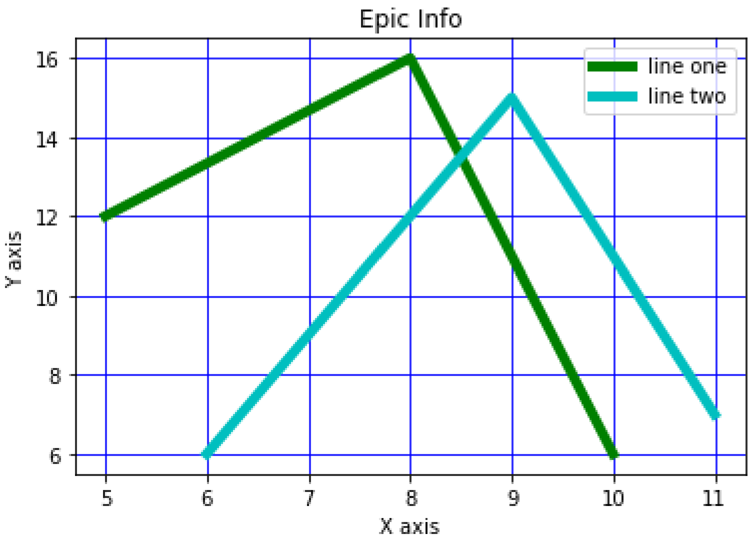
```
I have created dummy data to show how we visualise the data by using
matplotlib and seaborn. Note down the syntax that I have used.
```

Matplotlib

LinePlot

In [3]:

```
%matplotlib inline
import matplotlib.pyplot as plt
x = [5,8,10]
y = [12,16,6]
x2 = [6,9,11]
y2 = [6,15,7]
plt.plot(x,y,'g',label='line one', linewidth=5)
plt.plot(x2,y2,'c',label='line two',linewidth=5)
plt.title('Epic Info')
plt.ylabel('Y axis')
plt.xlabel('X axis')
plt.legend() #gives the information about the lines at the top right
plt.grid(True,color='b')
plt.show()
```

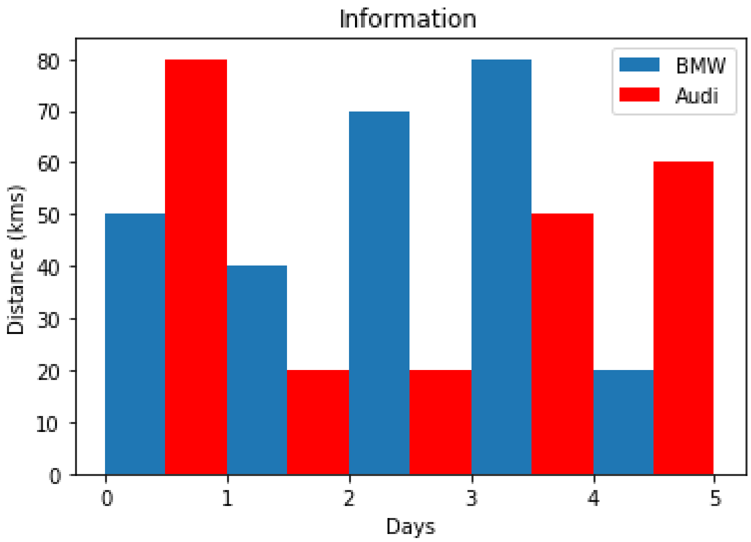


Bar Plot

Use for comparing the periodic change of quantities.

In [4]:

```
# at 0 it plots at the center of the number, with 0.25 it shifts a 0.25 dist
'''change the width to 0.33 for 3 bars, now for singlw bar the half part is
use the distance as 0.165*1 , 0.165*3 , 0.165*5 and so on'''
plt.bar([0.25,1.25,2.25,3.25,4.25],[50,40,70,80,20],label="BMW",width=.5)
plt.bar([.75,1.75,2.75,3.75,4.75],[80,20,20,50,60],label="Audi", color='r',w
# plt.bar([.825,1.75,2.75,3.75,4.75],[10,20,30,40,50],label="Benz", color='k
plt.legend()
plt.xlabel('Days')
plt.ylabel('Distance (kms)')
plt.title('Information')
plt.show()
```

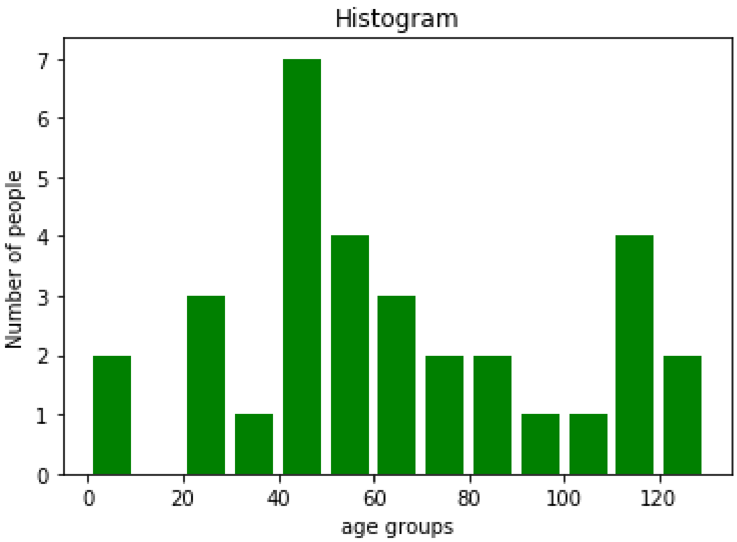


Histograms

Provides the distribution of the number of values, falling within the range specified in bins

In [5]:

```
population_age = [22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,118,70,65,
bins = [0,10,20,30,40,50,60,70,80,90,100,110,120,130]
plt.hist(population_age, bins, histtype='bar', rwidth=0.8, color = 'g')
plt.xlabel('age groups')
plt.ylabel('Number of people')
plt.title('Histogram')
plt.show()

```

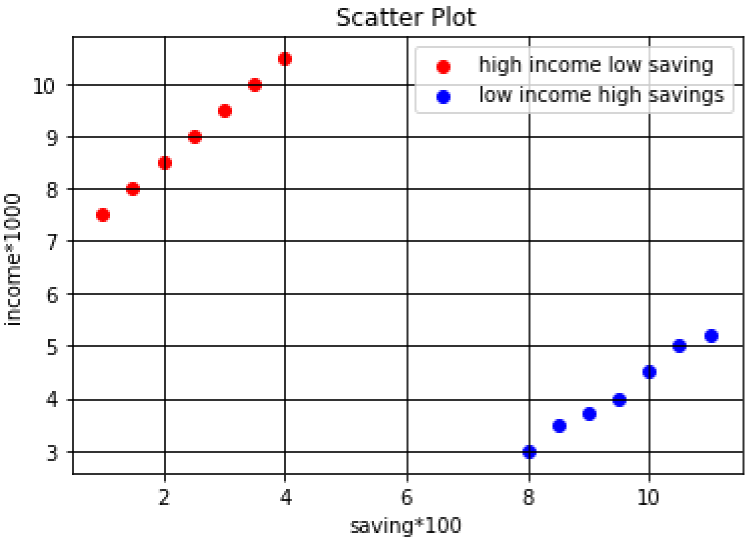


ScatterPlot

In order to compare variables

In [6]:

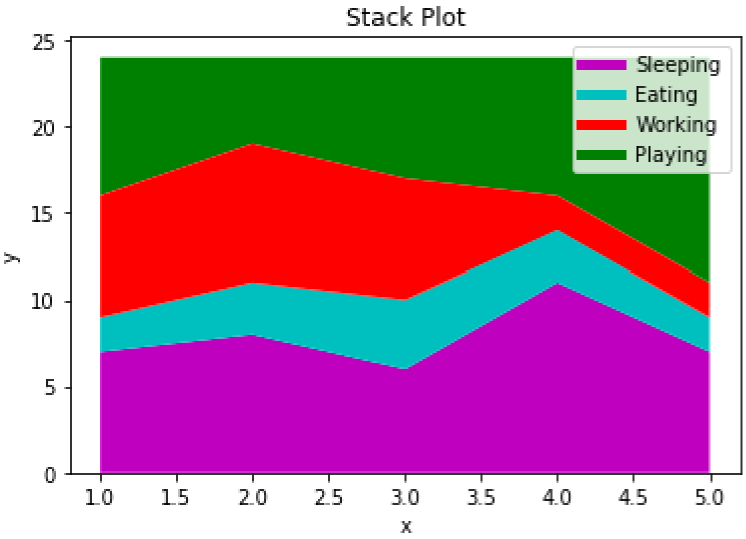
```
x = [1,1.5,2,2.5,3,3.5,4]
y = [7.5,8,8.5,9,9.5,10,10.5]

x1=[8,8.5,9,9.5,10,10.5,11]
y1=[3,3.5,3.7,4,4.5,5,5.2]

plt.scatter(x,y, label='high income low saving',color='r')
plt.scatter(x1,y1,label='low income high savings',color='b')
plt.xlabel('saving*100')
plt.ylabel('income*1000')
plt.title('Scatter Plot')
plt.grid(True, color='k')
plt.legend()
plt.show()
```



Area Plot

In [7]:

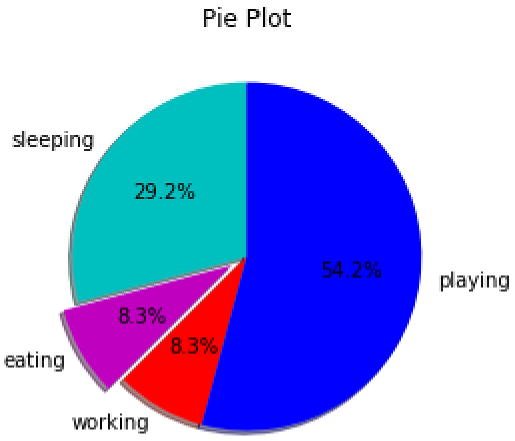
```python
import matplotlib.pyplot as plt
days = [1,2,3,4,5]

sleeping =[7,8,6,11,7]
eating = [2,3,4,3,2]
working =[7,8,7,2,2]
playing = [8,5,7,8,13]

plt.plot([],[],color='m', label='Sleeping', linewidth=5)
plt.plot([],[],color='c', label='Eating', linewidth=5)
plt.plot([],[],color='r', label='Working', linewidth=5)
plt.plot([],[],color='g', label='Playing', linewidth=5)

plt.stackplot(days, sleeping,eating,working,playing, colors=['m','c','r','g'

plt.xlabel('x')
plt.ylabel('y')
plt.title('Stack Plot')
plt.legend()
plt.show()

```



Pie-Chart

In [8]:

```python
import matplotlib.pyplot as plt

# days = [1,2,3,4,5]

# sleeping =[7,8,6,11,7]
# eating = [2,3,4,3,2]
# working =[7,8,7,2,2]
# playing = [8,5,7,8,13]
slices = [7,2,2,13]
activities = ['sleeping','eating','working','playing']
cols = ['c','m','r','b']

plt.pie(slices,
  labels=activities,
  colors=cols,
  startangle=90,
  shadow= True,
  explode=(0,0.1,0,0),
  autopct='%1.1f%%')

plt.title('Pie Plot')
plt.show()
```

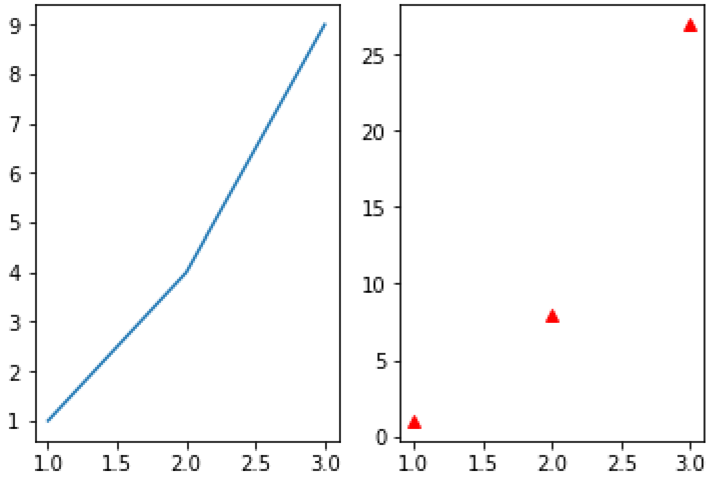


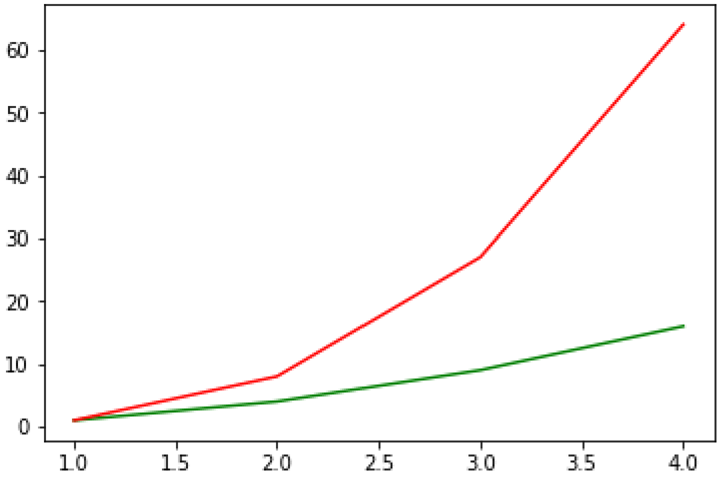
Subplots:

Subplots help in plotting multiple graphs in one figure

In [32]:

```
plt.subplot(1,2,1)
plt.plot([1,2,3],[1,4,9])
# plt.show()
plt.subplot(1,2,2)
plt.plot([1,2,3],[1,8,27],"r^")
plt.show()


import numpy as np
x = np.arange(1,5)
y = x**3
plt.plot([1,2,3,4],[1,4,9,16],'g',x,y,'r')
plt.show()
```





Seaborn

It is a plotting library. Use for plotting dataframe objects from pandas and has beautiful design and styles. pip install seaborn or conda install seaborn Mandatory dependencies numpy,scipy,matplotlib,pandas

In [2]:

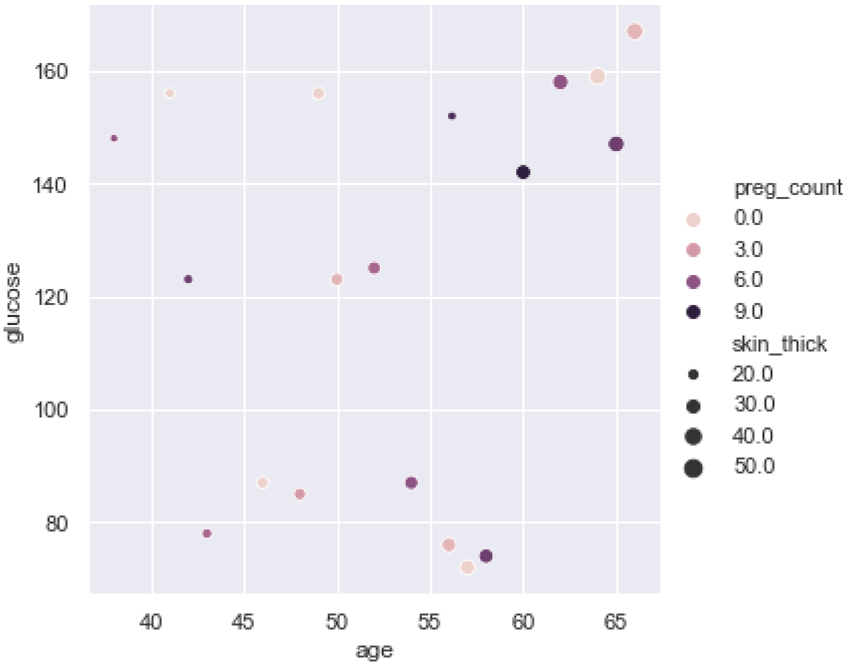
```
import seaborn as sns
import pandas as pd
import numpy as np
```

In [8]:

```
df = pd.read_excel(r"C:\Users\admin\Desktop\Pandas_sample_data - Copy.xlsx")
# print(df.head())
df.describe()
#visualising is very imp to find out the relationship between the variables
#variables
df['glucose']=df['glucose'].replace(1000,df['glucose'].mean())
df['age']=df['age'].replace(150,df['age'].mean())
sns.set(style='darkgrid') #provides a dark background instead of white with
# sns.scatterplot(x='age',y='glucose',hue='preg_count',size='skin_thick',dat
sns.relplot(x='age',y='glucose',hue='preg_count',size='skin_thick',height=5,
# sns.lmplot(x='age',y='glucose',data=df,fit_reg=True) #gives regression lin
#sns.relplot(kind='scatter') with relplot set the kind of the graph required
#In order to show multiple parameters use "hue","size","col(to differentiate
'''Anaconda3\lib\site-packages\seaborn\relational.py in scatterplot(x, y, hu
```

Out[8]:

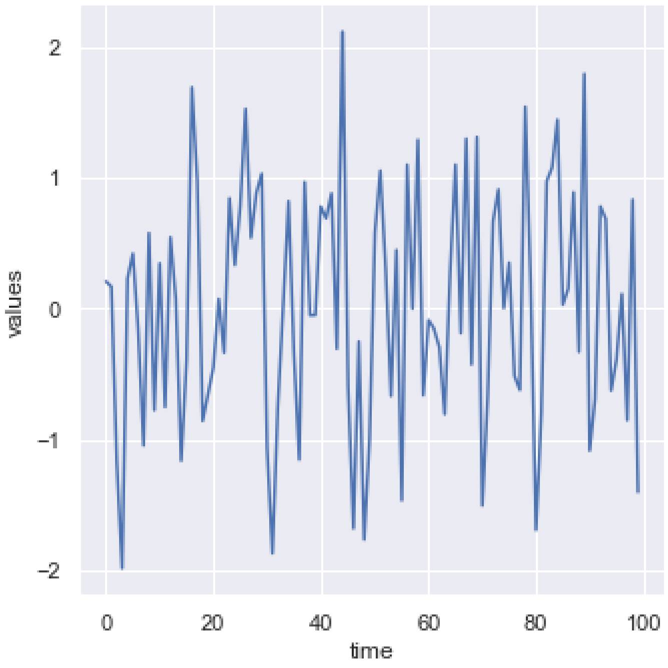
```
'Anaconda3\\lib\\site-packages\\seaborn\\relational.py in scatterplot(x, y, hue,
style, size, data, palette, hue_order, hue_norm, sizes, size_order, size_norm,
markers, style_order, x_bins, y_bins, units, estimator, ci, n_boot, alpha, x_ji
tter, y_jitter, legend, ax, **kwargs)'
```



With kind='x' in categorical plot catplot() we can plot different types of plot: Categorical scatterplots: stripplot() (with kind=”strip”; the default) swarmplot() (with kind=”swarm”) Categorical distribution plots: boxplot() (with kind=”box”) violinplot() (with kind=”violin”) boxenplot() (with kind=”boxen”) Categorical estimate plots: pointplot() (with kind=”point”) barplot() (with kind=”bar”) countplot() (with kind=”count”)

In [61]:
```
df1=pd.DataFrame(dict(time = np.arange(100),values= np.random.randn(100)))
sns.relplot(x='time',y='values',kind='line', data=df1)
```

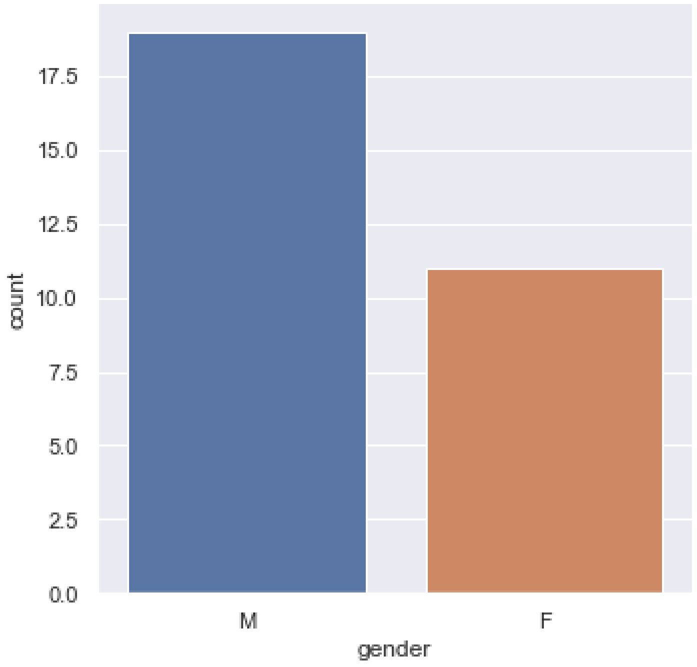
Out[61]:
```
<seaborn.axisgrid.FacetGrid at 0xe26fb7c518>
```



categorical plot in seaborn: catplot()

In [62]:
```
sns.catplot(x='gender',kind='count',data=df,orient='h') #this gives the coun
#graph used with only one variable
```

Out[62]:
```
<seaborn.axisgrid.FacetGrid at 0xe26fba00f0>
```



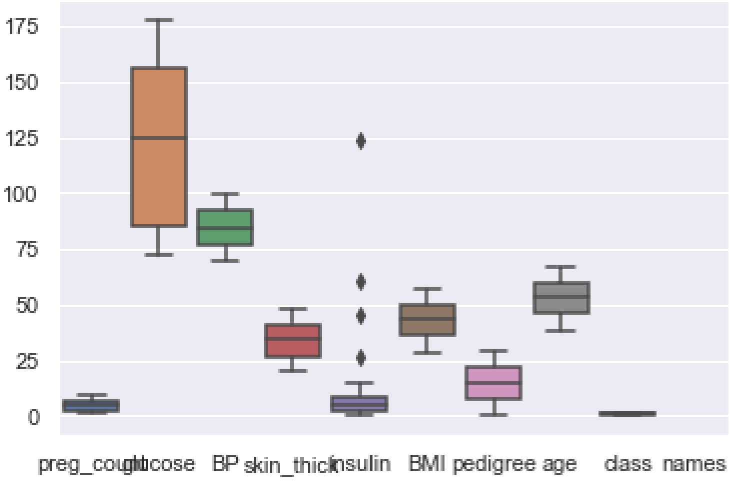
sns.rlplot() relational plot sns.catplot() categorical data plot , that takes only a limited value like gender =(male,female) sns.implot() use for plotting linearr regression plot

In [72]:

```
sns.boxplot(data=df)
```

Out[72]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xe26f1c9c88>
```



In [79]:

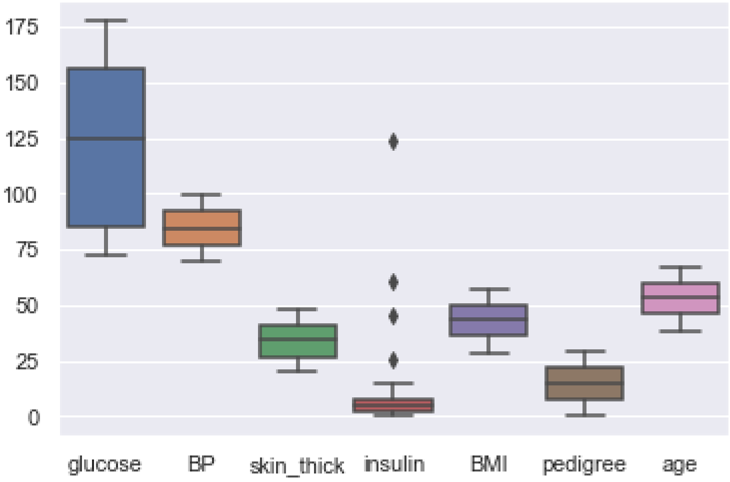
```
df=df.drop(['preg_count','names','class'],axis=1)
```

In [80]:

```
sns.boxplot(data=df) #boxplot depicts the value range of the fields(various
sns.swarmplot(x,y,dataset)
sns.set_style('whitegrid')
sns.violinplot(x,y,dataset)
```

Out[80]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xe26ee2ac18>
```



In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```