Day-29

Topic: Matplotlib

#30daysofpython

Day-29 of #30daysofpython

Topic: matplotlib

- Creating figure & axis
- title, labelling x & y axis
- Axes labels, legend, grid
- bar plot
  - Plotting '2' different values in same plot
  - To get graphs side by side in single plot
  - Horizontal bar plot
- Histograms
  - orientation, plots histogram horizontally
- piecharts
  - To show '%' in pie charts
  - explode
  - Rotating piecharts
- Saving plots

Importing libraries

In [1]:

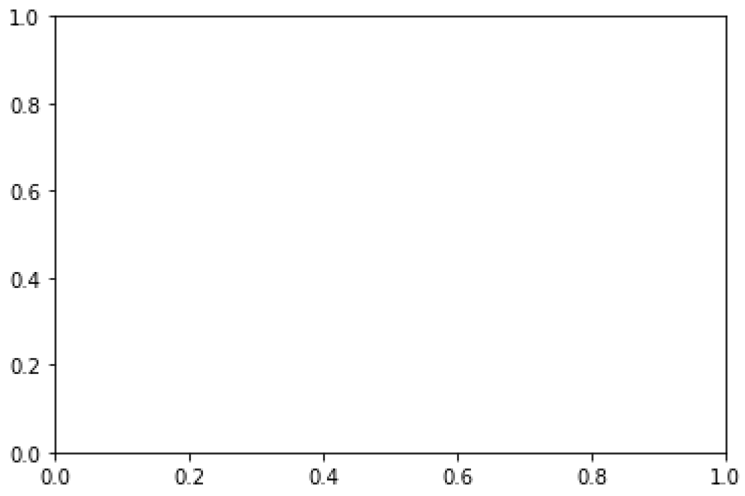
```
import matplotlib.pyplot as plt   # Importing matplotlib library
# plots the graph next to the cell
%matplotlib inline

import pandas as pd
import numpy as np
```

Creating figure + axis

In [2]:

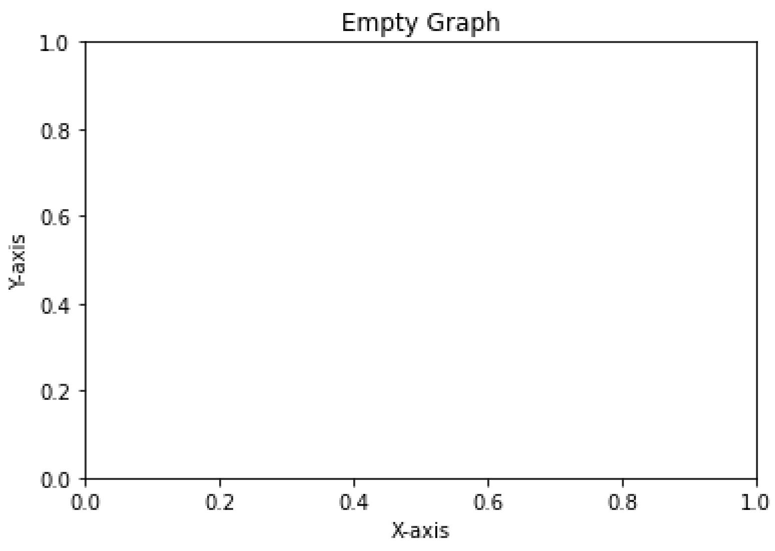
```
fig,ax = plt.subplots()
plt.show()   # Gives a empty graph
```



Title, Labelling x- axis & y- axis

In [3]:

```
plt.title("Empty Graph")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.show()
```



Example

In [4]:

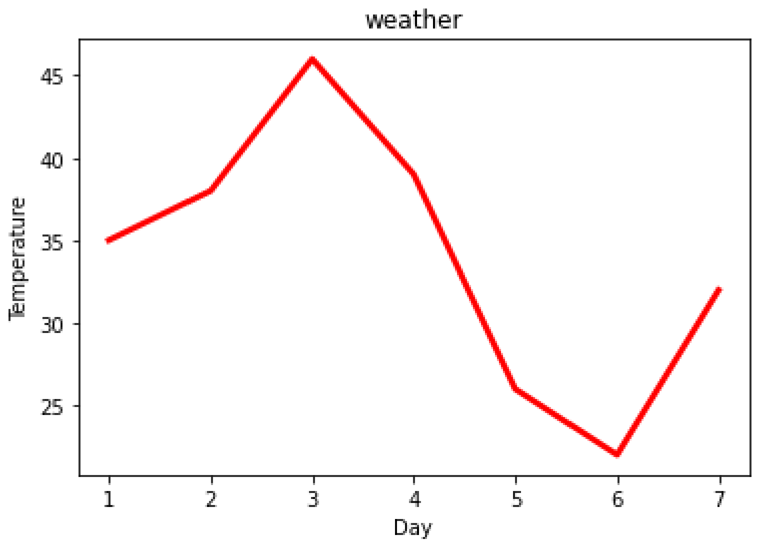
```
x = [1,2,3,4,5,6,7]          # x is number of days in a week
y = [35,38,46,39,26,22,32]    # y is temperatures corresponding to day
```

In [9]:

```
plt.title("weather")
plt.xlabel("Day")
plt.ylabel("Temperature")
plt.plot(x,y,color = "r",linewidth = 3)
```

Out[9]:

```
[<matplotlib.lines.Line2D at 0x1f83b45bdf0>]
```



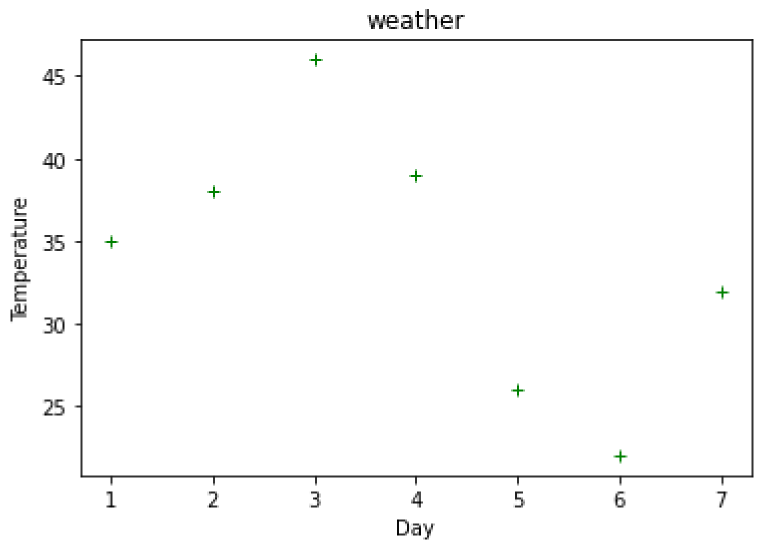
Same above example with different style & colour

In [11]:

```
x = [1,2,3,4,5,6,7]              # x is number of days in a week
y = [35,38,46,39,26,22,32]     # y is temperatures corresponding to day
plt.title("weather")
plt.xlabel("Day")
plt.ylabel("Temperature")
plt.plot(x,y,"g+",linewidth = 3)
```

Out[11]:

```
[<matplotlib.lines.Line2D at 0x1f83b892310>]
```

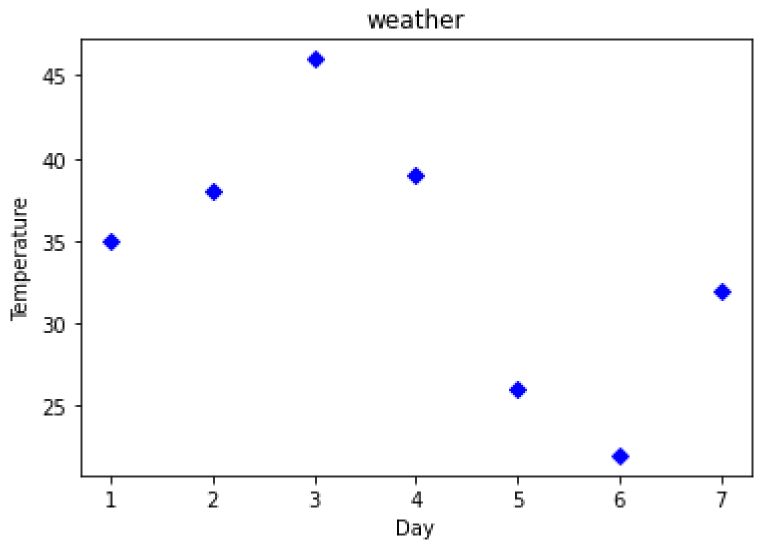


In [14]:

```
x = [1,2,3,4,5,6,7]              # x is number of days in a week
y = [35,38,46,39,26,22,32]     # y is temperatures corresponding to day
plt.title("weather")
plt.xlabel("Day")
plt.ylabel("Temperature")
plt.plot(x,y,"bD",linewidth = 3)
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x1f83b98ec10>]
```

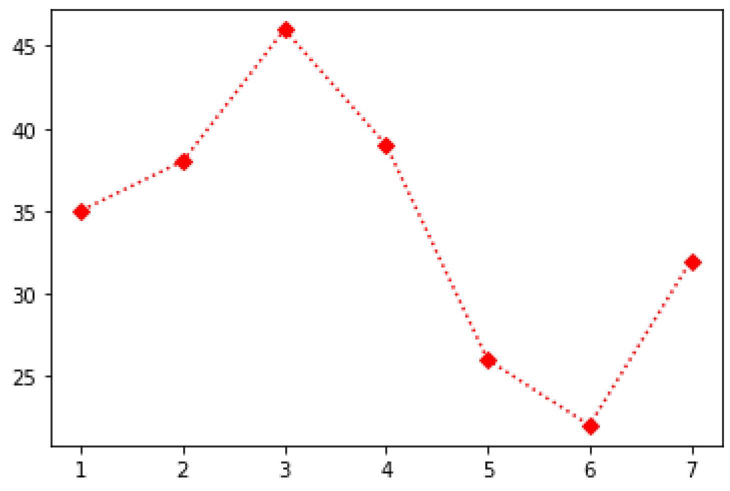


In [15]:

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

Out[15]:

```
[<matplotlib.lines.Line2D at 0x1f83b9e5d90>]
```



Axes Labels, Legend, Grid

Example

In [16]:

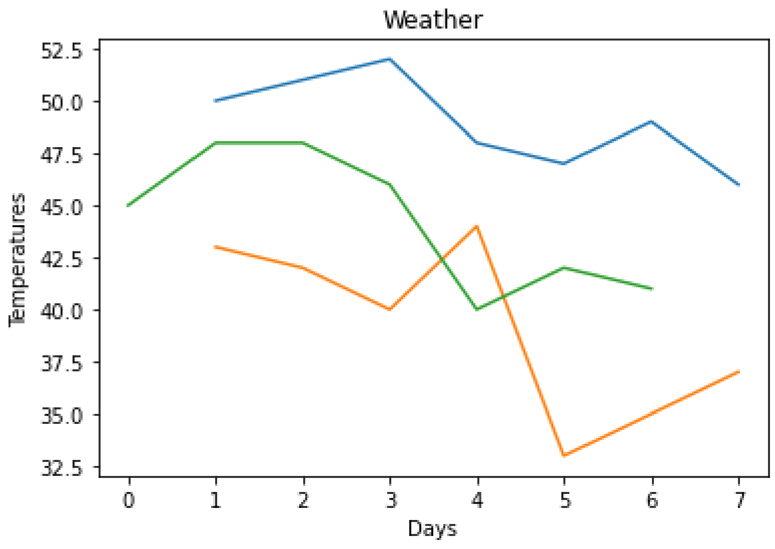
```
days=[1,2,3,4,5,6,7]          # Days in a week
max_t=[50,51,52,48,47,49,46]  # Maximum Temperature
min_t=[43,42,40,44,33,35,37]  # Minimum Temperature
avg_t=[45,48,48,46,40,42,41]  # Average Temperature
```

In [17]:

```
### Plotting Days on x-axis & other 3 lists on y-axis
plt.xlabel("Days")
plt.ylabel("Temperatures")
plt.title("Weather")
plt.plot(days,max_t)
plt.plot(days,min_t)
plt.plot(avg_t)
```

Out[17]:

```
[<matplotlib.lines.Line2D at 0x1f83d879640>]
```



- From above graph, we don't know which is max_t, min_t so, we need legend

legend()

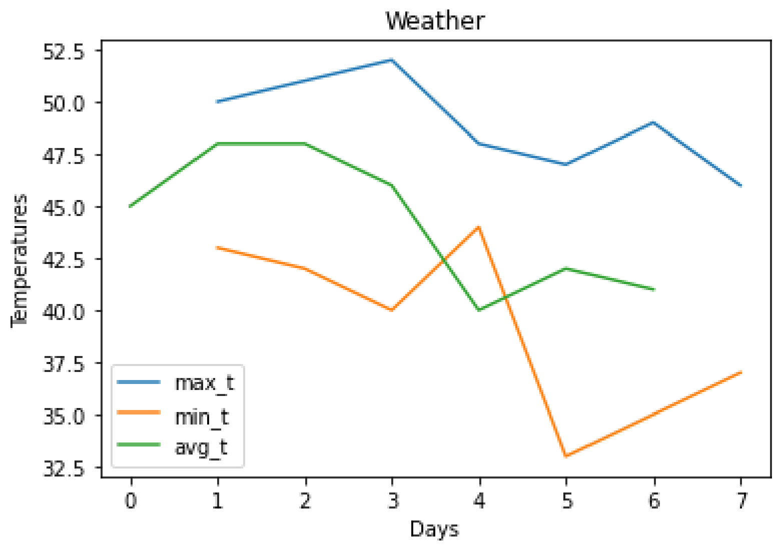
In [18]:

```
plt.xlabel("Days")
plt.ylabel("Temperatures")
plt.title("Weather")
plt.plot(days,max_t ,label="max_t")
plt.plot(days,min_t ,label="min_t")
plt.plot(avg_t ,label = "avg_t")

plt.legend()
```

Out[18]:

```
<matplotlib.legend.Legend at 0x1f83d8ad220>
```



- From above graph we got legend but at bottom to get at top right side:

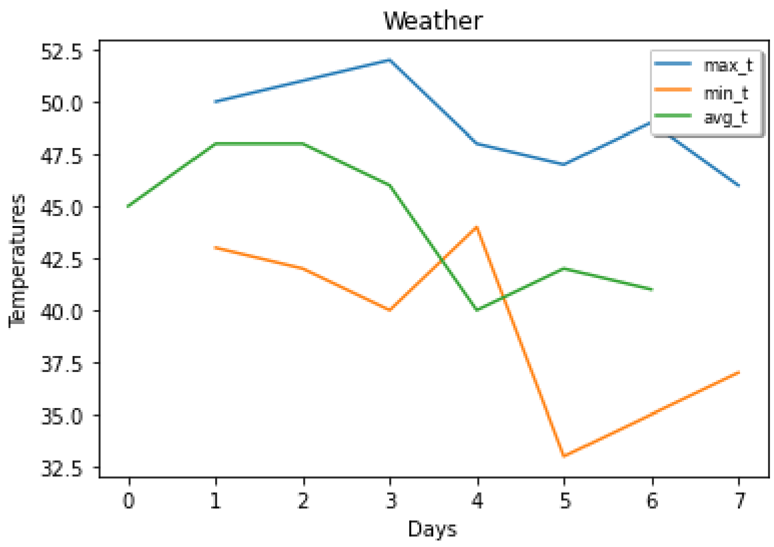
In [23]:

```
plt.xlabel("Days")
plt.ylabel("Temperatures")
plt.title("Weather")
plt.plot(days,max_t ,label="max_t")
plt.plot(days,min_t ,label="min_t")
plt.plot(avg_t ,label = "avg_t")

plt.legend(loc="upper right" , shadow = True , fontsize = "small")    # if loc = "best"  gi
```

Out[23]:

```
<matplotlib.legend.Legend at 0x1f83daf2d90>
```

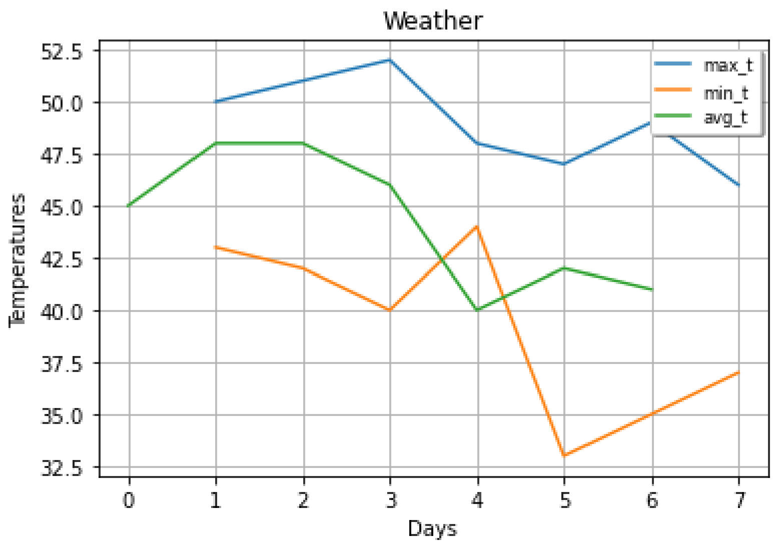


grid()

In [24]:

```
plt.xlabel("Days")
plt.ylabel("Temperatures")
plt.title("Weather")
plt.plot(days,max_t ,label="max_t")
plt.plot(days,min_t ,label="min_t")
plt.plot(avg_t ,label = "avg_t")

plt.legend(loc="upper right" , shadow = True , fontsize = "small")    # if loc = "best"  gi
plt.grid()
```



bar plot

Example

In [40]:

```
company=['GOOGLE','AMAZON','MICROSOFT','FACEBOOK']
revenue=[90,136,89,27]
profit=[40,2,34,12]
```

In [28]:

```
# Since elements in company is strings, we convert to ndarrays
xpos = np.arange(len(company))
xpos
```

Out[28]:

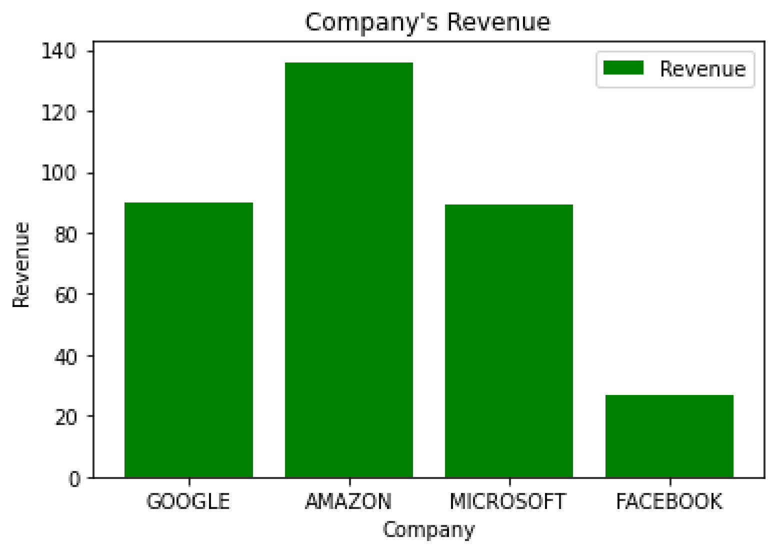
```
array([0, 1, 2, 3])
```

In [43]:

```
plt.xlabel("Company")
plt.ylabel("Revenue")
plt.title("Company's Revenue")
plt.xticks(xpos,company)  # To get company names on x-axis
plt.bar(company,revenue , color ='g',label="Revenue")
plt.legend()
```

Out[43]:

```
<matplotlib.legend.Legend at 0x1f83df70fd0>
```



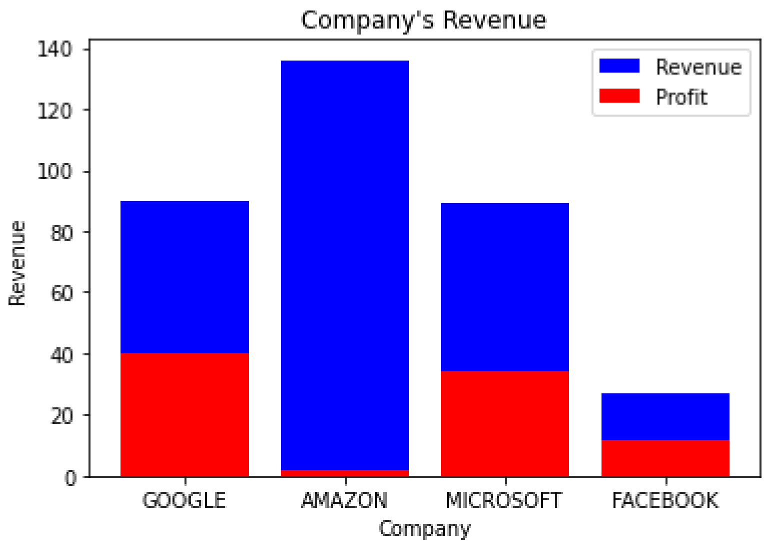
Plotting '2' different values in same plot

In [44]:

```
plt.xlabel("Company")
plt.ylabel("Revenue")
plt.title("Company's Revenue")
plt.xticks(xpos,company)  # To get company names on x-axis
plt.bar(company,revenue , color ='b',label="Revenue")
plt.bar(company,profit,color = 'r',label ='Profit')
plt.legend()
```

Out[44]:

```
<matplotlib.legend.Legend at 0x1f83dfee580>
```



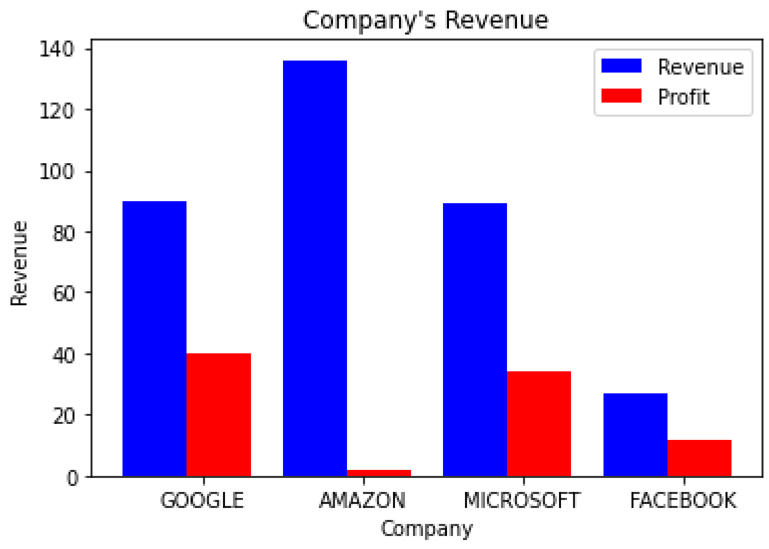
To get graphs side by side in single plot

In [49]:

```
plt.xlabel("Company")
plt.ylabel("Revenue")
plt.title("Company's Revenue")
plt.xticks(xpos,company)  # To get company names on x-axis
plt.bar(xpos-0.3,revenue,width = 0.4, color ='b',label="Revenue")
plt.bar(xpos+0.1,profit,width=0.4,color = 'r',label ='Profit')
plt.legend()
```

Out[49]:

```
<matplotlib.legend.Legend at 0x1f83dadb9d0>
```

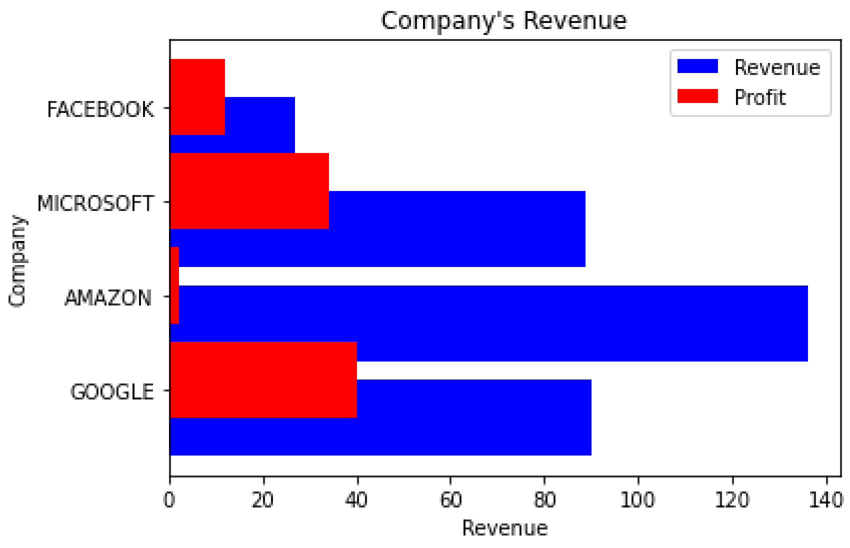


Horizontal bar chat

barh

In [80]:

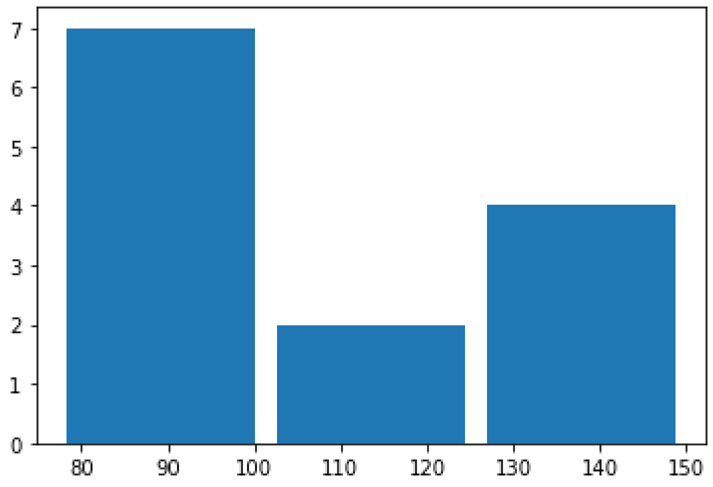
```
plt.xlabel("Revenue")
plt.ylabel("Company")
plt.title("Company's Revenue")
plt.yticks(xpos,company)  # To get company names on x-axis
plt.barh(xpos-0.3,revenue, color ='b',label="Revenue")
plt.barh(xpos+0.1,profit,color = 'r',label ='Profit')
plt.legend()
plt.show()
```



histograms

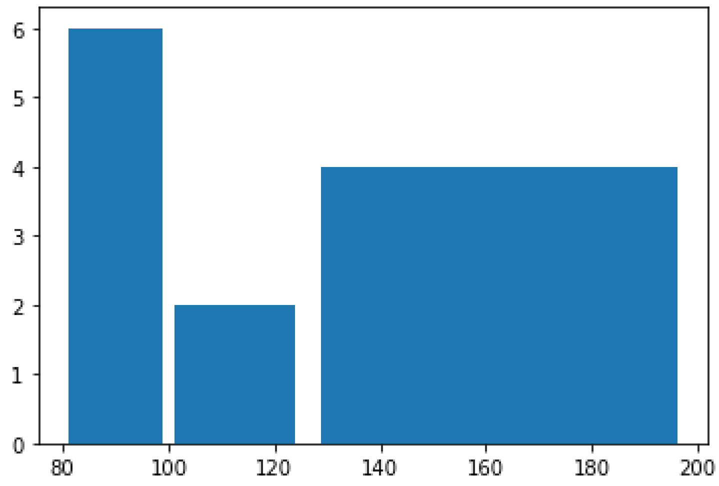
In [79]:

```
blood_sugar = [113, 85, 90, 150, 149, 88, 93, 115, 135, 80, 77, 82, 129]
plt.hist(blood_sugar,bins=3,rwidth=0.9) # By default y axis is frequency
plt.show()
```



In [78]:

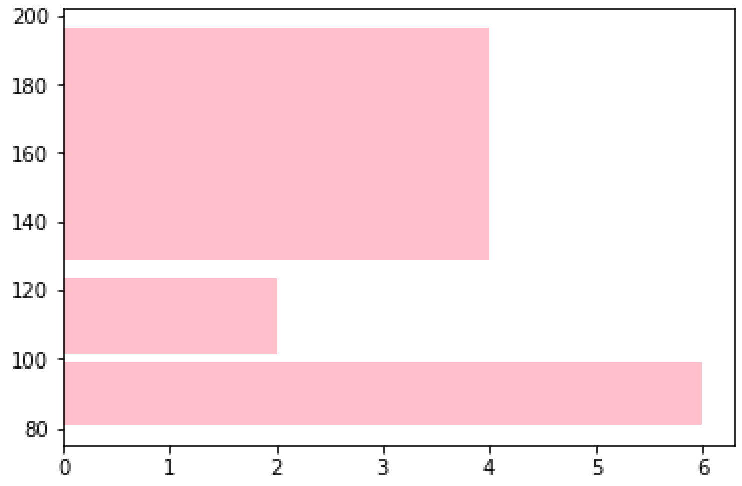
```
blood_sugar = [113, 85, 90, 150, 149, 88, 93, 115, 135, 80, 77, 82, 129]
plt.hist(blood_sugar,bins=[80,100,125,200],rwidth=0.9) # bins range
plt.show()
```



orientation, plots histogram horizontally

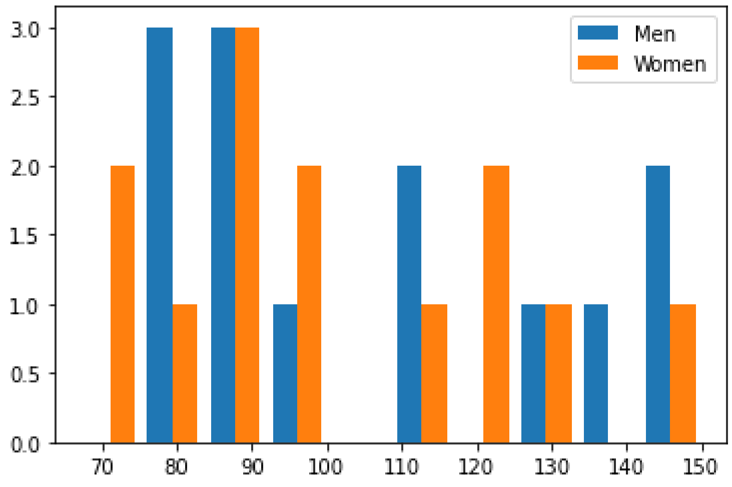
In [77]:

```
blood_sugar = [113, 85, 90, 150, 149, 88, 93, 115, 135, 80, 77, 82, 129]
plt.hist(blood_sugar,bins=[80,100,125,200],color = 'pink',rwidth=0.9,orientation ='horizont
plt.show()
```



In [76]:

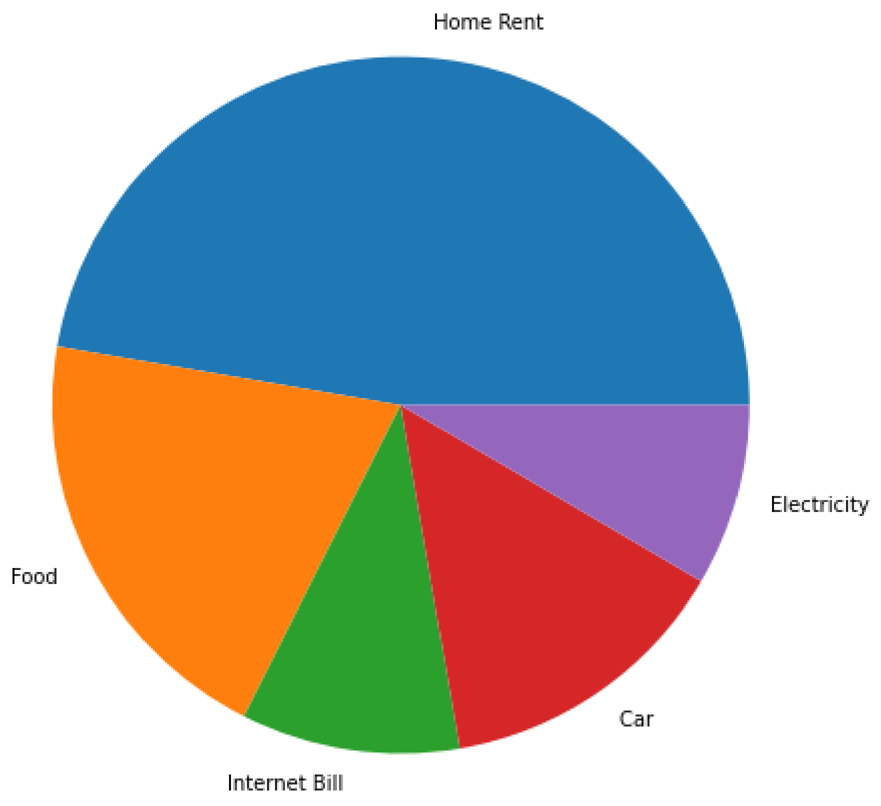
```
blood_sugar_men = [113, 85, 90, 150, 149, 88, 93, 115, 135, 80, 77, 82, 129]
blood_sugar_women = [67, 98, 89, 120, 133, 150, 84, 69, 89, 79, 120, 112, 100]
plt.hist([blood_sugar_men,blood_sugar_women],label = ['Men','Women'])
plt.legend()
plt.show()
```



Piecharts

In [81]:

```
value = [1400,600,300,410,250]
value_labels = ["Home Rent","Food","Internet Bill","Car ","Electricity"]
plt.pie(value,labels= value_labels,radius = 2)
plt.show()
```

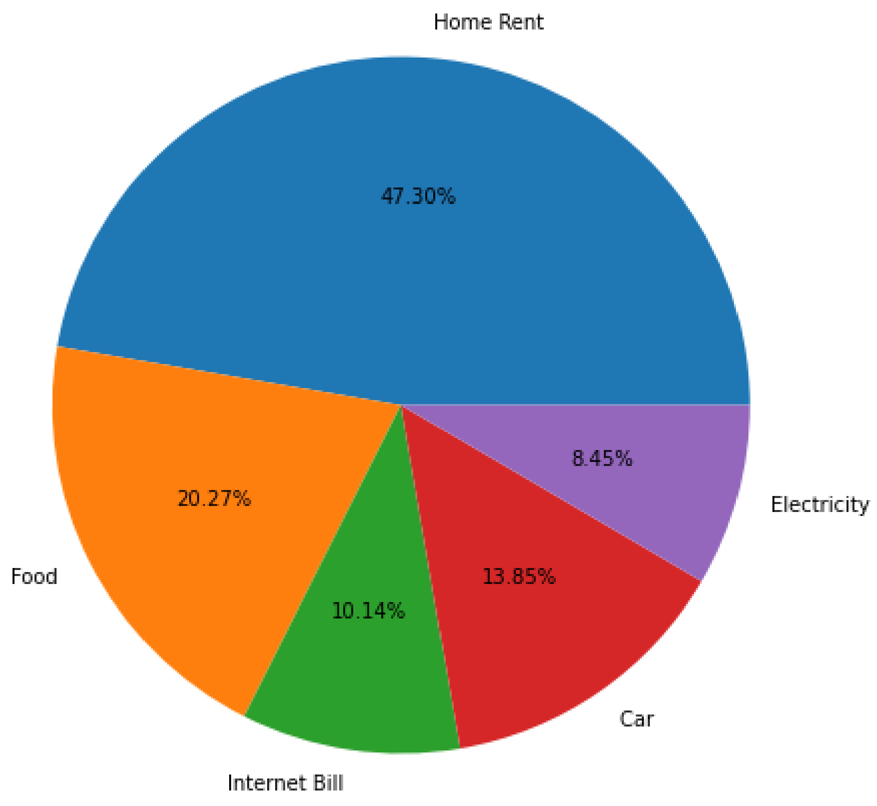


To show '%' in pie charts

autopct = '%.2f%%'

In [82]:

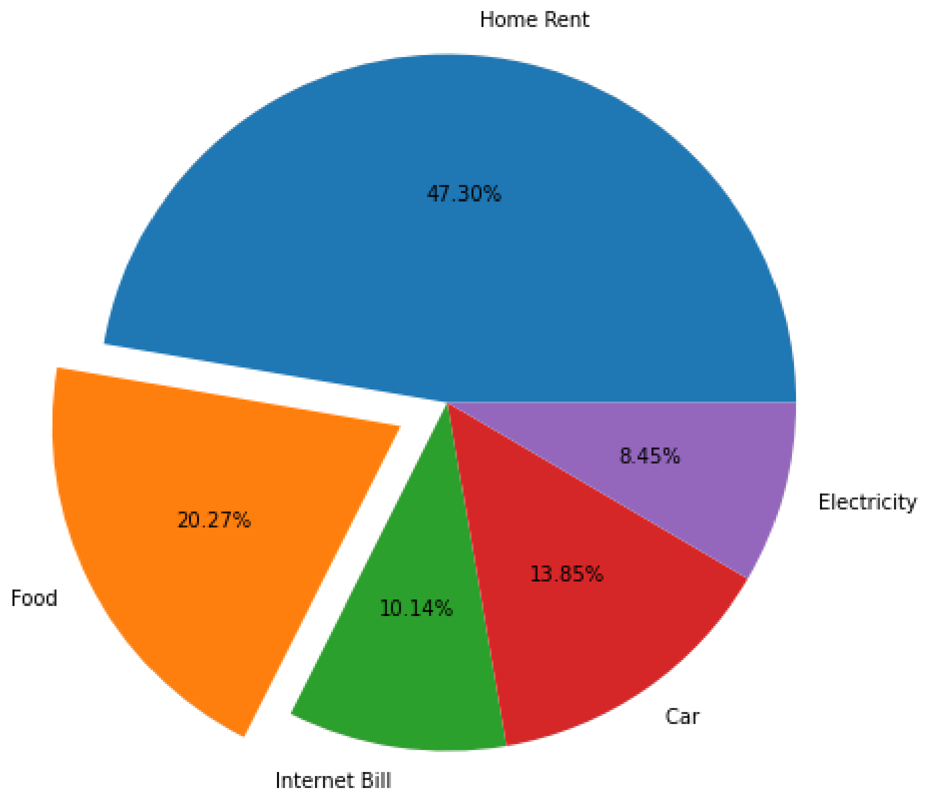
```
value = [1400,600,300,410,250]
value_labels = ["Home Rent","Food","Internet Bill","Car ","Electricity"]
plt.pie(value,labels= value_labels,radius = 2,autopct = '%.2f%%')
plt.show()
```



explode

In [83]:

```
value = [1400,600,300,410,250]
value_labels = ["Home Rent","Food","Internet Bill","Car ","Electricity"]
plt.pie(value,labels= value_labels,radius = 2,autopct = '%.2f%%',explode = [0,0.3,0,0,0])
plt.show()
```

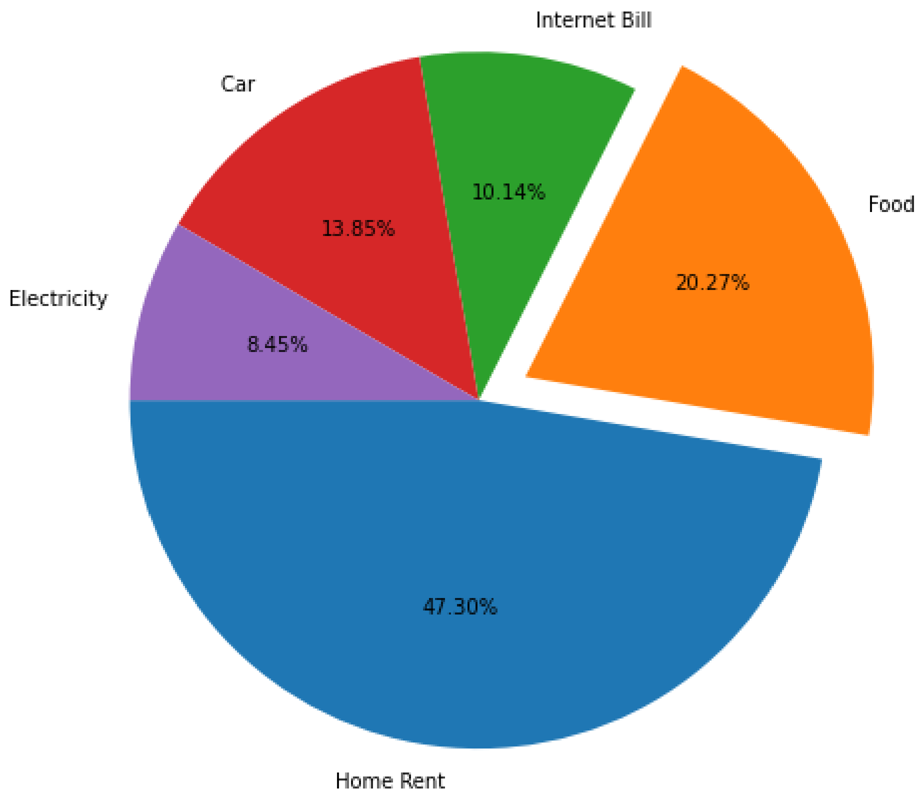


Rotating pie chart

startangle

In [84]:

```
value = [1400,600,300,410,250]
value_labels = ["Home Rent","Food","Internet Bill","Car ","Electricity"]
plt.pie(value,labels= value_labels,radius = 2,autopct = '%.2f%%',explode = [0,0.3,0,0,0],st
plt.show()
```



Saving plots as images to directory

savefig()

In [ ]:

```
# saves to directory as jpg image
plt.savefig("piechart.jpg", bbox_inches="tight", pad_inches=1, transparent=True)
```

In [ ]:

```
# saves to specified location as pdf file
plt.savefig("c:/code/piechart.pdf", bbox_inches="tight", pad_inches=10, transparent=True)
```