In [1]:

```
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:

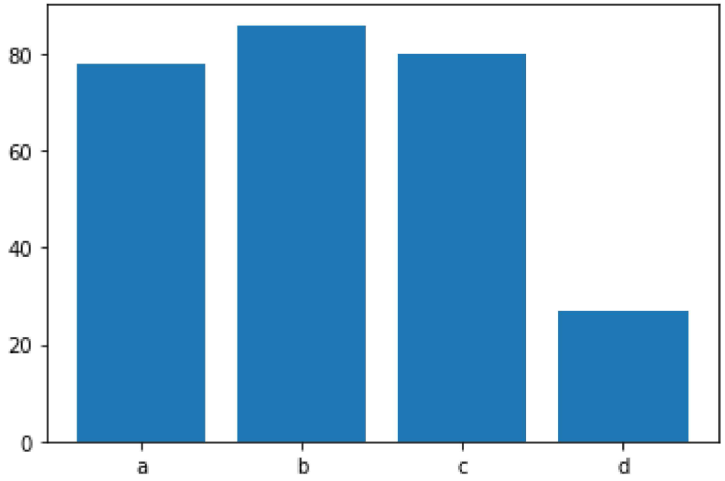
```
#bar chart
name=['a','b','c','d']
marks=[78,86,80,27]
```

In [3]:

```
plt.bar(name,marks)
```

Out[3]:

```
<BarContainer object of 4 artists>
```

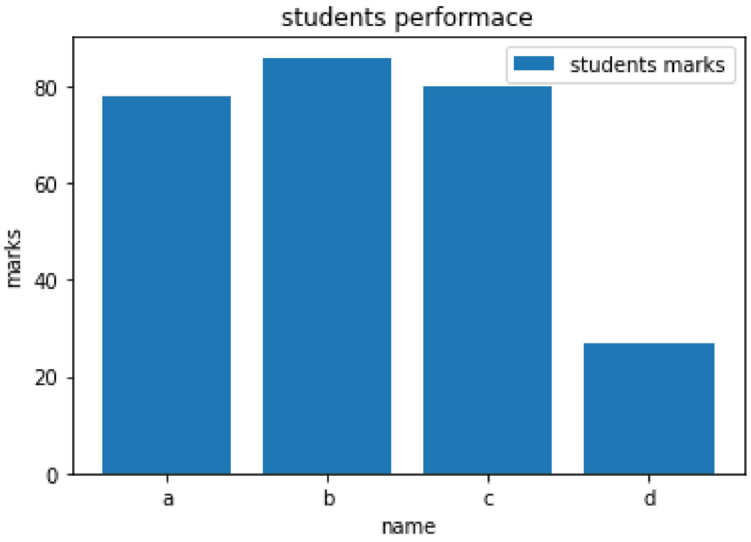


In [4]:

```
plt.bar(name,marks,label="students marks")
plt.xlabel("name")
plt.ylabel("marks")
plt.title('students performace')
plt.legend()
```

Out[4]:

```
<matplotlib.legend.Legend at 0x17c74c0cfa0>
```

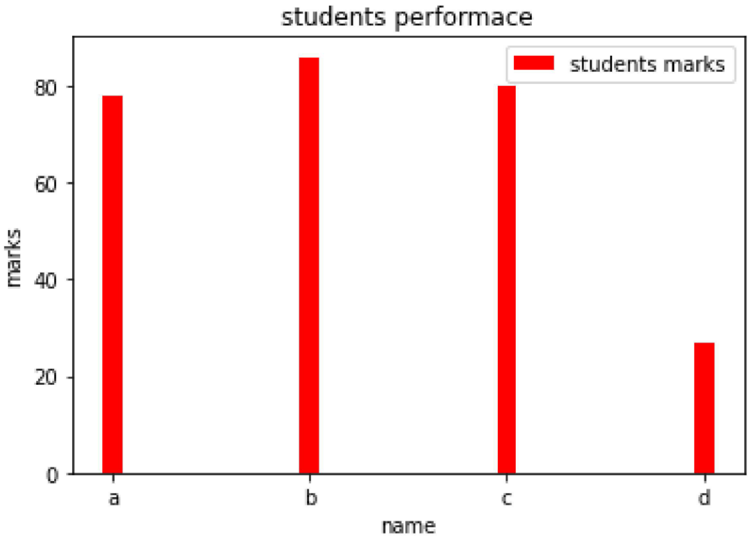


In [6]:

```
plt.bar(name,marks,label="students marks",width=0.1,color='r')
plt.xlabel("name")
plt.ylabel("marks")
plt.title('students performace')
plt.legend()
```

Out[6]:

```
<matplotlib.legend.Legend at 0x17c74d29fd0>
```

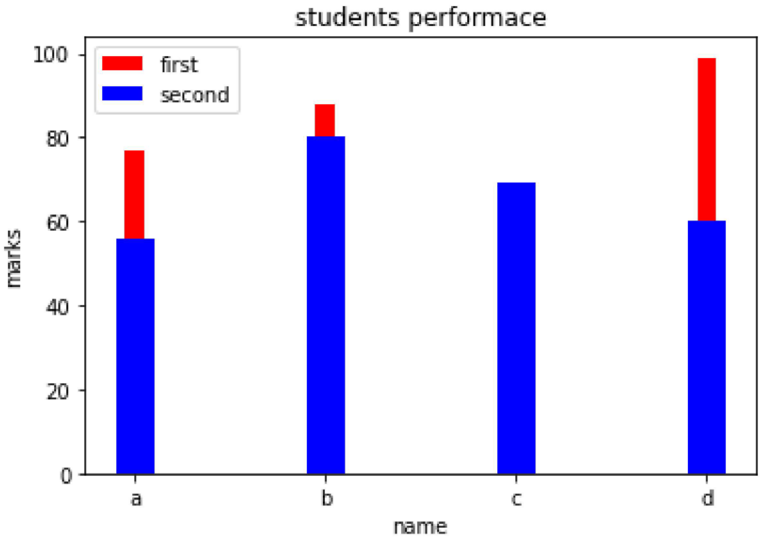


In [10]:

```
f=[77,88,36,99]
s=[56,80,69,60]
plt.bar(name,f,label="first",width=0.1,color='r')
plt.bar(name,s,label="second",width=0.2,color='b')
plt.xlabel("name")
plt.ylabel("marks")
plt.title('students performace')
plt.legend()
```

Out[10]:

```
<matplotlib.legend.Legend at 0x17c74ebfb80>
```

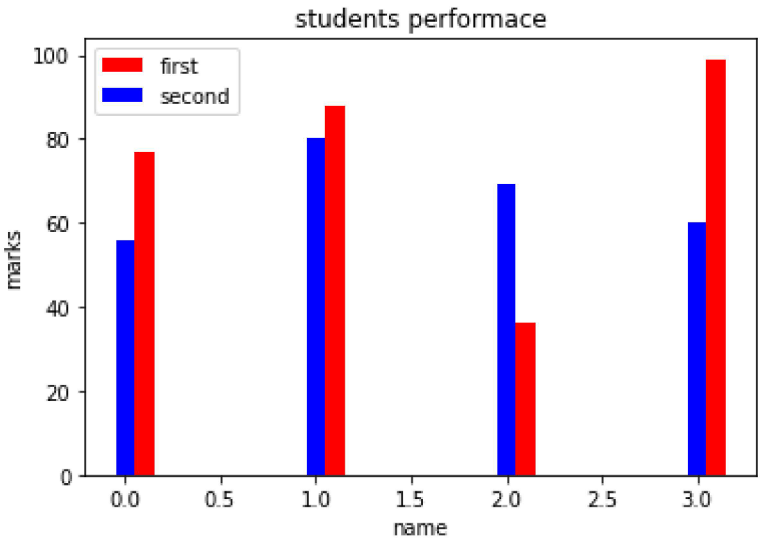


In [12]:

```
#print bar side by side

f=[77,88,36,99]
s=[56,80,69,60]
xpos = np.arange(len(name))
plt.bar(xpos+0.1,f,label="first",width=0.1,color='r')
plt.bar(xpos,s,label="second",width=0.1,color='b')
plt.xlabel("name")
plt.ylabel("marks")
plt.title('students performace')
plt.legend()
```

Out[12]:

```
<matplotlib.legend.Legend at 0x17c74f912e0>
```



In [13]:

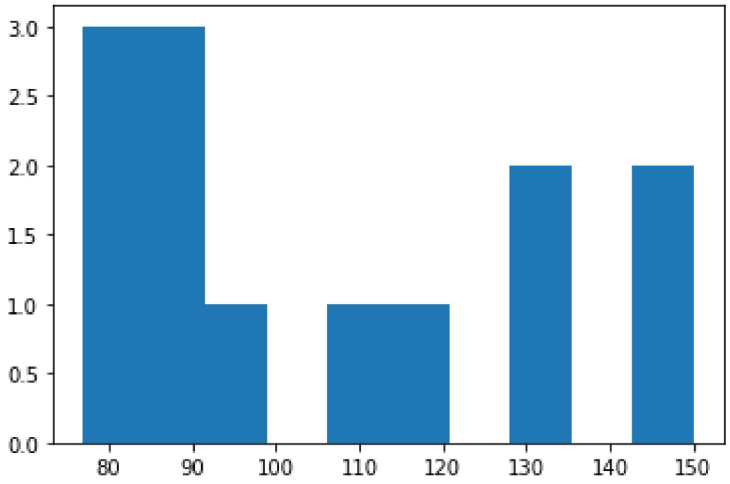
```
xpos
```

Out[13]:

```
array([0, 1, 2, 3])
```

In [19]:
```
#histogram is an accurate representation of the distribution of numerical data
#students test performace
sp = [113, 85, 90, 150, 149, 88, 93, 115, 135, 80, 77, 82, 129]
plt.hist(sp) # by default number of bins is set to 10
```

Out[19]:
```
(array([3., 3., 1., 0., 1., 1., 0., 2., 0., 2.]),
 array([ 77. ,  84.3,  91.6,  98.9, 106.2, 113.5, 120.8, 128.1, 135.4,
        142.7, 150. ]),
 <a list of 10 Patch objects>)
```

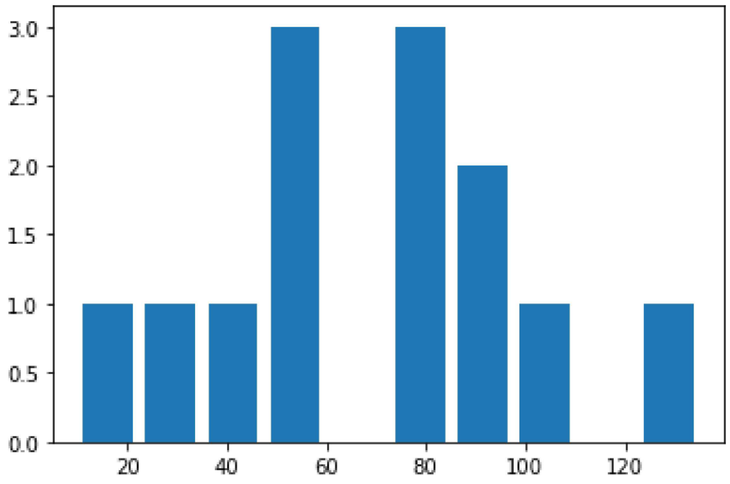


In [25]:

```
#histogram is an accurate representation of the distribution of numerical data
#students test performace
sp = [100, 85, 10, 50, 49, 50, 93, 35, 135, 80, 77, 82, 29]
plt.hist(sp, rwidth=0.8) # by default number of bins is set to 10

```

Out[25]:

```
(array([1., 1., 1., 3., 0., 3., 2., 1., 0., 1.]),
 array([ 10. ,  22.5,  35. ,  47.5,  60. ,  72.5,  85. ,  97.5, 110. ,
        122.5, 135. ]),
 <a list of 10 Patch objects>)
```



In [26]:

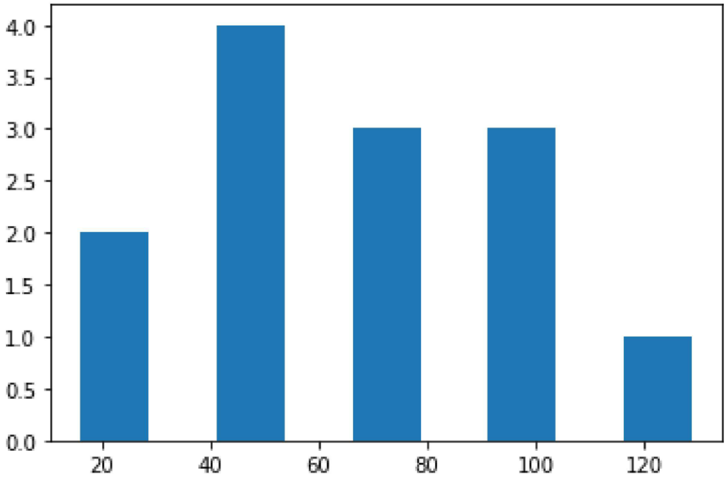
```
#To construct a histogram, follow these steps -
    #Bin the range of values.
    #Divide the entire range of values into a series of intervals.
    #Count how many values fall into each interval.
```

In [27]:

```
plt.hist(sp,rwidth=0.5,bins=5)
```

Out[27]:

```
(array([2., 4., 3., 3., 1.]),
 array([ 10.,  35.,  60.,  85., 110., 135.]),
 <a list of 5 Patch objects>)
```

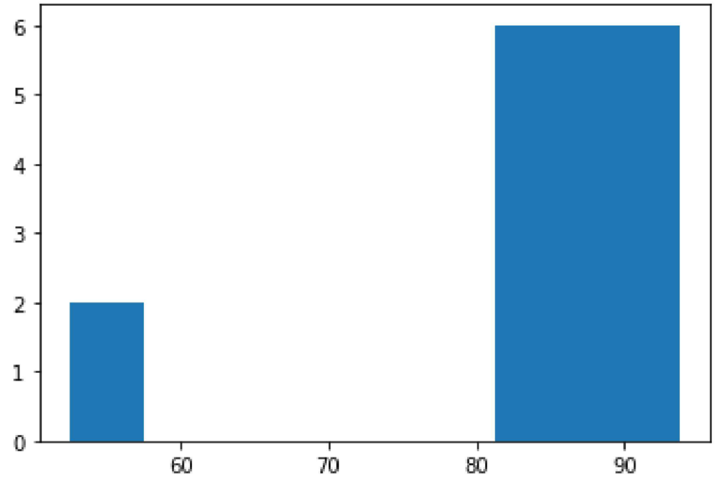


In [28]:

```
plt.hist(sp,rwidth=0.5,bins=[50,60,75,100])
```

Out[28]:

```
(array([2., 0., 6.]), array([ 50,  60,  75, 100]), <a list of 3 Patch objects>)
```



In [29]:

```
sp
```

Out[29]:

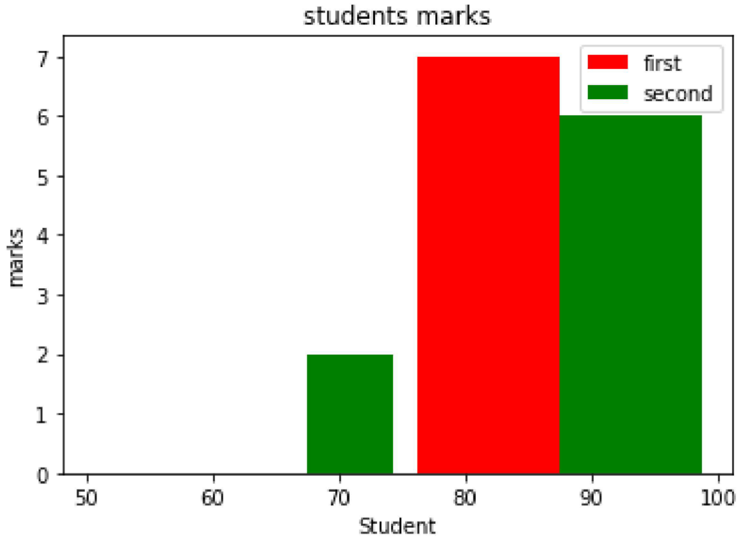
```
[100, 85, 10, 50, 49, 50, 93, 35, 135, 80, 77, 82, 29]
```

In [30]:

```
plt.xlabel("Student")
plt.ylabel("marks")
plt.title("students marks")

sp1 = [113, 85, 90, 150, 149, 88, 93, 115, 135, 80, 77, 82, 129]
sp2 = [67, 98, 89, 120, 133, 150, 84, 69, 89, 79, 120, 112, 100]

plt.hist([sp1,sp2], bins=[50,60,75,100], rwidth=0.9, color=['r','g'],label=['first','second'])
plt.legend()
```

Out[30]:

```
<matplotlib.legend.Legend at 0x17c7639d8b0>
```

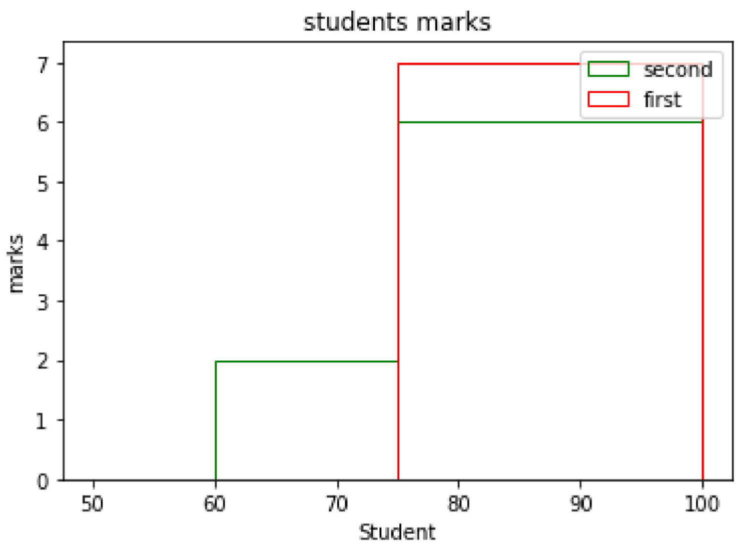


In [31]:

```
plt.xlabel("Student")
plt.ylabel("marks")
plt.title("students marks")

sp1 = [113, 85, 90, 150, 149, 88, 93, 115, 135, 80, 77, 82, 129]
sp2 = [67, 98, 89, 120, 133, 150, 84, 69, 89, 79, 120, 112, 100]

plt.hist([sp1,sp2], bins=[50,60,75,100], rwidth=0.9, color=['r','g'],label=['first','second'],histtype='ste
plt.legend()


```

Out[31]:

```
<matplotlib.legend.Legend at 0x17c76404550>
```

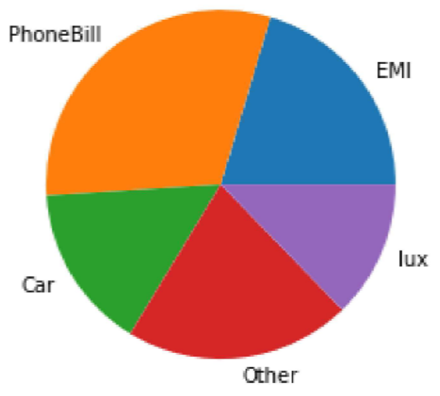


In [33]:

```
exp_vals = [400,600,300,410,250]
exp_labels = ["EMI","PhoneBill","Car","Other","lux"]
plt.pie(exp_vals,labels=exp_labels)
```

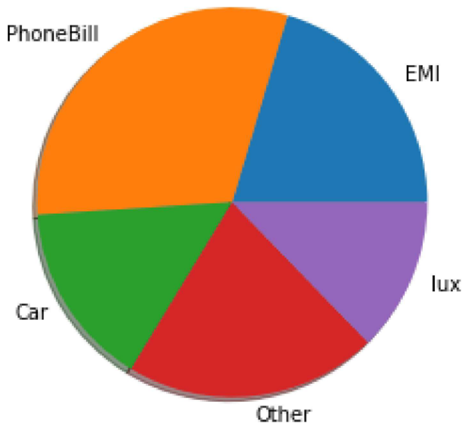
Out[33]:

```
([<matplotlib.patches.Wedge at 0x17c765ceca0>,
  <matplotlib.patches.Wedge at 0x17c765db190>,
  <matplotlib.patches.Wedge at 0x17c765db610>,
  <matplotlib.patches.Wedge at 0x17c765dba90>,
  <matplotlib.patches.Wedge at 0x17c765dbf10>],
 [Text(0.881554969597829, 0.6579216029112976, 'EMI'),
  Text(-0.6858388280562522, 0.8600145940217683, 'PhoneBill'),
  Text(-0.9406570006261246, -0.5702318889478766, 'Car'),
  Text(0.12316101874536206, -1.093083420175059, 'Other'),
  Text(1.0128613072960144, -0.4290827101883842, 'lux')])
```



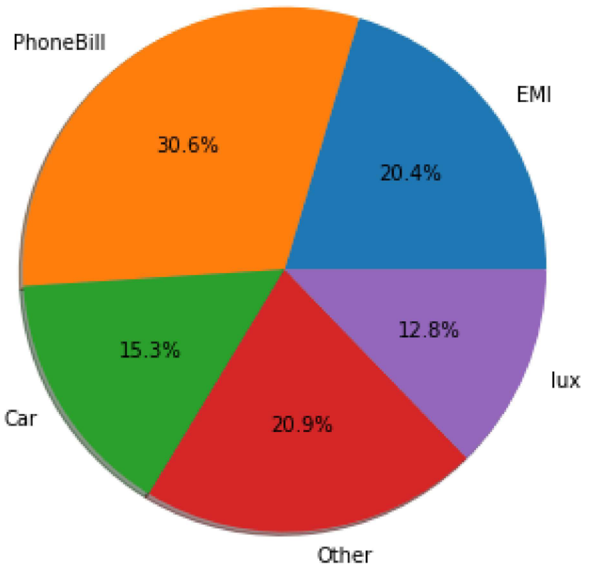
In [34]:

```
#draw perfect circle
plt.pie(exp_vals,labels=exp_labels, shadow=True)
plt.axis("equal")
plt.show()
```



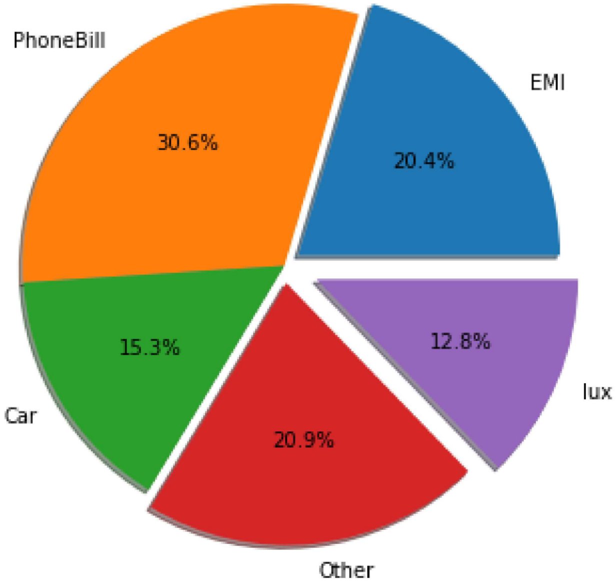
In [35]:

```
#Show percentages for every pie. Specify radius to increase chart size
plt.axis("equal")
plt.pie(exp_vals,labels=exp_labels, shadow=True, autopct='%1.1f%%',radius=1.5)
plt.show()


```



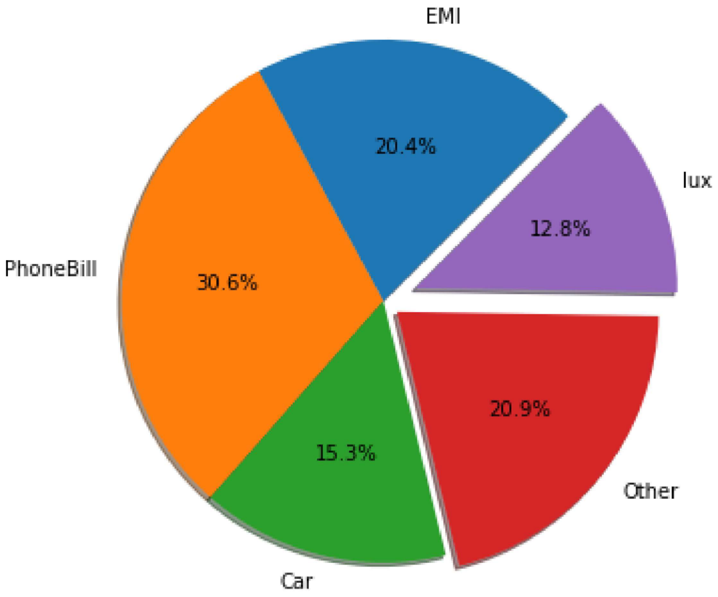
In [37]:

```
#Explode
plt.axis("equal")
plt.pie(exp_vals,labels=exp_labels, shadow=True, autopct='%1.1f%%',radius=1.5,explode=[0.1,0,0,0.1,0.2])
plt.show()
```



In [38]:

```
#counterclock and angle properties
plt.axis("equal")
plt.pie(exp_vals,labels=exp_labels, shadow=True, autopct='%1.1f%%',radius=1.5,explode=[0,0,0,0.1,0.2],counte
plt.show()
```



In [ ]: