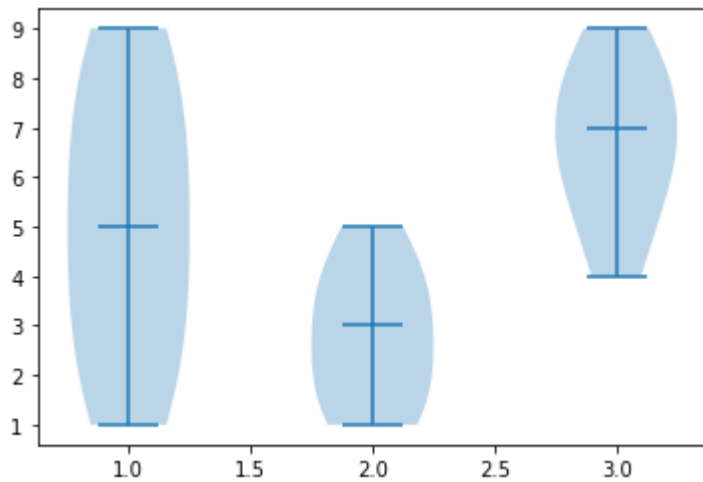
In [105]:

```
plt.violinplot(data)
plt.show()
```



In [106]:

```
plt.violinplot(data,showmedians=True)  
plt.show()
```



PIE CHART

In [10]:

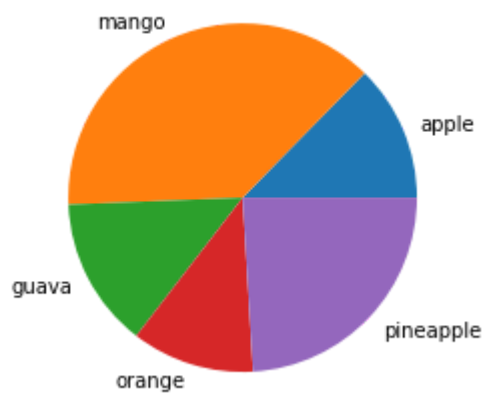
```
fruits = ['apple','mango','guava','orange','pineapple']
```

In [11]:

```
quantity = [10,30,11,9,19]
```

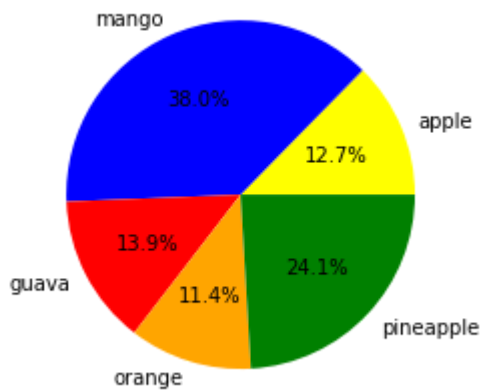
In [12]:

```
plt.pie(quantity,labels=fruits)  
plt.show()
```



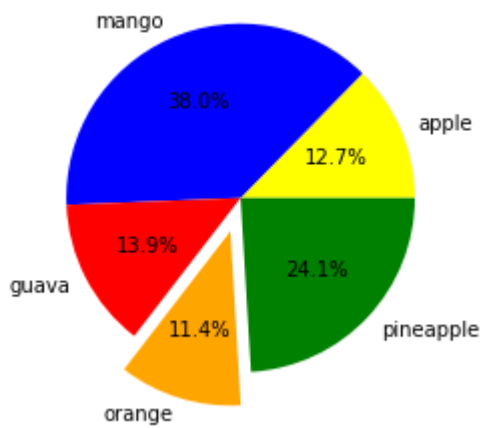
In [13]:

```
plt.pie(quantity, labels=fruits, autopct='%0.1f%%', colors=['yellow', 'blue', 'red', 'orange', 'green'],  
plt.show())
```



In [15]:

```
labels=fruits, autopct='%0.1f%%', colors=['yellow', 'blue', 'red', 'orange', 'green'], explode=(0, 0.1)
```



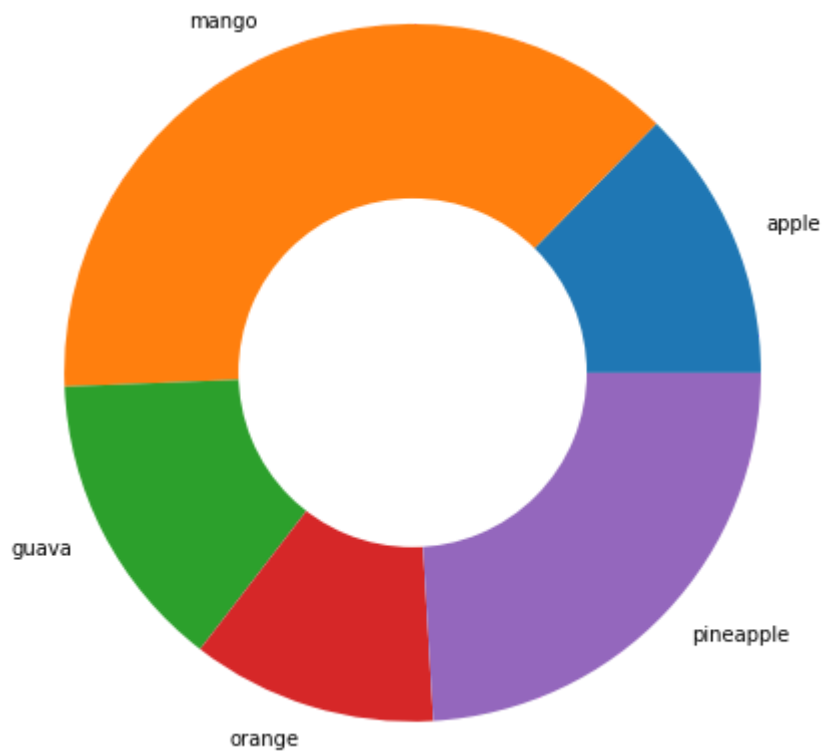
doughNut chart

In [118]:

```
#same as pie
```

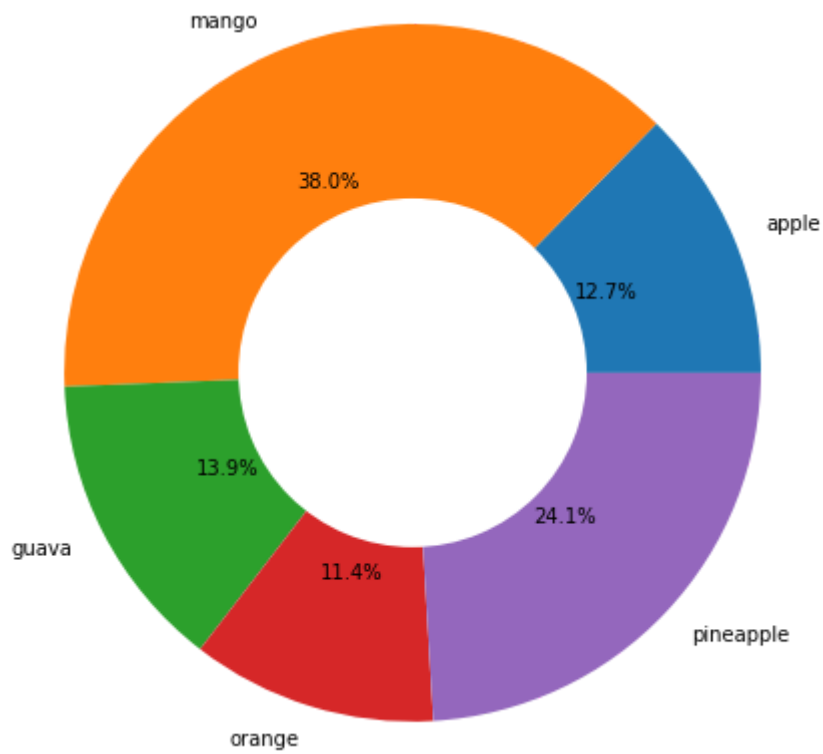
In [119]:

```
plt.pie(quantity, labels=fruits, radius=2)  
plt.pie([1], colors=['w'], radius=1) #color of inner circle white  
plt.show()
```



In [120]:

```
plt.pie(quantity, labels=fruits, autopct='%0.1f%%', radius=2)  
plt.pie([1], colors=['w'], radius=1) #color of inner circle white  
plt.show()
```



In []: