In [7]:
```
import pandas as pd
dict={'names':['rohan','amar','ashwini','lohith','mohan'],
     'type':['regular','adhoc','regular','adhoc','contract'],
     'department':['cs','cs','ec','ec','cs'],
     'experience':['10','20','5','14','9'],
     'salary':['5000','15000','2000','3000','4000']}
d=pd.DataFrame(dict)
print(d)
```

```
     names      type department experience salary
0    rohan   regular         cs         10   5000
1     amar     adhoc         cs         20  15000
2  ashwini   regular         ec          5   2000
3   lohith     adhoc         ec         14   3000
4    mohan  contract         cs          9   4000
```

In [8]:
```
d.type.max()
```

Out[8]:
```
'regular'
```

In [9]:
```
d.type.min()
```

Out[9]:
```
'adhoc'
```

In [14]:
```
d.mean()
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_6288\1214089992.py:1: FutureW
arning: The default value of numeric_only in DataFrame.mean is deprecated.
In a future version, it will default to False. In addition, specifying 'nu
meric_only=None' is deprecated. Select only valid columns or specify the v
alue of numeric_only to silence this warning.
  d.mean()
```

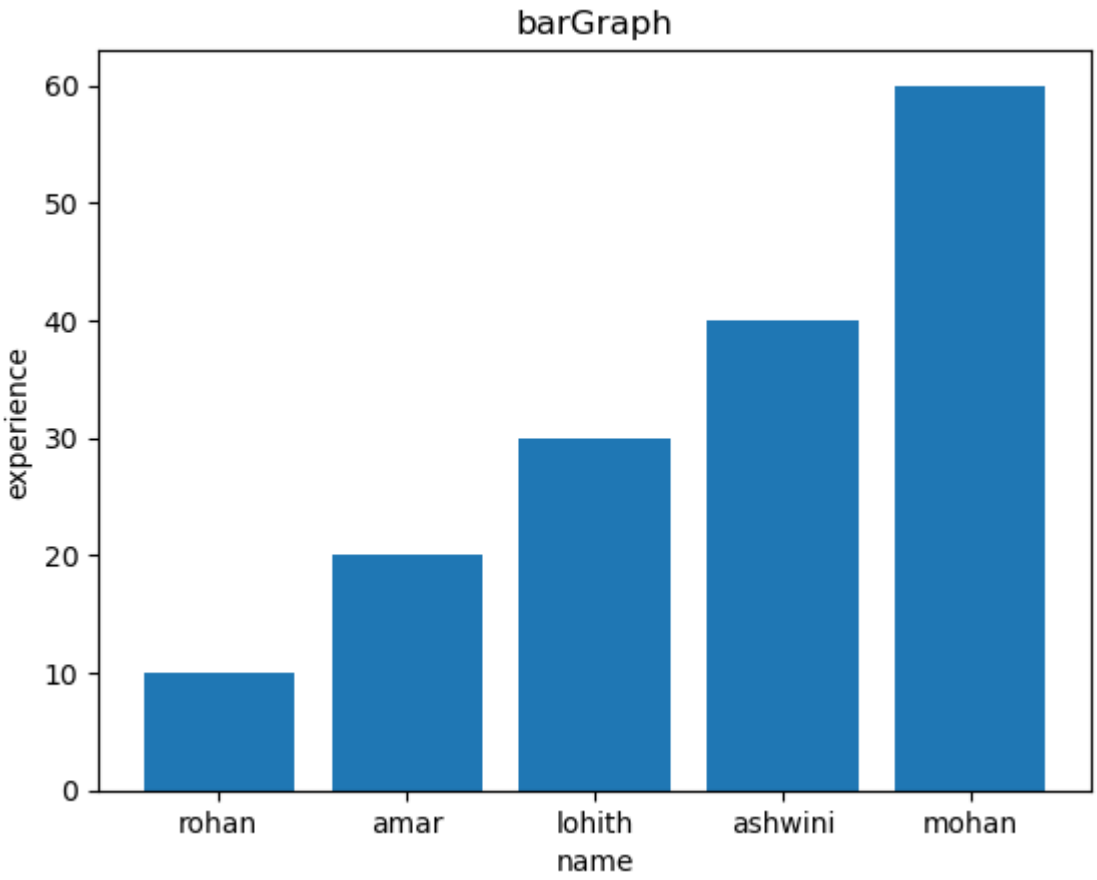
Out[14]:
```
experience    2.041030e+06
salary        1.000030e+20
dtype: float64
```

In [15]:
```
d.count()
```

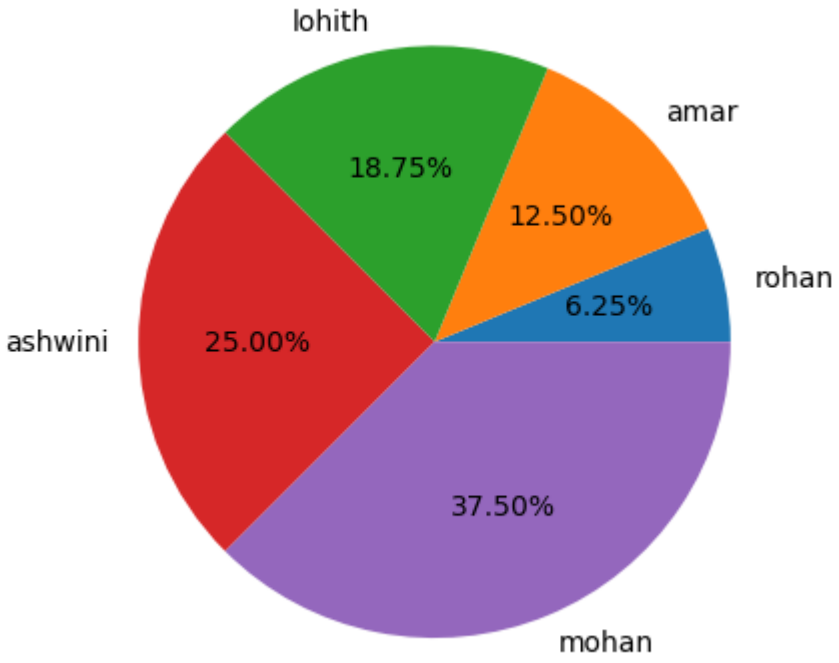
Out[15]:
```
names         5
type          5
department    5
experience    5
salary        5
dtype: int64
```

In [19]:
```
name=["rohan","amar","lohith","ashwini","mohan"]
experience=[10,20,30,40,60]
plt.bar(name,experience)
plt.xlabel("name")
plt.ylabel("experience")
plt.title("barGraph")
plt.show
```

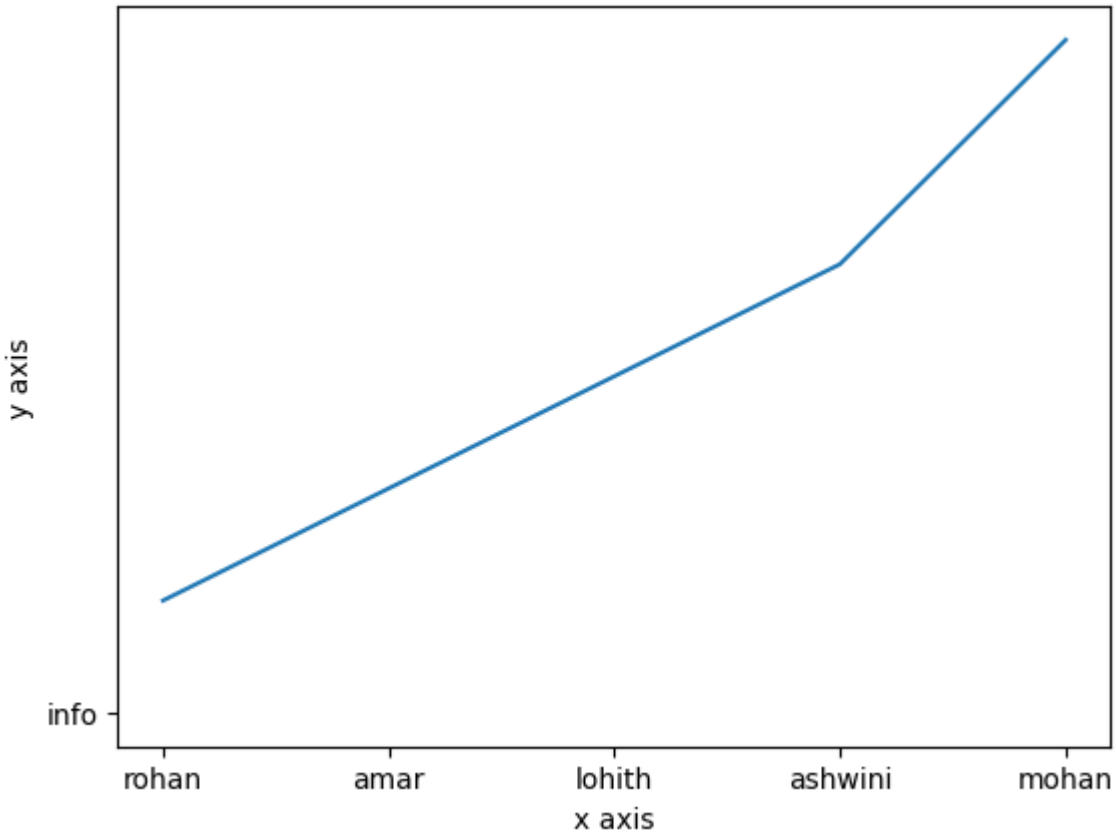
Out[19]: <function matplotlib.pyplot.show(close=None, block=None)>



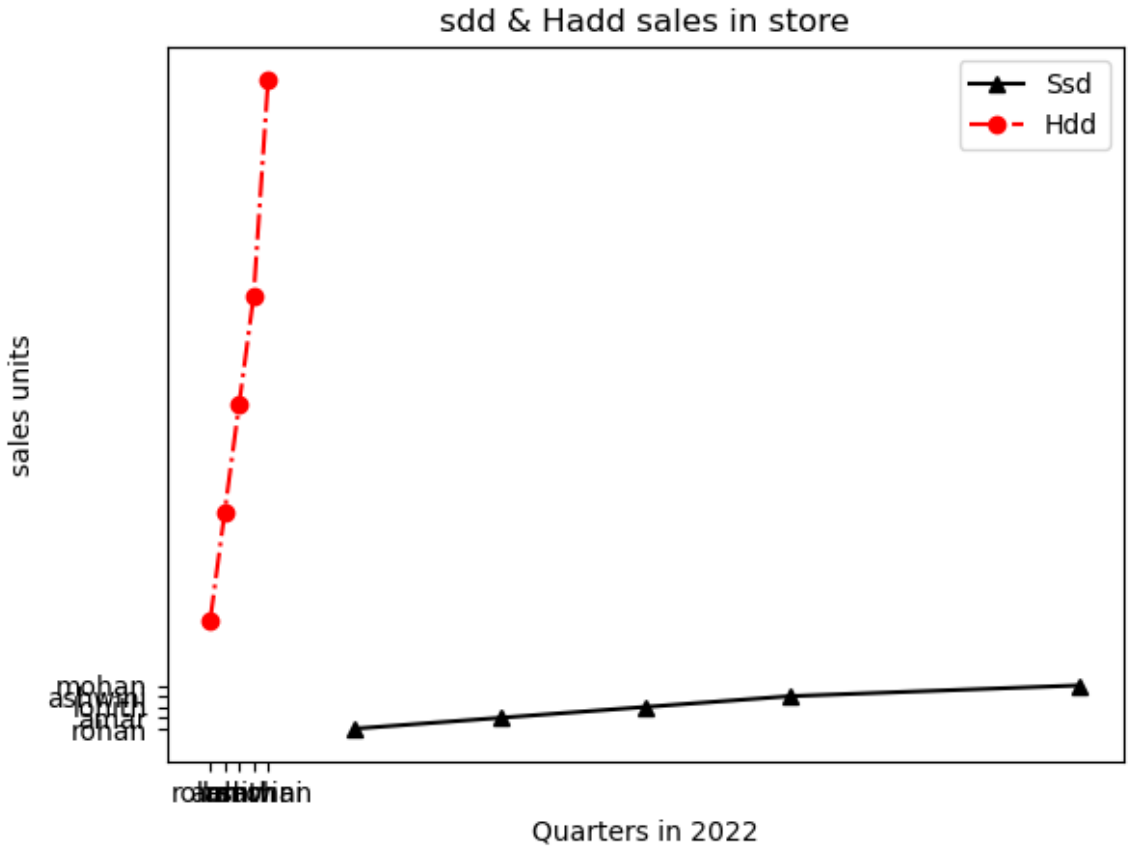
In [24]:
```
name=["rohan","amar","lohith","ashwini","mohan"]
experience=[10,20,30,40,60]
plt.pie(experience,labels=name,autopct='%1.2f%%')
plt.show()
```



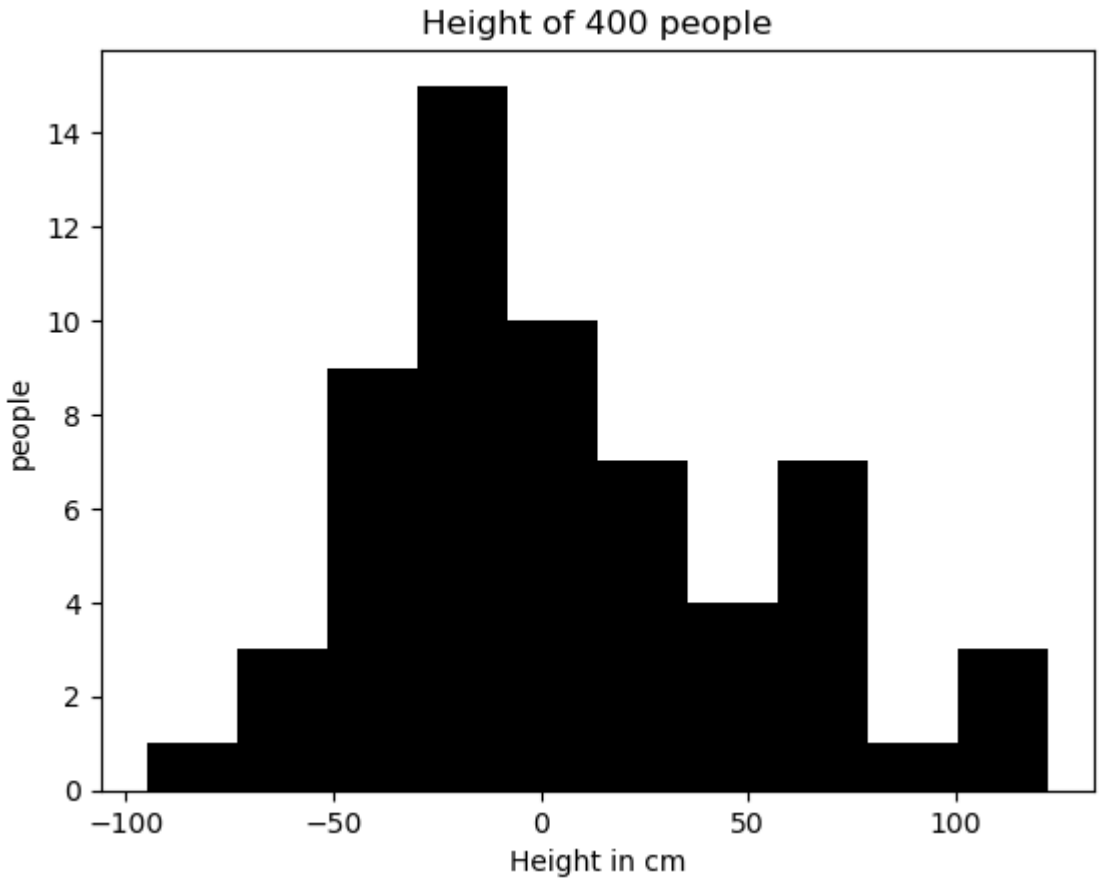
In [12]:
```
from matplotlib import pyplot as plt
x=["rohan","amar","lohith","ashwini","mohan"]
y=[10,20,30,40,60]
plt.plot(x,y)
plt.plot('info')
plt.ylabel('y axis')
plt.xlabel('x axis')
plt.show()
```



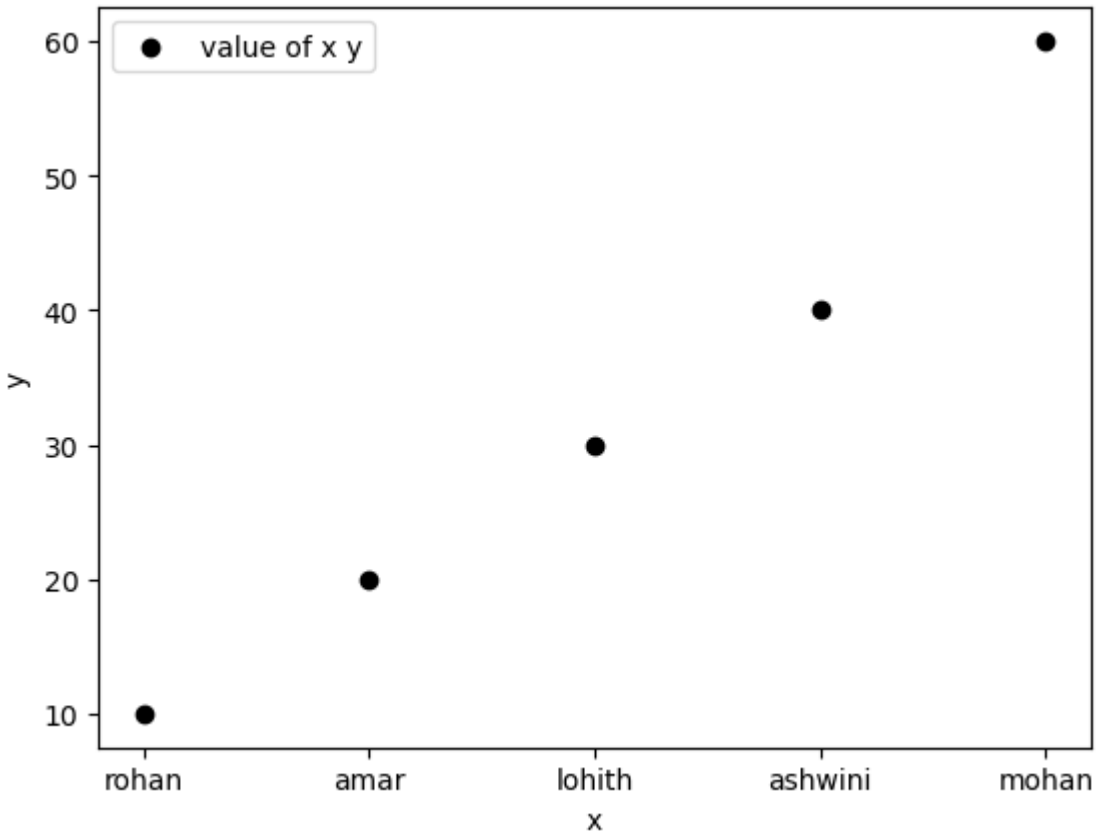
In [14]:
```
from matplotlib import pyplot as plt
ssd=["rohan","amar","lohith","ashwini","mohan"]
hdd=[10,20,30,40,60]
plt.plot(hdd,ssd,'^-',color='black')
plt.plot(ssd,hdd,'o-.r')
plt.xlabel("Quarters in 2022"),plt.ylabel("sales units")
plt.title("sdd & Hadd sales in store")
plt.legend(['Ssd','Hdd'])
plt.show()
```



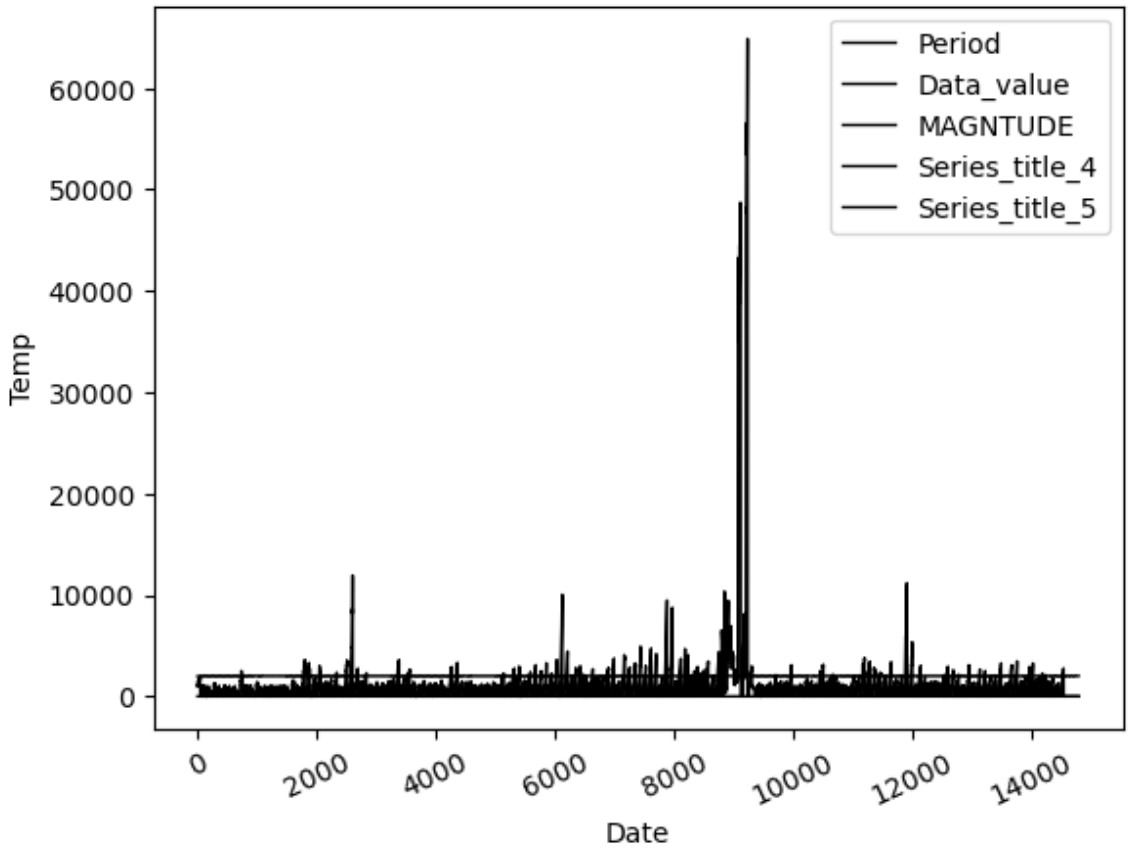
In [17]:
```
from matplotlib import pyplot as p
import numpy as n
x=n.random.normal(10,40,60)
p.hist(x,color='k')
p.xlabel("Height in cm"),p.ylabel("people")
p.title("Height of 400 people")
p.show()
```



In [18]:
```
from matplotlib import pyplot as p
x=["rohan","amar","lohith","ashwini","mohan"]
y=[10,20,30,40,60]
c=['k','b']
p.scatter(x,y,label='value of x y',color='k')
p.xlabel('x')
p.ylabel('y')
p.legend()
p.show()
```



In [24]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
t = pd.read_csv("C:/Users/lenovo/Documents/ps-leela.csv")
t
a = t.plot(color='k', linewidth=1)
plt.xticks(rotation=25)
a.set_ylabel('Temp')
plt.xlabel('Date')
plt.show()
```



In [4]:
```
import seaborn as s
import pandas as pd
import matplotlib.pyplot as p
d=s.load_dataset('iris')
p.figure(figsize=(10,6))
s.boxplot(x="species",y="sepal_lenght",data=d)
p.show()
```

```
---------------------------------------------------------------------------
gaierror                                  Traceback (most recent call la
st)
File ~\anaconda3\Lib\urllib\request.py:1348, in AbstractHTTPHandler.do_o
pen(self, http_class, req, **http_conn_args)
   1347 try:
-> 1348     h.request(req.get_method(), req.selector, req.data, headers,
   1349               encode_chunked=req.has_header('Transfer-encodin
g'))
   1350 except OSError as err: # timeout error

File ~\anaconda3\Lib\http\client.py:1286, in HTTPConnection.request(sel
f, method, url, body, headers, encode_chunked)
   1285 """Send a complete request to the server."""
-> 1286 self._send_request(method, url, body, headers, encode_chunked)

File ~\anaconda3\Lib\http\client.py:1332, in HTTPConnection._send_reques
t(self, method, url, body, headers, encode_chunked)
```

In [ ]:

In [ ]: