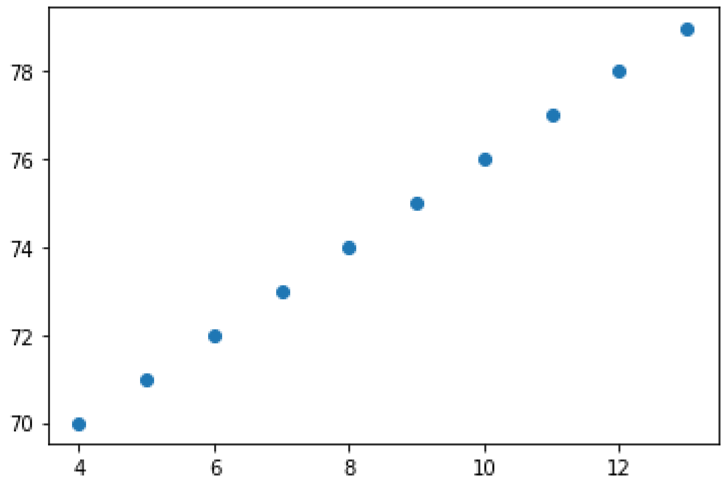
Matplotlib

- plotting library
- numerical extension: numpy
- inline plotting comfort

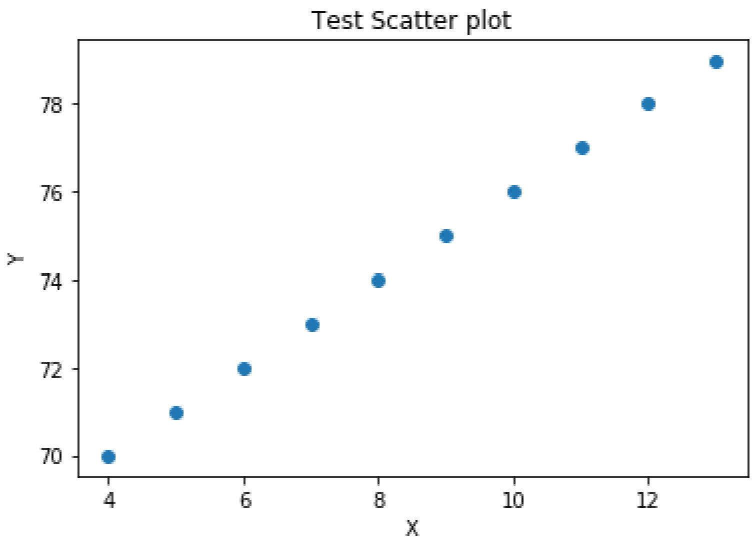
In [1]:
```
import matplotlib.pyplot as plt

%matplotlib inline

import pandas as pd
import numpy as np
```

In [5]:
```
plt.scatter(np.arange(4,14), np.arange(70,80))
```

Out[5]:
```
<matplotlib.collections.PathCollection at 0x7356409cc8>
```

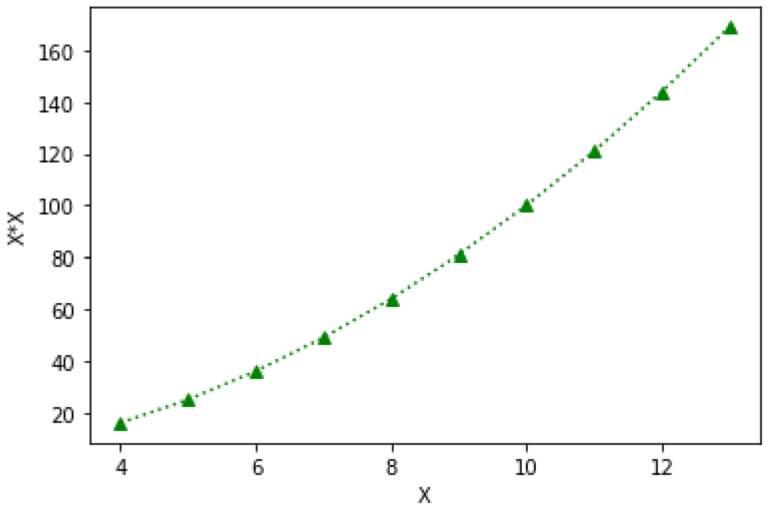


In [8]:
```
plt.scatter(np.arange(4,14), np.arange(70,80))
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Test Scatter plot")
plt.savefig('scatter.png')
```



In [31]:
```
#using plt.plot
import math as m

plt.plot(np.arange(4,14), np.arange(4,14)*np.arange(4,14), 'g^', linestyle='dotted')
plt.xlabel('X')
plt.ylabel('X*X')
```

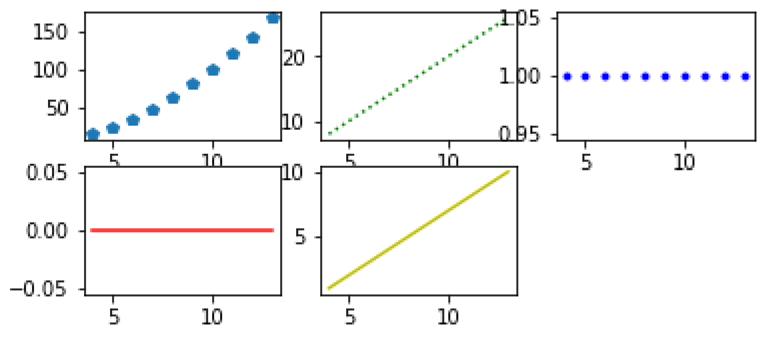
Out[31]:
```
Text(0, 0.5, 'X*X')
```



Subplots

In [43]:
```
plt.subplot(3,3,1)
plt.plot(x,x*x,'p')

plt.subplot(3,3,2)
plt.plot(x, x+x, 'g:')

plt.subplot(3,3,3)
plt.plot(x, x/x, 'b.')

plt.subplot(3,3,4)
plt.plot(x, x-x, 'r')


plt.subplot(3,3,5)
plt.plot(x, x-3, 'y')
```

Out[43]:
```
[<matplotlib.lines.Line2D at 0x735ce271c8>]
```

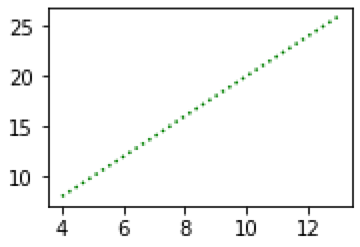


In [34]:
```
x
```

Out[34]:
```
array([ 4,  5,  6,  7,  8,  9, 10, 11, 12, 13])
```

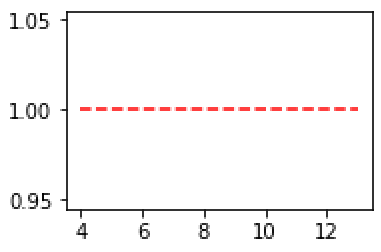
In [40]:
```
# If you dont run all the plot code snippets at one go, they will get plotted separately
plt.subplot(2,2,2)
plt.plot(x, x+x, 'g:')
```

Out[40]:
```
[<matplotlib.lines.Line2D at 0x735d1b1848>]
```



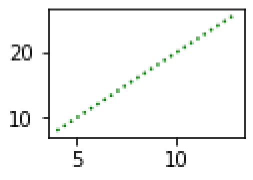
In [39]:
```
plt.subplot(2,2,3)
plt.plot(x, x/x, 'r--')
```

Out[39]:
```
[<matplotlib.lines.Line2D at 0x735d15efc8>]
```



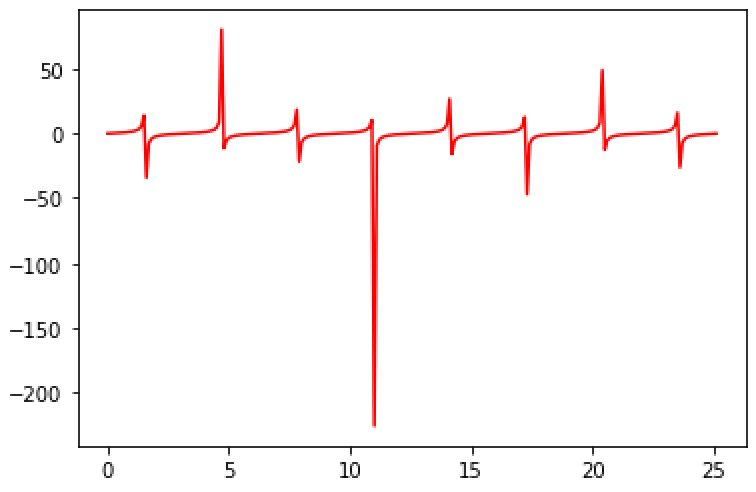
In [44]:
```
plt.subplot(3,3,2)
plt.plot(x, x+x, 'g:')
```

Out[44]:
```
[<matplotlib.lines.Line2D at 0x735d52cc88>]
```



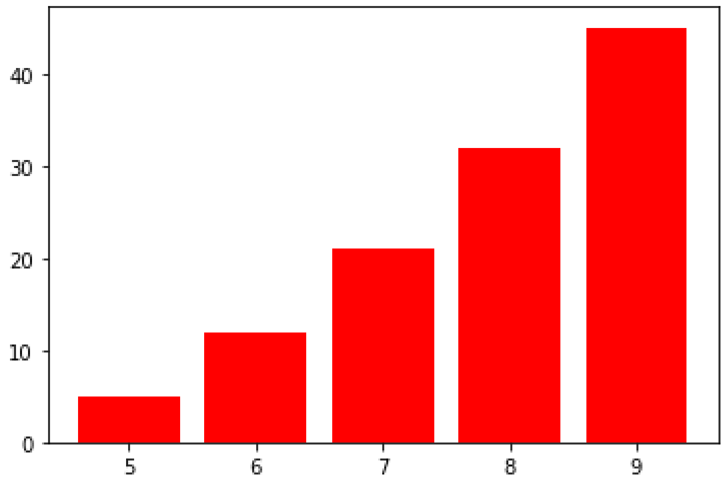
In [54]:
```
x= np.arange(0, 8*np.pi, 0.1)
y= np.tan(x)
plt.plot(x,y, 'r')
```

Out[54]:
```
[<matplotlib.lines.Line2D at 0x735e8bae08>]
```

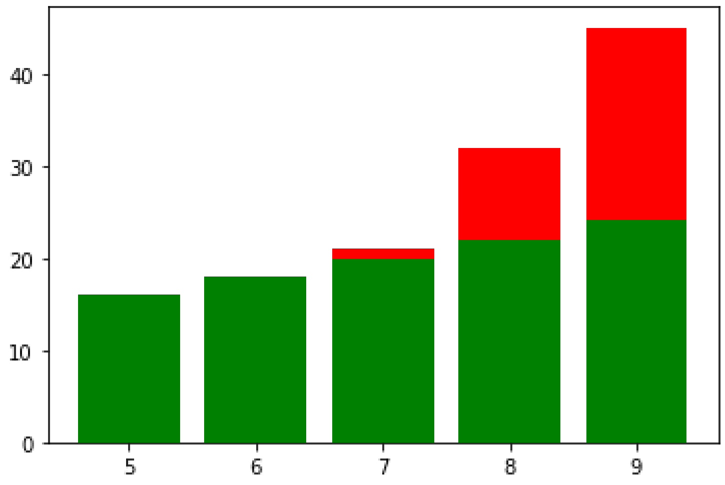


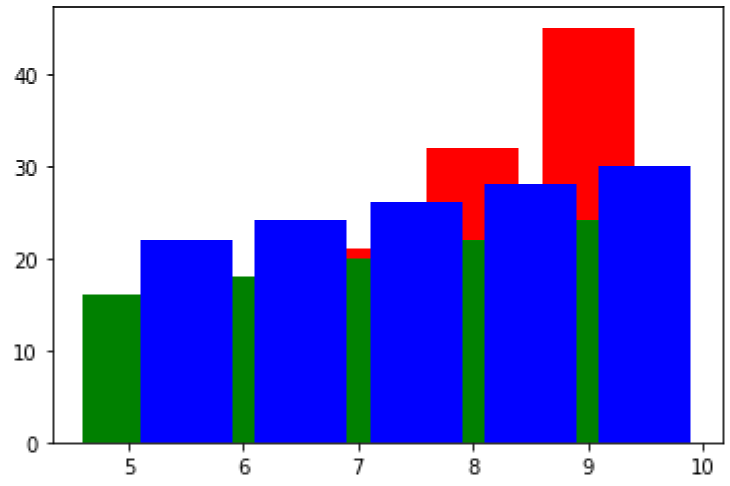
Bar Plots

```
In [65]: plt.bar(np.arange(5,10), np.arange(5,10)*np.arange(1,6), color='r')
         #plt.bar(np.arange(5,10), np.arange(8,13)*2, color='g')
         #plt.bar(np.arange(5.5,10.5), np.arange(11,16)*2, color='b')

Out[65]: <BarContainer object of 5 artists>
```



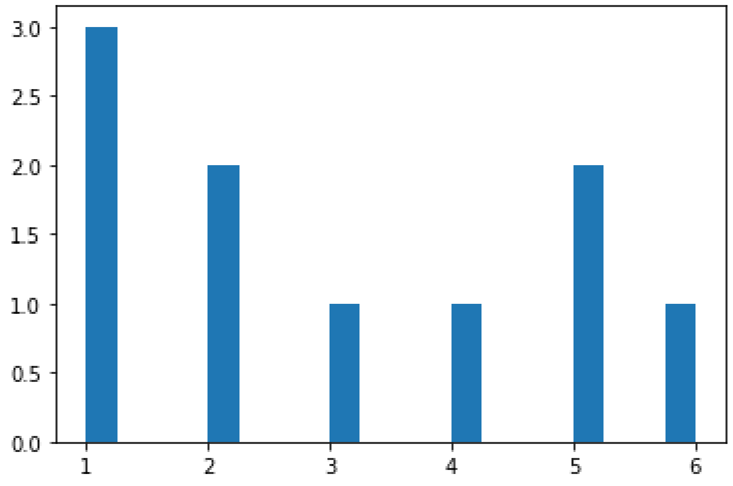
```
In [66]: plt.bar(np.arange(5,10), np.arange(5,10)*np.arange(1,6), color='r')
         plt.bar(np.arange(5,10), np.arange(8,13)*2, color='g')
         #plt.bar(np.arange(5.5,10.5), np.arange(11,16)*2, color='b')

Out[66]: <BarContainer object of 5 artists>
```



In [67]:
```
plt.bar(np.arange(5,10), np.arange(5,10)*np.arange(1,6), color='r')
plt.bar(np.arange(5,10), np.arange(8,13)*2, color='g')
plt.bar(np.arange(5.5,10.5), np.arange(11,16)*2, color='b')
```
Out[67]:
```
<BarContainer object of 5 artists>
```


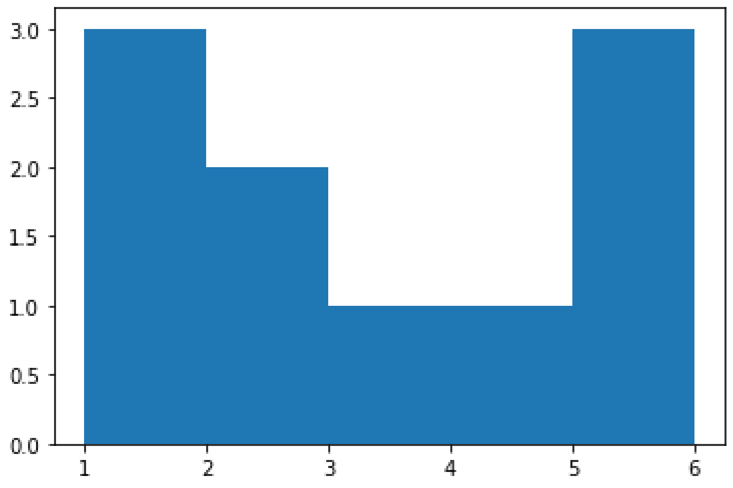
Histograms

HIstograms are plotted based on the frequency of our values in the dataset. On x-axis we have buckets to specify the window in which we are going to plot the frequency. on the y-axis we have the frequency of values falling into that bin.

In [70]:
```
array1= np.array([1,1,3,4,5,2,2,5,1,6])
plt.hist(array1, bins=20)
```
Out[70]:
```
(array([3., 0., 0., 0., 2., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0., 2.,
        0., 0., 1.]),
 array([1.  , 1.25, 1.5 , 1.75, 2.  , 2.25, 2.5 , 2.75, 3.  , 3.25, 3.5 ,
        3.75, 4.  , 4.25, 4.5 , 4.75, 5.  , 5.25, 5.5 , 5.75, 6.  ]),
 <a list of 20 Patch objects>)
```


In [71]:
```
array1= np.array([1,1,3,4,5,2,2,5,1,6])
plt.hist(array1, bins=5)
```

Out[71]:
```
(array([3., 2., 1., 1., 3.]),
 array([1., 2., 3., 4., 5., 6.]),
 <a list of 5 Patch objects>)
```



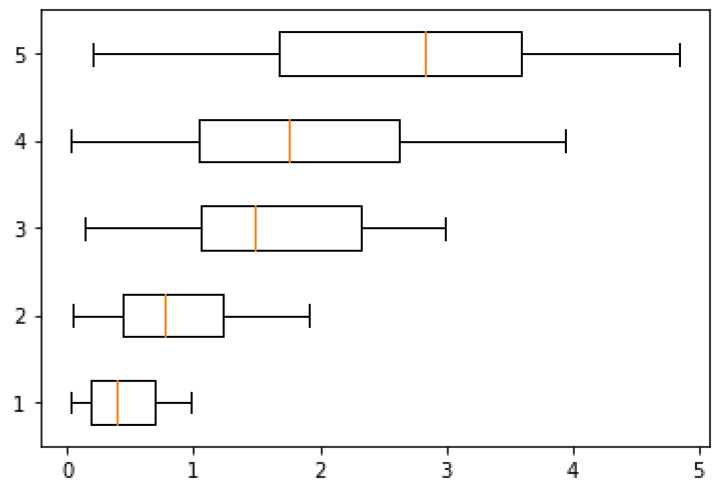
Box-Plots

```
In [75]: # Taking a random uniform distribution of 50 numeric values between 0 to standard deviat
         ion and
         # creating 5  such random variable sets to be plotted as box-plots
         x= [np.random.uniform(0,std, 50) for std in range(1,6)]
         x
```

```
Out[75]: [array([0.67530718, 0.13396924, 0.8193195 , 0.36171395, 0.75451011,
                 0.06724406, 0.32087612, 0.08396411, 0.44017797, 0.59954075,
                 0.29668746, 0.48359393, 0.08450414, 0.91492096, 0.05419235,
                 0.91631869, 0.96810025, 0.10153415, 0.32322887, 0.39231127,
                 0.19181091, 0.18361572, 0.37762395, 0.20241067, 0.68904362,
                 0.53428405, 0.07479945, 0.7188092 , 0.17883747, 0.98089024,
                 0.48700916, 0.85374457, 0.04655196, 0.22336433, 0.39665791,
                 0.98360176, 0.58400799, 0.87147659, 0.65725043, 0.31304598,
                 0.02137324, 0.59426811, 0.0631819 , 0.95224589, 0.50435698,
                 0.34227572, 0.73179601, 0.49407955, 0.04447487, 0.37853025]),
          array([0.15115182, 0.26369367, 0.64132745, 0.71398989, 0.61465344,
                 1.79830472, 0.20686686, 0.40956323, 1.31070456, 0.43606488,
                 1.2943931 , 0.71408346, 1.79844698, 1.14018007, 0.45905639,
                 1.87053275, 0.87603059, 1.19916778, 1.16497261, 0.51145731,
                 0.98716026, 1.2121237 , 0.97206781, 0.89451958, 0.04132753,
                 1.70584089, 1.3034242 , 1.58616246, 0.69708402, 0.31112572,
                 0.7171789 , 0.09260319, 0.43168363, 0.96875292, 0.77844781,
                 0.34446124, 1.91079919, 0.09145035, 1.76981224, 1.17478757,
                 0.41858528, 0.75854071, 0.63983851, 0.6016059 , 1.05974418,
                 1.23081354, 1.25926201, 0.44793447, 0.04962468, 1.89664628]),
          array([1.20898966, 1.41323422, 1.94799523, 0.68573925, 0.18437438,
                 2.06812746, 1.76164025, 1.14366724, 0.15996543, 0.9693056 ,
                 1.79940474, 2.70647704, 2.69175694, 2.53391224, 1.30541521,
                 0.81388361, 1.90622993, 2.72770292, 0.70841019, 0.12879204,
                 2.39590889, 0.50178231, 1.30894774, 2.09991315, 1.35901671,
                 1.84727811, 2.71904561, 0.8825165 , 0.75739965, 1.33553642,
                 2.04087626, 1.29656803, 2.42180281, 2.98792998, 1.04105399,
                 1.08200648, 2.52674125, 1.28900565, 1.44672589, 2.43214056,
                 2.43658557, 0.28447407, 1.95731376, 1.08992932, 2.92877545,
                 0.31384263, 2.81091827, 1.68434291, 1.52928441, 1.97282153]),
          array([3.46630302, 3.75111202, 2.63925287, 0.07956043, 3.8881961 ,
                 1.21597241, 2.38261161, 0.49584589, 1.7066383 , 1.79678917,
                 3.03732332, 1.31894156, 3.27640447, 1.92035698, 1.36096509,
                 1.15720037, 1.88991507, 0.15025854, 2.22959073, 0.89334748,
                 3.93698192, 2.2102151 , 2.25384903, 0.29667444, 0.16619087,
                 2.45204712, 2.95600354, 3.17384913, 2.06851568, 0.09779989,
                 3.04670499, 1.03012189, 0.81578727, 2.59111867, 1.65179016,
                 1.69799975, 3.15519157, 1.44228405, 0.49151543, 2.06326062,
                 1.32247178, 0.02990853, 3.28230194, 1.05344938, 0.32267716,
                 0.66155209, 3.59055653, 1.51935297, 2.47075798, 1.52742047]),
          array([3.98061964, 3.49312446, 1.09840712, 3.11911547, 1.29145661,
                 2.61829172, 0.99610218, 4.82576544, 1.83279478, 1.88048053,
                 2.82247225, 2.83888067, 2.36491387, 3.01324867, 3.63224814,
                 1.38160679, 3.41587349, 4.27605469, 3.53434911, 3.42903424,
                 3.25625917, 1.89067102, 3.60428726, 3.51887458, 1.67083075,
                 1.00122219, 3.61770276, 3.03232195, 2.56777017, 4.72509342,
                 2.82877223, 4.85017462, 0.38043225, 3.42360067, 1.54415468,
                 2.67554725, 2.73883976, 3.75775687, 1.86643631, 1.00752592,
                 0.78413941, 4.83454249, 0.19363999, 1.67801254, 2.73439705,
                 0.39630237, 3.86435572, 4.54681549, 4.59507102, 1.27341991])]
```

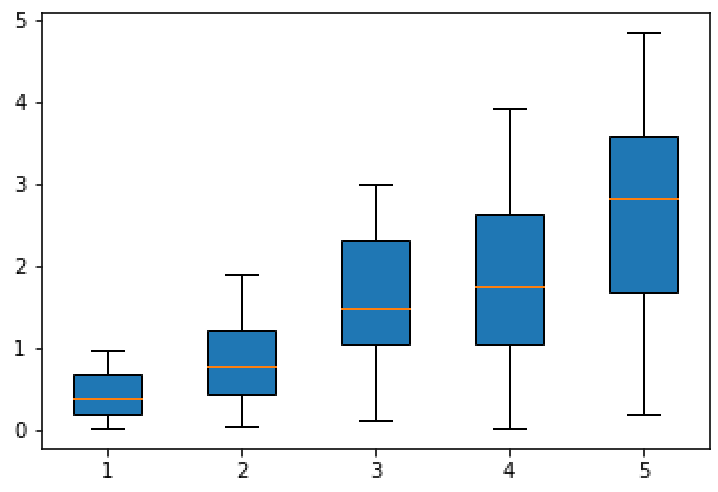
```
In [78]: plt.boxplot(x, vert=False)
```

```
Out[78]: {'whiskers': [<matplotlib.lines.Line2D at 0x735eeae888>,
           <matplotlib.lines.Line2D at 0x735eeaef08>,
           <matplotlib.lines.Line2D at 0x735eeb8cc8>,
           <matplotlib.lines.Line2D at 0x735eebec08>,
           <matplotlib.lines.Line2D at 0x735eecd648>,
           <matplotlib.lines.Line2D at 0x735eed29c8>,
           <matplotlib.lines.Line2D at 0x735ece1788>,
           <matplotlib.lines.Line2D at 0x735eed78c8>,
           <matplotlib.lines.Line2D at 0x735eef0848>,
           <matplotlib.lines.Line2D at 0x735eed2708>],
          'caps': [<matplotlib.lines.Line2D at 0x735eeaedc8>,
           <matplotlib.lines.Line2D at 0x735eeb3fc8>,
           <matplotlib.lines.Line2D at 0x735eec3fc8>,
           <matplotlib.lines.Line2D at 0x735eec3cc8>,
           <matplotlib.lines.Line2D at 0x735eed2748>,
           <matplotlib.lines.Line2D at 0x735eed7b08>,
           <matplotlib.lines.Line2D at 0x735eedcbc8>,
           <matplotlib.lines.Line2D at 0x735eee5448>,
           <matplotlib.lines.Line2D at 0x735eeecb08>,
           <matplotlib.lines.Line2D at 0x735eed7888>],
          'boxes': [<matplotlib.lines.Line2D at 0x735eea8f08>,
           <matplotlib.lines.Line2D at 0x735ee43148>,
           <matplotlib.lines.Line2D at 0x735eecdf48>,
           <matplotlib.lines.Line2D at 0x735eedcac8>,
           <matplotlib.lines.Line2D at 0x735eee05c8>],
          'medians': [<matplotlib.lines.Line2D at 0x735eeb3d88>,
           <matplotlib.lines.Line2D at 0x735eec7f08>,
           <matplotlib.lines.Line2D at 0x735eedcc88>,
           <matplotlib.lines.Line2D at 0x735eee56c8>,
           <matplotlib.lines.Line2D at 0x735eef6c88>],
          'fliers': [<matplotlib.lines.Line2D at 0x735eeb8f08>,
           <matplotlib.lines.Line2D at 0x735eec7dc8>,
           <matplotlib.lines.Line2D at 0x735eedc9c8>,
           <matplotlib.lines.Line2D at 0x735eeec388>,
           <matplotlib.lines.Line2D at 0x735ef02ec8>],
          'means': []}
```



In [79]:
```
# making the plots stand vertical with color patching

plt.boxplot(x, vert=True, patch_artist=True)
```

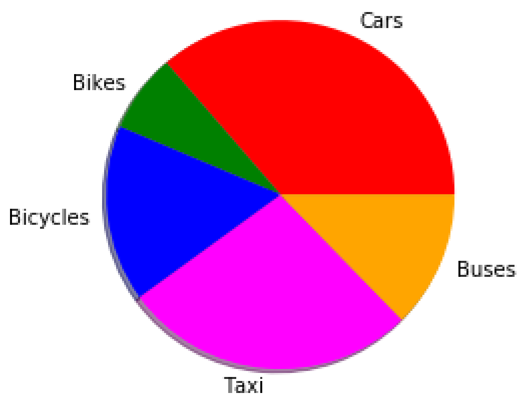
Out[79]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x735ef6d788>,
  <matplotlib.lines.Line2D at 0x735ef6df08>,
  <matplotlib.lines.Line2D at 0x735ef7ec88>,
  <matplotlib.lines.Line2D at 0x735ef78d08>,
  <matplotlib.lines.Line2D at 0x735ef93148>,
  <matplotlib.lines.Line2D at 0x735ef8db08>,
  <matplotlib.lines.Line2D at 0x735efa3a48>,
  <matplotlib.lines.Line2D at 0x735ef97b48>,
  <matplotlib.lines.Line2D at 0x735efb3e08>,
  <matplotlib.lines.Line2D at 0x735ef8d1c8>],
 'caps': [<matplotlib.lines.Line2D at 0x735ef6dd48>,
  <matplotlib.lines.Line2D at 0x735ef73fc8>,
  <matplotlib.lines.Line2D at 0x735ef83d08>,
  <matplotlib.lines.Line2D at 0x735ef83c88>,
  <matplotlib.lines.Line2D at 0x735ef93748>,
  <matplotlib.lines.Line2D at 0x735ef97bc8>,
  <matplotlib.lines.Line2D at 0x735efa82c8>,
  <matplotlib.lines.Line2D at 0x735efa80c8>,
  <matplotlib.lines.Line2D at 0x735efb9688>,
  <matplotlib.lines.Line2D at 0x735ef97a88>],
 'boxes': [<matplotlib.patches.PathPatch at 0x735ef6d148>,
  <matplotlib.patches.PathPatch at 0x735ef78f48>,
  <matplotlib.patches.PathPatch at 0x735ef8da08>,
  <matplotlib.patches.PathPatch at 0x735ef9bb88>,
  <matplotlib.patches.PathPatch at 0x735efb3188>],
 'medians': [<matplotlib.lines.Line2D at 0x735ef73d88>,
  <matplotlib.lines.Line2D at 0x735ef89c48>,
  <matplotlib.lines.Line2D at 0x735ef97908>,
  <matplotlib.lines.Line2D at 0x735efa8848>,
  <matplotlib.lines.Line2D at 0x735efae208>],
 'fliers': [<matplotlib.lines.Line2D at 0x735ef78fc8>,
  <matplotlib.lines.Line2D at 0x735ef89a08>,
  <matplotlib.lines.Line2D at 0x735ef9bd48>,
  <matplotlib.lines.Line2D at 0x735efa8cc8>,
  <matplotlib.lines.Line2D at 0x735efbed08>],
 'means': []}
```



Pie-Charts in Matplotlib

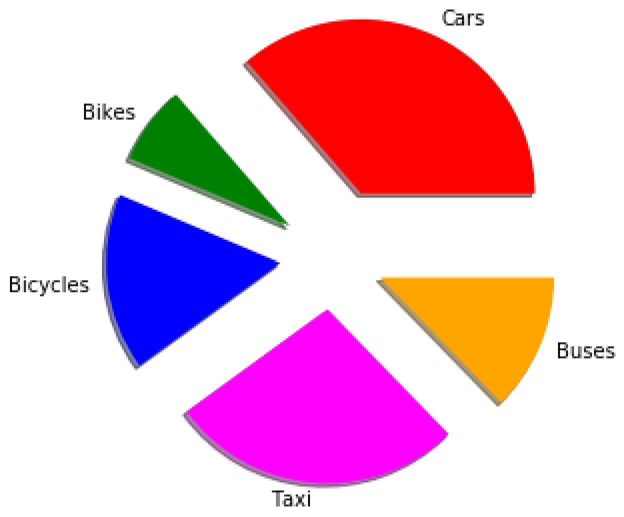
In [94]:
```
sizes=[100, 20, 45, 75, 35]
labels1='Cars', 'Bikes', 'Bicycles', 'Taxi', 'Buses'
colors1=['red', 'green', 'blue', 'magenta', 'orange']
explode1=(0.4,0.3, 0.3, 0.3, 0.3)

plt.pie(sizes,  labels=labels1, colors=colors1,  shadow=True)
```

Out[94]:
```
([<matplotlib.patches.Wedge at 0x735f253108>,
  <matplotlib.patches.Wedge at 0x735f253848>,
  <matplotlib.patches.Wedge at 0x735f25c148>,
  <matplotlib.patches.Wedge at 0x735f25c688>,
  <matplotlib.patches.Wedge at 0x735f265548>],
 [Text(0.4569564802357173, 1.000595210447554, 'Cars'),
  Text(-0.8899187331606259, 0.6465637233635886, 'Bikes'),
  Text(-1.0928299049345258, -0.1253905852956893, 'Bicycles'),
  Text(-0.09413247108056755, -1.0959649072339253, 'Taxi'),
  Text(1.0132360413873753, -0.42819706261678314, 'Buses')])
```



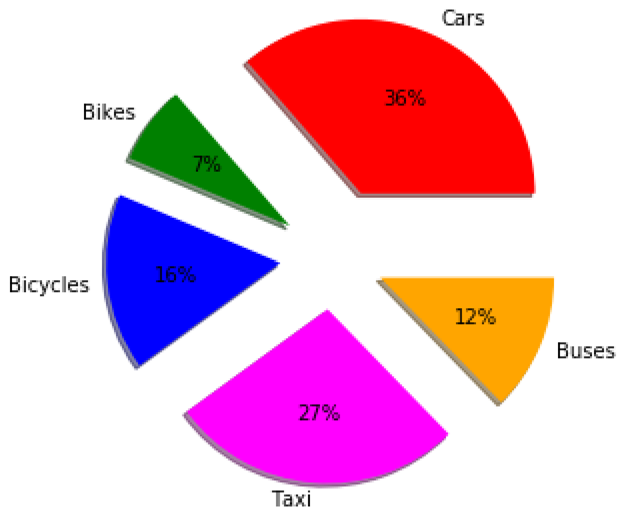
```
In [95]: sizes=[100, 20, 45, 75, 35]
         labels1='Cars', 'Bikes', 'Bicycles', 'Taxi', 'Buses'
         colors1=['red', 'green', 'blue', 'magenta', 'orange']
         explode1=(0.4,0.3, 0.3, 0.3, 0.3)

         plt.pie(sizes,  labels=labels1, colors=colors1,  shadow=True, explode=explode1)
```

```
Out[95]: ([<matplotlib.patches.Wedge at 0x735f2a3b88>,
           <matplotlib.patches.Wedge at 0x735f2af608>,
           <matplotlib.patches.Wedge at 0x735f2b44c8>,
           <matplotlib.patches.Wedge at 0x735f2ba448>,
           <matplotlib.patches.Wedge at 0x735f2c04c8>],
          [Text(0.6231224730487054, 1.3644480142466644, 'Cars'),
           Text(-1.1326238422044328, 0.822899284280931, 'Bikes'),
           Text(-1.3908744244621238, -0.1595880176490591, 'Bicycles'),
           Text(-0.1198049631934496, -1.3948644273886321, 'Taxi'),
           Text(1.2895731435839322, -0.5449780796940876, 'Buses')])
```



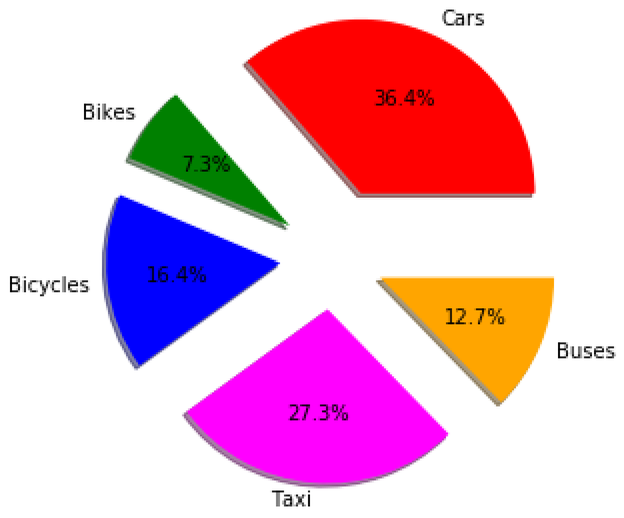
```
In [109]: #if you wish to see the percentage share as well we use autopct as:

          sizes=[100, 20, 45, 75, 35]
          labels1='Cars', 'Bikes', 'Bicycles', 'Taxi', 'Buses'
          colors1=['red', 'green', 'blue', 'magenta', 'orange']
          explode1=(0.4,0.3, 0.3, 0.3, 0.3)

          plt.pie(sizes,  labels=labels1, colors=colors1,  shadow=True, explode=explode1, autopct=
          '%1i%%') #integer

          #plt.pie(sizes,  labels=labels1, colors=colors1,  shadow=True, explode=explode1, autopct
          = '%1.1f%%') #float
```

```
Out[109]: ([<matplotlib.patches.Wedge at 0x735f0e20c8>,
            <matplotlib.patches.Wedge at 0x735f0e2a08>,
            <matplotlib.patches.Wedge at 0x735f0dd648>,
            <matplotlib.patches.Wedge at 0x735f003848>,
            <matplotlib.patches.Wedge at 0x735efff108>],
           [Text(0.6231224730487054, 1.3644480142466644, 'Cars'),
            Text(-1.1326238422044328, 0.822899284280931, 'Bikes'),
            Text(-1.3908744244621238, -0.1595880176490591, 'Bicycles'),
            Text(-0.1198049631934496, -1.3948644273886321, 'Taxi'),
            Text(1.2895731435839322, -0.5449780796940876, 'Buses')],
           [Text(0.41541498203247024, 0.9096320094977761, '36%'),
            Text(-0.728115327131421, 0.529006682752027, '7%'),
            Text(-0.8941335585827938, -0.10259229706010942, '16%'),
            Text(-0.07701747633864617, -0.8966985604641204, '27%'),
            Text(0.8290113065896707, -0.35034305123191345, '12%')])
```



In [110]:
```
#if you wish to see the percentage share as well we use autopct as:

sizes=[100, 20, 45, 75, 35]
labels1='Cars', 'Bikes', 'Bicycles', 'Taxi', 'Buses'
colors1=['red', 'green', 'blue', 'magenta', 'orange']
explode1=(0.4,0.3, 0.3, 0.3, 0.3)

#plt.pie(sizes,  labels=labels1, colors=colors1,  shadow=True, explode=explode1, autopct
= '%1i%%') #integer

plt.pie(sizes,  labels=labels1, colors=colors1,  shadow=True, explode=explode1, autopct=
'%1.1f%%') #float
```

Out[110]:
```
([<matplotlib.patches.Wedge at 0x73603b7148>,
  <matplotlib.patches.Wedge at 0x73603bd048>,
  <matplotlib.patches.Wedge at 0x73603c2408>,
  <matplotlib.patches.Wedge at 0x73603c88c8>,
  <matplotlib.patches.Wedge at 0x73603cee88>],
 [Text(0.6231224730487054, 1.3644480142466644, 'Cars'),
  Text(-1.1326238422044328, 0.822899284280931, 'Bikes'),
  Text(-1.3908744244621238, -0.1595880176490591, 'Bicycles'),
  Text(-0.1198049631934496, -1.3948644273886321, 'Taxi'),
  Text(1.2895731435839322, -0.5449780796940876, 'Buses')],
 [Text(0.41541498203247024, 0.9096320094977761, '36.4%'),
  Text(-0.728115327131421, 0.529006682752027, '7.3%'),
  Text(-0.8941335585827938, -0.10259229706010942, '16.4%'),
  Text(-0.07701747633864617, -0.8966985604641204, '27.3%'),
  Text(0.8290113065896707, -0.35034305123191345, '12.7%')])
```



/