In [1]:
```
import pandas as pd
```

In [2]:
```
df1 = pd.read_csv('E:\\emp.csv')
df1
```

Out[2]:

	eid	ename	edept	eplace	ecost
0	101	raj	sales	pune	1000
1	102	leo	prod	bglore	2000
2	103	paul	HR	chennai	3000
3	104	anu	hr	hyderabad	4000
4	456	kumar	sales	bglore	3000
5	105	zion	Hr	mumbai	5000
6	106	bibu	sales	bglore	1450
7	107	theeb	sales	noida	4590
8	108	bibu	sales	bglore	5000
9	113	kumar	prod	hyderabad	5423

In [3]:
```
d={'K1':[10,20,30],'K2':[11,12,13]}


pd.DataFrame(d)
```

Out[3]:

	K1	K2
0	10	11
1	20	12
2	30	13

In [4]:
```
df2 = pd.DataFrame(d)
df2.columns
```

Out[4]:
```
Index(['K1', 'K2'], dtype='object')
```

In [5]:
```
df2['K3']=[15,25,35] # Add new data to an existing dataframe -like dict
df2
```

Out[5]:

	K1	K2	K3
0	10	11	15
1	20	12	25
2	30	13	35

In [7]:
```
df2
```

Out[7]:

	K1	K2	K3
0	10	11	15
1	20	12	25
2	30	13	35

In [9]:
```
# remove any specific columns
df2.drop('K2',axis='columns')
```

Out[9]:

	K1	K3
0	10	15
1	20	25
2	30	35

In [10]:
```
df2
```

Out[10]:

	K1	K2	K3
0	10	11	15
1	20	12	25
2	30	13	35

In [11]:
```
# remove any specific columns
df2.drop('K2',axis='columns',inplace=True)
# <or>
# df2.drop('K2',axis=1,inplace=True)
```

In [12]:
```
df2
```

Out[12]:

	K1	K3
0	10	15
1	20	25
2	30	35

In [13]:
```
df2.index
```

Out[13]:
```
RangeIndex(start=0, stop=3, step=1)
```

In [14]:
```
df2.drop(1) # remove 1st index
```

Out[14]:

	K1	K3
0	10	15
2	30	35

In [15]:
```
df2.drop(1,axis='index') # <or> df2.drop(1,axis=0)
```

Out[15]:

	K1	K3
0	10	15
2	30	35

In [16]:
```
df2
```

Out[16]:

	K1	K3
0	10	15
1	20	25
2	30	35

In [17]:
```
df2.drop(1,axis='index',inplace=True) # <or> df2.drop(1,axis=0,inplace=True)
```

In [18]:
```
df2
```

Out[18]:

	K1	K3
0	10	15
2	30	35

In [19]:
```
pd.read_csv('E:\\emp.csv')
```

Out[19]:

	eid	ename	edept	eplace	ecost
0	101	raj	sales	pune	1000
1	102	leo	prod	bglore	2000
2	103	paul	HR	chennai	3000
3	104	anu	hr	hyderabad	4000
4	456	kumar	sales	bglore	3000
5	105	zion	Hr	mumbai	5000
6	106	bibu	sales	bglore	1450
7	107	theeb	sales	noida	4590
8	108	bibu	sales	bglore	5000
9	113	kumar	prod	hyderabad	5423

In [22]:
```
import json
```

In [25]:
```
d={}
d['Type']=['eth0','eth1','eth2']
d['IP']=['10.20.30.40','11.23.45.45','10.20.34.55']
d['port']=[3000,4000,5000]
d['app']=['gl','fl','pm']

wobj = open('E:\\test.json','w')
json.dump(d,wobj)
wobj.close()
```

In [26]:
```
pd.read_json('E:\\test.json')
```

Out[26]:

	Type	IP	port	app
0	eth0	10.20.30.40	3000	gl
1	eth1	11.23.45.45	4000	fl
2	eth2	10.20.34.55	5000	pm

In [28]:
```
df3 = pd.read_json('E:\\test.json')
df3.index
```

Out[28]:
```
RangeIndex(start=0, stop=3, step=1)
```

In [29]:
```
df3.columns
```

Out[29]:
```
Index(['Type', 'IP', 'port', 'app'], dtype='object')
```

In [30]:
```
df3['Type']
```

Out[30]:
```
0    eth0
1    eth1
2    eth2
Name: Type, dtype: object
```

In [32]:
```
import requests
```

In [33]:
```
def download_url(url):
    r=requests.get(url)
    if(r.status_code != 200):
        print('url download is failed')
        return False
    if('application/json' in r.headers['Content-Type']):
        jd = r.text
        python_data = json.loads(jd) # convert to python
        return python_data
```

In [34]:
```
pd.DataFrame(download_url('https://api.github.com/users/hadley/orgs'))
```

Out[34]:

	login	id	node_id	url	
0	ggobi	423638	MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA==	https://api.github.com/orgs/ggobi	http
1	rstudio	513560	MDEyOk9yZ2FuaXphdGlvbjUxMzU2MA==	https://api.github.com/orgs/rstudio	https
2	rstats	722735	MDEyOk9yZ2FuaXphdGlvbjcyMjczNQ==	https://api.github.com/orgs/rstats	http
3	ropensci	1200269	MDEyOk9yZ2FuaXphdGlvbjEyMDAyNjk=	https://api.github.com/orgs/ropensci	https://
4	rjournal	3330561	MDEyOk9yZ2FuaXphdGlvbjMzMzA1NjE=	https://api.github.com/orgs/rjournal	https:/
5	r-dbi	5695665	MDEyOk9yZ2FuaXphdGlvbjU2OTU2NjU=	https://api.github.com/orgs/r-dbi	htt
6	RConsortium	15366137	MDEyOk9yZ2FuaXphdGlvbjE1MzY2MTM3	https://api.github.com/orgs/RConsortium	https://api.g
7	tidyverse	22032646	MDEyOk9yZ2FuaXphdGlvbjIyMDMyNjQ2	https://api.github.com/orgs/tidyverse	https://a
8	r-lib	22618716	MDEyOk9yZ2FuaXphdGlvbjIyNjE4NzE2	https://api.github.com/orgs/r-lib	htt
9	rstudio-education	34165516	MDEyOk9yZ2FuaXphdGlvbjM0MTY1NTE2	https://api.github.com/orgs/rstudio-education	

In [35]:
```
df4 = pd.DataFrame(download_url('https://api.github.com/users/hadley/orgs'))
```

In [37]:
```
df4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   login               10 non-null     object
 1   id                  10 non-null     int64 
 2   node_id             10 non-null     object
 3   url                 10 non-null     object
 4   repos_url           10 non-null     object
 5   events_url          10 non-null     object
 6   hooks_url           10 non-null     object
 7   issues_url          10 non-null     object
 8   members_url         10 non-null     object
 9   public_members_url  10 non-null     object
 10  avatar_url          10 non-null     object
 11  description         8 non-null      object
dtypes: int64(1), object(11)
memory usage: 1.1+ KB
```

In [38]:
```
df4.columns
```

Out[38]:
```
Index(['login', 'id', 'node_id', 'url', 'repos_url', 'events_url', 'hooks_url',
       'issues_url', 'members_url', 'public_members_url', 'avatar_url',
       'description'],
      dtype='object')
```

In [39]:
```
df4.shape
```

Out[39]:
```
(10, 12)
```

In [41]:
```
# df4.drop('node_id',axis=1) <== to remove single column
#
#    Vs
df4.drop(['repos_url','events_url','issues_url','members_url','description'],axis=1)
```

Out[41]:

	login	id	node_id	url	
0	ggobi	423638	MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA==	https://api.github.com/orgs/ggobi	http
1	rstudio	513560	MDEyOk9yZ2FuaXphdGlvbjUxMzU2MA==	https://api.github.com/orgs/rstudio	https
2	rstats	722735	MDEyOk9yZ2FuaXphdGlvbjcyMjczNQ==	https://api.github.com/orgs/rstats	http
3	ropensci	1200269	MDEyOk9yZ2FuaXphdGlvbjEyMDAyNjk=	https://api.github.com/orgs/ropensci	https://
4	rjournal	3330561	MDEyOk9yZ2FuaXphdGlvbjMzMzA1NjE=	https://api.github.com/orgs/rjournal	https:/
5	r-dbi	5695665	MDEyOk9yZ2FuaXphdGlvbjU2OTU2NjU=	https://api.github.com/orgs/r-dbi	htt
6	RConsortium	15366137	MDEyOk9yZ2FuaXphdGlvbjE1MzY2MTM3	https://api.github.com/orgs/RConsortium	https://api.g
7	tidyverse	22032646	MDEyOk9yZ2FuaXphdGlvbjIyMDMyNjQ2	https://api.github.com/orgs/tidyverse	https://a
8	r-lib	22618716	MDEyOk9yZ2FuaXphdGlvbjIyNjE4NzE2	https://api.github.com/orgs/r-lib	htt
9	rstudio-education	34165516	MDEyOk9yZ2FuaXphdGlvbjM0MTY1NTE2	https://api.github.com/orgs/rstudio-education	https://api.g

In [42]:
```
# df4.drop('node_id',axis=1) <== to remove single column
#
#    Vs
df4.drop(['repos_url','events_url','issues_url','members_url','description'],
         axis=1,inplace=True)
```

In [43]:
```
df4.shape
```

Out[43]:
```
(10, 7)
```

In [48]:
```
df4.drop(index=[3,6,8,9],axis=0,inplace=True)
```

In [49]:
```
df4.shape
```

Out[49]:
```
(6, 7)
```

In [50]:
```
df4
```

Out[50]:

	login	id	node_id	url	
0	ggobi	423638	MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA==	https://api.github.com/orgs/ggobi	https://api.github
1	rstudio	513560	MDEyOk9yZ2FuaXphdGlvbjUxMzU2MA==	https://api.github.com/orgs/rstudio	https://api.github.c
2	rstats	722735	MDEyOk9yZ2FuaXphdGlvbjcyMjczNQ==	https://api.github.com/orgs/rstats	https://api.github
4	rjournal	3330561	MDEyOk9yZ2FuaXphdGlvbjMzMzA1NjE=	https://api.github.com/orgs/rjournal	https://api.github.c
5	r-dbi	5695665	MDEyOk9yZ2FuaXphdGlvbjU2OTU2NjU=	https://api.github.com/orgs/r-dbi	https://api.githul
7	tidyverse	22032646	MDEyOk9yZ2FuaXphdGlvbjIyMDMyNjQ2	https://api.github.com/orgs/tidyverse	https://api.github.cor

In [52]:
```
# dataframe ->write to file <or> convert to another format
#                   pd.to_
#                   pd.read_
#
import sys
```

In [53]:
```
df4.to_csv(sys.stdout)
```

```
,login,id,node_id,url,hooks_url,public_members_url,avatar_url
0,ggobi,423638,MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA==,https://api.github.com/orgs/ggobi,https://
api.github.com/orgs/ggobi/hooks,https://api.github.com/orgs/ggobi/public_members{/member},
https://avatars.githubusercontent.com/u/423638?v=4
1,rstudio,513560,MDEyOk9yZ2FuaXphdGlvbjUxMzU2MA==,https://api.github.com/orgs/rstudio,http
s://api.github.com/orgs/rstudio/hooks,https://api.github.com/orgs/rstudio/public_members{/
member},https://avatars.githubusercontent.com/u/513560?v=4
2,rstats,722735,MDEyOk9yZ2FuaXphdGlvbjcyMjczNQ==,https://api.github.com/orgs/rstats,http
s://api.github.com/orgs/rstats/hooks,https://api.github.com/orgs/rstats/public_members{/me
mber},https://avatars.githubusercontent.com/u/722735?v=4
4,rjournal,3330561,MDEyOk9yZ2FuaXphdGlvbjMzMzA1NjE=,https://api.github.com/orgs/rjournal,h
ttps://api.github.com/orgs/rjournal/hooks,https://api.github.com/orgs/rjournal/public_memb
ers{/member},https://avatars.githubusercontent.com/u/3330561?v=4
5,r-dbi,5695665,MDEyOk9yZ2FuaXphdGlvbjU2OTU2NjU=,https://api.github.com/orgs/r-dbi,http
s://api.github.com/orgs/r-dbi/hooks,https://api.github.com/orgs/r-dbi/public_members{/memb
er},https://avatars.githubusercontent.com/u/5695665?v=4
7,tidyverse,22032646,MDEyOk9yZ2FuaXphdGlvbjIyMDMyNjQ2,https://api.github.com/orgs/tidyvers
e,https://api.github.com/orgs/tidyverse/hooks,https://api.github.com/orgs/tidyverse/public
_members{/member},https://avatars.githubusercontent.com/u/22032646?v=4
```

In [54]:
```
df3
```

Out[54]:

	Type	IP	port	app
0	eth0	10.20.30.40	3000	gl
1	eth1	11.23.45.45	4000	fl
2	eth2	10.20.34.55	5000	pm

In [55]:
```
df3.to_csv(sys.stdout)
```

```
,Type,IP,port,app
0,eth0,10.20.30.40,3000,gl
1,eth1,11.23.45.45,4000,fl
2,eth2,10.20.34.55,5000,pm
```

In [56]:
```
df4.to_csv('E:\\test1.csv')
```

In [57]:
```
pd.read_csv('E:\\test1.csv')
```

Out[57]:

	Unnamed: 0	login	id	node_id	url	
0	0	ggobi	423638	MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA==	https://api.github.com/orgs/ggobi	http
1	1	rstudio	513560	MDEyOk9yZ2FuaXphdGlvbjUxMzU2MA==	https://api.github.com/orgs/rstudio	https
2	2	rstats	722735	MDEyOk9yZ2FuaXphdGlvbjcyMjczNQ==	https://api.github.com/orgs/rstats	http
3	4	rjournal	3330561	MDEyOk9yZ2FuaXphdGlvbjMzMzA1NjE=	https://api.github.com/orgs/rjournal	https:
4	5	r-dbi	5695665	MDEyOk9yZ2FuaXphdGlvbjU2OTU2NjU=	https://api.github.com/orgs/r-dbi	htt
5	7	tidyverse	22032646	MDEyOk9yZ2FuaXphdGlvbjIyMDMyNjQ2	https://api.github.com/orgs/tidyverse	https://

In [58]:
```
df5= pd.read_csv('E:\\test1.csv')
df5.columns
```

Out[58]:
```
Index(['Unnamed: 0', 'login', 'id', 'node_id', 'url', 'hooks_url',
       'public_members_url', 'avatar_url'],
      dtype='object')
```

In [59]:
```
df5.drop(columns=['Unnamed: 0','hooks_url'],axis=1,inplace=True)
```

In [60]:
```
df5.columns
```

Out[60]:
```
Index(['login', 'id', 'node_id', 'url', 'public_members_url', 'avatar_url'], dtype='objec
t')
```

In [62]:
```
df5.rename({'public_members_url':'public'},axis=1)
```

Out[62]:

	node_id	url	publi
MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA==	https://api.github.com/orgs/ggobi	https://api.github.com/orgs/ggobi/public_membe.
MDEyOk9yZ2FuaXphdGlvbjUxMzU2MA==	https://api.github.com/orgs/rstudio	https://api.github.com/orgs/rstudio/public_mem.
MDEyOk9yZ2FuaXphdGlvbjcyMjczNQ==	https://api.github.com/orgs/rstats	https://api.github.com/orgs/rstats/public_memb.
MDEyOk9yZ2FuaXphdGlvbjMzMzA1NjE=	https://api.github.com/orgs/rjournal	https://api.github.com/orgs/rjournal/public_me.
MDEyOk9yZ2FuaXphdGlvbjU2OTU2NjU=	https://api.github.com/orgs/r-dbi	https://api.github.com/orgs/r-dbi/public_membe.
MDEyOk9yZ2FuaXphdGlvbjIyMDMyNjQ2	https://api.github.com/orgs/tidyverse	https://api.github.com/orgs/tidyverse/public_m.

In [63]:
```
df5.rename({'public_members_url':'public'},axis=1,inplace=True)
```

In [65]:
```
df5.columns
```

Out[65]:
```
Index(['login', 'id', 'node_id', 'url', 'public', 'avatar_url'], dtype='object')
```

In [66]:
```
df5.to_csv(sys.stdout)
```

```
,login,id,node_id,url,public,avatar_url
0,ggobi,423638,MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA==,https://api.github.com/orgs/ggobi,https://
api.github.com/orgs/ggobi/public_members{/member},https://avatars.githubusercontent.com/u/
423638?v=4
1,rstudio,513560,MDEyOk9yZ2FuaXphdGlvbjUxMzU2MA==,https://api.github.com/orgs/rstudio,http
s://api.github.com/orgs/rstudio/public_members{/member},https://avatars.githubusercontent.
com/u/513560?v=4
2,rstats,722735,MDEyOk9yZ2FuaXphdGlvbjcyMjczNQ==,https://api.github.com/orgs/rstats,http
s://api.github.com/orgs/rstats/public_members{/member},https://avatars.githubusercontent.c
om/u/722735?v=4
3,rjournal,3330561,MDEyOk9yZ2FuaXphdGlvbjMzMzA1NjE=,https://api.github.com/orgs/rjournal,h
ttps://api.github.com/orgs/rjournal/public_members{/member},https://avatars.githubusercont
ent.com/u/3330561?v=4
4,r-dbi,5695665,MDEyOk9yZ2FuaXphdGlvbjU2OTU2NjU=,https://api.github.com/orgs/r-dbi,http
s://api.github.com/orgs/r-dbi/public_members{/member},https://avatars.githubusercontent.co
m/u/5695665?v=4
5,tidyverse,22032646,MDEyOk9yZ2FuaXphdGlvbjIyMDMyNjQ2,https://api.github.com/orgs/tidyvers
e,https://api.github.com/orgs/tidyverse/public_members{/member},https://avatars.githubuser
content.com/u/22032646?v=4
```

In [67]:
```
df5.to_csv(sys.stdout,sep="|") # Vs pd.read_csv('filename',sep='|')
```

```
|login|id|node_id|url|public|avatar_url
0|ggobi|423638|MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA==|https://api.github.com/orgs/ggobi|https://
api.github.com/orgs/ggobi/public_members{/member}|https://avatars.githubusercontent.com/u/
423638?v=4
1|rstudio|513560|MDEyOk9yZ2FuaXphdGlvbjUxMzU2MA==|https://api.github.com/orgs/rstudio|http
s://api.github.com/orgs/rstudio/public_members{/member}|https://avatars.githubusercontent.
com/u/513560?v=4
2|rstats|722735|MDEyOk9yZ2FuaXphdGlvbjcyMjczNQ==|https://api.github.com/orgs/rstats|http
s://api.github.com/orgs/rstats/public_members{/member}|https://avatars.githubusercontent.c
om/u/722735?v=4
3|rjournal|3330561|MDEyOk9yZ2FuaXphdGlvbjMzMzA1NjE=|https://api.github.com/orgs/rjournal|h
ttps://api.github.com/orgs/rjournal/public_members{/member}|https://avatars.githubusercont
ent.com/u/3330561?v=4
4|r-dbi|5695665|MDEyOk9yZ2FuaXphdGlvbjU2OTU2NjU=|https://api.github.com/orgs/r-dbi|http
s://api.github.com/orgs/r-dbi/public_members{/member}|https://avatars.githubusercontent.co
m/u/5695665?v=4
5|tidyverse|22032646|MDEyOk9yZ2FuaXphdGlvbjIyMDMyNjQ2|https://api.github.com/orgs/tidyvers
e|https://api.github.com/orgs/tidyverse/public_members{/member}|https://avatars.githubuser
content.com/u/22032646?v=4
```

In [68]:
```
df5.to_csv(sys.stdout,columns=['node_id','url'])
```

```
,node_id,url
0,MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA==,https://api.github.com/orgs/ggobi
1,MDEyOk9yZ2FuaXphdGlvbjUxMzU2MA==,https://api.github.com/orgs/rstudio
2,MDEyOk9yZ2FuaXphdGlvbjcyMjczNQ==,https://api.github.com/orgs/rstats
3,MDEyOk9yZ2FuaXphdGlvbjMzMzA1NjE=,https://api.github.com/orgs/rjournal
4,MDEyOk9yZ2FuaXphdGlvbjU2OTU2NjU=,https://api.github.com/orgs/r-dbi
5,MDEyOk9yZ2FuaXphdGlvbjIyMDMyNjQ2,https://api.github.com/orgs/tidyverse
```

In [69]:
```
pd.read_excel('E:\\course.xlsx')
```

Out[69]:

	Course	Fees	Duration	Tutor
0	C	25000	24	Mr.Tom
1	C++	26000	22	Ms.Anu
2	Java	30000	30	Mr.Shiv
3	html	5000	3	Ms.Theeb
4	python	15000	10	Mr.Raj
5	ruby	13000	20	Mr.Tom
6	ML	23000	30	Ms.Anu
7	Mysql	10000	12	Mr.Shiv
8	Oracle	9000	10	Mr.Leo
9	DA	12300	15	Ms.Anj
10	AI	34000	30	Ms.Anu

In [72]:
```
df5.to_excel('E:\\test1.xlsx')
```

In [74]:
```
#df5.to_excel('E:\\test3.xls') ->Error
```

In [75]:
```
pd.read_excel('E:\\course.xlsx')
```

Out[75]:

	Course	Fees	Duration	Tutor
0	C	25000	24	Mr.Tom
1	C++	26000	22	Ms.Anu
2	Java	30000	30	Mr.Shiv
3	html	5000	3	Ms.Theeb
4	python	15000	10	Mr.Raj
5	ruby	13000	20	Mr.Tom
6	ML	23000	30	Ms.Anu
7	Mysql	10000	12	Mr.Shiv
8	Oracle	9000	10	Mr.Leo
9	DA	12300	15	Ms.Anj
10	AI	34000	30	Ms.Anu

In [76]:
```
pd.read_excel('E:\\course.xlsx')
```

Out[76]:

	Course	Fees	Duration	Tutor
0	C	25000	24	Mr.Tom
1	C++	26000	22	Ms.Anu
2	Java	30000	30	Mr.Shiv
3	html	5000	3	Ms.Theeb
4	python	15000	10	Mr.Raj
5	ruby	13000	20	Mr.Tom
6	ML	23000	30	Ms.Anu
7	Mysql	10000	12	Mr.Shiv
8	Oracle	9000	10	Mr.Leo
9	DA	12300	15	Ms.Anj
10	AI	34000	30	Ms.Anu

In [77]:
```
pd.read_excel('E:\\course.xlsx',sheet_name='products')
```

Out[77]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2
0	NaN	product	price
1	NaN	prodA	1000
2	NaN	prodB	2000
3	NaN	prodC	3000
4	NaN	prodD	3000
5	NaN	prodA	1234
6	NaN	prodB	6000
7	NaN	prodE	5412
8	NaN	prodB	8000

In [78]:
```
pd.read_excel('E:\\course.xlsx',sheet_name=['One','products'])
```

Out[78]:
```
{'One':     Course   Fees  Duration     Tutor
 0         C  25000        24    Mr.Tom
 1       C++  26000        22    Ms.Anu
 2      Java  30000        30   Mr.Shiv
 3      html   5000         3  Ms.Theeb
 4    python  15000        10    Mr.Raj
 5      ruby  13000        20    Mr.Tom
 6        ML  23000        30    Ms.Anu
 7     Mysql  10000        12   Mr.Shiv
 8    Oracle   9000        10    Mr.Leo
 9        DA  12300        15    Ms.Anj
 10       AI  34000        30    Ms.Anu,
 'products':    Unnamed: 0 Unnamed: 1 Unnamed: 2
 0          NaN    product      price
 1          NaN      prodA       1000
 2          NaN      prodB       2000
 3          NaN      prodC       3000
 4          NaN      prodD       3000
 5          NaN      prodA       1234
 6          NaN      prodB       6000
 7          NaN      prodE       5412
 8          NaN      prodB       8000}
```

In [79]:
```
df6 = pd.read_excel('E:\\course.xlsx',sheet_name=['One','products'])
df6['One']
```

Out[79]:

	Course	Fees	Duration	Tutor
0	C	25000	24	Mr.Tom
1	C++	26000	22	Ms.Anu
2	Java	30000	30	Mr.Shiv
3	html	5000	3	Ms.Theeb
4	python	15000	10	Mr.Raj
5	ruby	13000	20	Mr.Tom
6	ML	23000	30	Ms.Anu
7	Mysql	10000	12	Mr.Shiv
8	Oracle	9000	10	Mr.Leo
9	DA	12300	15	Ms.Anj
10	AI	34000	30	Ms.Anu

In [80]:
```
df6['products']
```

Out[80]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2
0	NaN	product	price
1	NaN	prodA	1000
2	NaN	prodB	2000
3	NaN	prodC	3000
4	NaN	prodD	3000
5	NaN	prodA	1234
6	NaN	prodB	6000
7	NaN	prodE	5412
8	NaN	prodB	8000

In [86]:
```
#excel_df = pd.read_excel('E:\\test1.xlsx')
#           Vs
xl_df = pd.ExcelFile('E:\\course.xlsx')
xl_df.sheet_names # To get list of sheets from excel file
```

Out[86]:
```
['One', 'products']
```

In [87]:
```
pd.read_excel(xl_df,'One') # read a particular sheet name
```

Out[87]:

	Course	Fees	Duration	Tutor
0	C	25000	24	Mr.Tom
1	C++	26000	22	Ms.Anu
2	Java	30000	30	Mr.Shiv
3	html	5000	3	Ms.Theeb
4	python	15000	10	Mr.Raj
5	ruby	13000	20	Mr.Tom
6	ML	23000	30	Ms.Anu
7	Mysql	10000	12	Mr.Shiv
8	Oracle	9000	10	Mr.Leo
9	DA	12300	15	Ms.Anj
10	AI	34000	30	Ms.Anu

In [88]:
```
import sqlite3
```

In [89]:
```
sqlite3.connect('mydb\products.db')
```

Out[89]:
```
<sqlite3.Connection at 0x1c2e70fbc40>
```

In [90]:
```
conn = sqlite3.connect('mydb\products.db')
pd.read_sql_query('select *from prod',conn)
```

Out[90]:

	id	pname	pcost
0	101	prodA	1000
1	102	prodB	2000
2	103	prodC	3000
3	104	prodD	4000
4	105	prodE	5000

In [91]:
```
df7 = pd.read_sql_query('select *from prod',conn)
df7.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   id      5 non-null      int64
 1   pname   5 non-null      object
 2   pcost   5 non-null      int64
dtypes: int64(2), object(1)
memory usage: 252.0+ bytes
```

In [92]:
```
df7.to_excel('E:\\mydb.xlsx') # write to Excel file
```

In [94]:
```
# To access word document
#
# pip instll python-docx

!pip install python-docx
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting python-docx
  Obtaining dependency information for python-docx from https://files.pythonhosted.org/packages/3e/3d/330d9efbdb816d3f60bf2ad92f05e1708e4a1b9abe80461ac3444c83f749/python_docx-1.1.2-py3-none-any.whl.metadata (https://files.pythonhosted.org/packages/3e/3d/330d9efbdb816d3f60bf2ad92f05e1708e4a1b9abe80461ac3444c83f749/python_docx-1.1.2-py3-none-any.whl.metadata)
  Downloading python_docx-1.1.2-py3-none-any.whl.metadata (2.0 kB)
Requirement already satisfied: lxml>=3.1.0 in c:\programdata\anaconda3\lib\site-packages
(from python-docx) (4.9.3)
Requirement already satisfied: typing-extensions>=4.9.0 in c:\users\raja\appdata\roaming\p
ython\python311\site-packages (from python-docx) (4.12.2)
Downloading python_docx-1.1.2-py3-none-any.whl (244 kB)
   ---------------------------------------- 0.0/244.3 kB ? eta -:--:--
   - -------------------------------------- 10.2/244.3 kB ? eta -:--:--
   ------ --------------------------------- 41.0/244.3 kB 653.6 kB/s eta 0:00:01
   ----------------- ---------------------- 112.6/244.3 kB 1.1 MB/s eta 0:00:01
   --------------------------- ------------ 174.1/244.3 kB 1.2 MB/s eta 0:00:01
   ---------------------------------------  235.5/244.3 kB 1.2 MB/s eta 0:00:01
   ---------------------------------------- 244.3/244.3 kB 1.2 MB/s eta 0:00:00
Installing collected packages: python-docx
Successfully installed python-docx-1.1.2

DEPRECATION: Loading egg at c:\programdata\anaconda3\lib\site-packages\vboxapi-1.0-py3.11.
egg is deprecated. pip 23.3 will enforce this behaviour change. A possible replacement is
to use pip for package installation..
```

In [95]:
```
import docx
```

In [96]:
```
print(docx.Document)
```

```
<function Document at 0x000001C2E57B1F80>
```

In [97]:
```
docx.Document('E:\\Activity-1.docx')
```

Out[97]:
```
<docx.document.Document at 0x1c2e7db3e10>
```

In [98]:
```
doc_obj = docx.Document('E:\\Activity-1.docx')
doc_obj.paragraphs
```

Out[98]:
```
[<docx.text.paragraph.Paragraph at 0x1c2e572f0d0>,
 <docx.text.paragraph.Paragraph at 0x1c2e74f23d0>,
 <docx.text.paragraph.Paragraph at 0x1c2e74f0850>,
 <docx.text.paragraph.Paragraph at 0x1c2e74f2590>,
 <docx.text.paragraph.Paragraph at 0x1c2e74f25d0>,
 <docx.text.paragraph.Paragraph at 0x1c2e74f2350>,
 <docx.text.paragraph.Paragraph at 0x1c2e74e9e90>,
 <docx.text.paragraph.Paragraph at 0x1c2e74e8890>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604bd0>,
 <docx.text.paragraph.Paragraph at 0x1c2e74f2650>,
 <docx.text.paragraph.Paragraph at 0x1c2e7605050>,
 <docx.text.paragraph.Paragraph at 0x1c2e7606f90>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604210>,
 <docx.text.paragraph.Paragraph at 0x1c2e7606cd0>,
 <docx.text.paragraph.Paragraph at 0x1c2e7607350>,
 <docx.text.paragraph.Paragraph at 0x1c2e7606810>,
 <docx.text.paragraph.Paragraph at 0x1c2e7606f10>,
 <docx.text.paragraph.Paragraph at 0x1c2e7607590>,
 <docx.text.paragraph.Paragraph at 0x1c2e7607710>,
 <docx.text.paragraph.Paragraph at 0x1c2e76061d0>,
 <docx.text.paragraph.Paragraph at 0x1c2e7605850>,
 <docx.text.paragraph.Paragraph at 0x1c2e7605f10>,
 <docx.text.paragraph.Paragraph at 0x1c2e76064d0>,
 <docx.text.paragraph.Paragraph at 0x1c2e7607c10>,
 <docx.text.paragraph.Paragraph at 0x1c2e7607e10>,
 <docx.text.paragraph.Paragraph at 0x1c2e76040d0>,
 <docx.text.paragraph.Paragraph at 0x1c2e7605390>,
 <docx.text.paragraph.Paragraph at 0x1c2e7607250>,
 <docx.text.paragraph.Paragraph at 0x1c2e7607d50>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604190>,
 <docx.text.paragraph.Paragraph at 0x1c2e76041d0>,
 <docx.text.paragraph.Paragraph at 0x1c2e7606fd0>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604050>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604550>,
 <docx.text.paragraph.Paragraph at 0x1c2e7606b10>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604250>,
 <docx.text.paragraph.Paragraph at 0x1c2e7606e90>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604090>,
 <docx.text.paragraph.Paragraph at 0x1c2e7607950>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604610>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604310>,
 <docx.text.paragraph.Paragraph at 0x1c2e7605410>,
 <docx.text.paragraph.Paragraph at 0x1c2e7607c50>,
 <docx.text.paragraph.Paragraph at 0x1c2e76044d0>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604710>,
 <docx.text.paragraph.Paragraph at 0x1c2e7605350>,
 <docx.text.paragraph.Paragraph at 0x1c2e7605590>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604e90>,
 <docx.text.paragraph.Paragraph at 0x1c2e76055d0>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604ed0>,
 <docx.text.paragraph.Paragraph at 0x1c2e7605610>,
 <docx.text.paragraph.Paragraph at 0x1c2e7606590>,
 <docx.text.paragraph.Paragraph at 0x1c2e7607b50>,
 <docx.text.paragraph.Paragraph at 0x1c2e7605650>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604f10>,
 <docx.text.paragraph.Paragraph at 0x1c2e7605690>,
 <docx.text.paragraph.Paragraph at 0x1c2e7604f50>,
 <docx.text.paragraph.Paragraph at 0x1c2e76056d0>]
```

In [103]:
```
doc_obj = docx.Document('E:\\Activity-1.docx')

for var in doc_obj.paragraphs:
    print(var.text)
```

Q1. Write a C++ program to check overflow/underflow during various arithmetic operations.
Sample Output:
Check overflow/underflow during various arithmetical operation:

```
Range of int is [-2147483648, 2147483647]
Overflow the integer range and set in minimum range: -2147483648
Increasing from its minimum range: -2147483647
Product is :1
Underflow the range and set in maximum range: 2147483647
Decreasing from its maximum range: 2147483646
Product is: 0
```

Q2. Write a program in C++ that converts kilometers per hour to miles per hour.
Sample Output:
Convert kilometers per hour to miles per hour :

```
--------------------------------------------------
Input the distance in kilometer : 25
The 25 Km./hr. means 15.5343 Miles/hr.
```

Q3. Write a program in C++ to convert temperature in Kelvin to Fahrenheit.

Q4. Write C++ Program do following task
Largest of 3 nos using conditional op
Largest of 2 nos without branching
To check for equality of two numbers without using arithmetic or comparison operator
Print “Welcome” without using semicolon in c/c++
To check if the given number is even without using arithmetic or relational operators.
Print grade using switch case

Q5. Write C++ Program do following task
Convert Hex number to binary
Quadratic equation solving
Hexadecimal to decimal
Decimal to hexa
Find square root of a number without using sqrt fun
print series using structure

```
1           s

2           s           t

3           s           i

4           s           t

5           s

6           s           t           i

7           s
```

```
8            s            t

9            s            i

10          s            t

Print numbers from 1 to 10 without using loops or goto statements
```

In [104]:
```
doc_obj = docx.Document('E:\\Activity-1.docx')

for var in doc_obj.paragraphs:
    if 'range' in var.text:
        print(var.text)
```

```
Overflow the integer range and set in minimum range: -2147483648
Increasing from its minimum range: -2147483647
Product is :1
Underflow the range and set in maximum range: 2147483647
Decreasing from its maximum range: 2147483646
Product is: 0
```

In [105]:
```
L=[]
doc_obj = docx.Document('E:\\Activity-1.docx')

for var in doc_obj.paragraphs:
    if 'range' in var.text:
        L.append(var.text)
```

In [108]:
```
d={}
d['range']=L
d
```

Out[108]:
```
{'range': ['Overflow the integer range and set in minimum range: -2147483648\nIncreasing f
rom its minimum range: -2147483647\nProduct is :1\nUnderflow the range and set in maximum
range: 2147483647\nDecreasing from its maximum range: 2147483646\nProduct is: 0']}
```

In [109]:
```
pd.DataFrame(d)
```

Out[109]:

	range
0	Overflow the integer range and set in minimum ...

In [111]:
```
pd.DataFrame(d).info()
pd.DataFrame(d).shape
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1 entries, 0 to 0
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   range   1 non-null      object
dtypes: object(1)
memory usage: 140.0+ bytes
```

Out[111]:
```
(1, 1)
```

In [117]:
```
doc_obj = docx.Document('E:\\Test Cases.docx')
```

In [118]:
```
for var in doc_obj.paragraphs:
    print(var.text)
```

```
The test environment has been setup in conformity with the requirements for the infrastr
ucture, functionality and test data sets, and the intake of this environment has been su
ccessfully concluded
There are no functional defects outstanding from the FAT that obstructs the execution of
the UAT. These are defects that have impact on <name system> to such an extent that the
operational line, and functional administrative processes, the technical administrative
processes and the technical quality aspects can’t be verified
The version of <name system> that is suitable for the UAT has been delivered, migrated a
nd setup in the acceptance environment and the intake of this version have been successf
ully concluded >>

The following exit criteria have been defined for the FAT:
<< For example:
Test results have been delivered by means of a test report and are approved by the accep
tant
There are no functional defects outstanding that obstruct the execution of the PAT. Thes
e are defects that have impact on <name system> to such an extent that the operational l
ine, and functional administrative processes, the technical administrative processes and
the technical quality aspects can’t be verified
There are no functional defects outstanding that bring unacceptable business risks when
```

In [132]:
```
doc_obj.core_properties.keywords
```

Out[132]:
```
'v2.00'
```

In [136]:
```
doc_obj.core_properties.comments
```

Out[136]:
```
''
```

In [138]:
```
import re
```

In [151]:
```
doc_obj = docx.Document('E:\\Test Cases.docx')

test_scenarios_start = None
test_case_info = []
current_scenarios = {}
for var in doc_obj.paragraphs:
    if re.search('4.4.*Test Scenarios',var.text):
        test_scenarios_start = True
        continue
    # after finiding this pattern we begin extract test case details
    if test_scenarios_start:
        if var.text.strip() == '':
            if current_scenarios:
                test_case_info.appen(current_scenarios)
                current_scenarios = {} # reset for next test case
        else:
            if re.search('Test Flow 1.*',var.text):
                print(var.text.strip())
```

```
4.4.1   Test Flow 1     8
Test Flow 1
```

In [155]:
```
xl_obj = pd.ExcelFile('E:\\testcase.xlsx')
sheet_name = xl_obj.sheet_names
xl_df = pd.read_excel(xl_obj,sheet_name)
xl_df['Sheet1']
```

Out[155]:

	SIno	Navigation	Action	Data	Expected Results	Comments
0	4.4.1.1.1.	Login: mfg/welcome	NaN	NaN	NaN	NaN
1	NaN	Resp: Manufacturing and Distribution Manager	NaN	NaN	NaN	NaN
2	NaN	Inventory->Kanban->Pull Sequences	NaN	NaN	Find Pull Sequence form opens	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	Action: Click “New”	Enter Data	Pull Sequence summary form opens.	NaN
5	NaN	NaN	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	In Supplier Tab:	NaN	NaN
7	NaN	NaN	NaN	Item: XXX100	NaN	NaN
8	NaN	NaN	NaN	Subinventory: XXXSUB1	NaN	NaN
9	NaN	NaN	NaN	Source Type: Supplier	NaN	NaN
10	NaN	NaN	NaN	Supplier: Allied Manufacturing	NaN	NaN
11	NaN	NaN	NaN	NaN	NaN	NaN
12	NaN	NaN	NaN	In Kanban Tab:	NaN	NaN
13	NaN	NaN	NaN	Calculate: Number of Cards	NaN	NaN
14	NaN	NaN	NaN	Size: 100	NaN	NaN
15	NaN	NaN	NaN	Number of Cards: 5	NaN	NaN

In [167]:
```
d={'K1':[10,20,20],'K2':[20,20,20],'K3':[20,20,20]}
pd.DataFrame(d)
```

Out[167]:

	K1	K2	K3
0	10	20	20
1	20	20	20
2	20	20	20

In [168]:
```
df8 = pd.DataFrame(d)
df8.duplicated() # Test any duplicate details is exists
```

Out[168]:
```
0    False
1    False
2     True
dtype: bool
```

In [169]:
```
df8.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   K1      3 non-null      int64
 1   K2      3 non-null      int64
 2   K3      3 non-null      int64
dtypes: int64(3)
memory usage: 204.0 bytes
```

In [170]:
```
df8.duplicated().sum() # Count no.of duplicate items
```

Out[170]: 1

In [171]:
```
df8.duplicated().any() # returns bool ->True <== there is duplicate items exists
```

Out[171]: True

In [172]:
```
df8.drop_duplicates() # drop duplicate items
```

Out[172]:

	K1	K2	K3
0	10	20	20
1	20	20	20

In [173]:
```
df8
```

Out[173]:

	K1	K2	K3
0	10	20	20
1	20	20	20
2	20	20	20

In [174]:
```
df8.drop_duplicates(inplace=True)
```

In [175]:
```
df8
```

Out[175]:

	K1	K2	K3
0	10	20	20
1	20	20	20

In [176]:
```
df8.duplicated().sum() # ->0 There is no duplicate item
```

Out[176]: 0

In [177]:
```
df8.duplicated().any() # ->False - There is no duplicate item
```

Out[177]: False

In [178]:
```
xl_df['Sheet1'].duplicated().any()
```

Out[178]:
```
True
```

In [179]:
```
xl_df['Sheet1'].duplicated().sum()
```

Out[179]:
```
2
```

In [180]:
```
xl_df['Sheet1'].duplicated()
```

Out[180]:
```
0     False
1     False
2     False
3     False
4     False
5      True
6     False
7     False
8     False
9     False
10    False
11     True
12    False
13    False
14    False
15    False
dtype: bool
```

In [184]:
```
xl_df['Sheet1'].drop_duplicates(inplace=True)
```

In [185]:
```
xl_df['Sheet1'].duplicated().sum()
```

Out[185]:
```
0
```

In [191]:
```
import numpy as np
d={'K1':[10,np.nan,30],'K2':[20,np.nan,np.nan],'K3':[np.nan,None,np.nan]}
d['K4']=[1,2,3]
pd.DataFrame(d)
```

Out[191]:

	K1	K2	K3	K4
0	10.0	20.0	NaN	1
1	NaN	NaN	NaN	2
2	30.0	NaN	NaN	3

In [192]:
```
df9 = pd.DataFrame(d)

# To determine any null(NaN) value is present
df9.isnull() # ->True - NaN value is exists
```

Out[192]:

	K1	K2	K3	K4
0	False	False	True	False
1	True	True	True	False
2	False	True	True	False

In [193]:
```
df9.isnull().any() # ->True - NaN value is exists
```

Out[193]:
```
K1     True
K2     True
K3     True
K4    False
dtype: bool
```

In [194]:
```
df9.isnull().sum() # Each Column based - display NaN value count
```

Out[194]:
```
K1    1
K2    2
K3    3
K4    0
dtype: int64
```

In [195]:
```
df9.isna() # same as df9.isnull()
```

Out[195]:

	K1	K2	K3	K4
0	False	False	True	False
1	True	True	True	False
2	False	True	True	False

In [196]:
```
print(df9.isna().sum(),"\n\n",df9.isna().any())
```

```
K1    1
K2    2
K3    3
K4    0
dtype: int64

 K1     True
K2     True
K3     True
K4    False
dtype: bool
```

In [197]:
```
# How to replace NaN value ?
# fillna(value=<replaceValue>)

df9
```

Out[197]:

	K1	K2	K3	K4
0	10.0	20.0	NaN	1
1	NaN	NaN	NaN	2
2	30.0	NaN	NaN	3

In [198]:
```
df9.fillna('DATA')
```

Out[198]:

	K1	K2	K3	K4
0	10.0	20.0	DATA	1
1	DATA	DATA	DATA	2
2	30.0	DATA	DATA	3

In [199]:
```
df9.fillna(56)
```

Out[199]:

	K1	K2	K3	K4
0	10.0	20.0	56.0	1
1	56.0	56.0	56.0	2
2	30.0	56.0	56.0	3

In [200]:
```
df9.fillna(value=56)
```

Out[200]:

	K1	K2	K3	K4
0	10.0	20.0	56.0	1
1	56.0	56.0	56.0	2
2	30.0	56.0	56.0	3

In [201]:
```
df9['K1'].fillna(value=56)
```

Out[201]:
```
0    10.0
1    56.0
2    30.0
Name: K1, dtype: float64
```

In [202]:
```
df9['K1'].fillna(value=56,inplace=True)
```

In [203]:
```
df9
```

Out[203]:

	K1	K2	K3	K4
0	10.0	20.0	NaN	1
1	56.0	NaN	NaN	2
2	30.0	NaN	NaN	3

In [205]:
```
d={'K1':[10,20,30],'K2':[15,None,25],'K3':[100,None,50]}
df10 = pd.DataFrame(d)
df10.isnull().sum()
```

Out[205]:
```
K1    0
K2    1
K3    1
dtype: int64
```

In [206]:
```
df10.fillna(value=55)
```

Out[206]:

	K1	K2	K3
0	10	15.0	100.0
1	20	55.0	55.0
2	30	25.0	50.0

In [207]:
```
np.mean([1,2,3,4,5])
```

Out[207]:
```
3.0
```

In [208]:
```
df10['K2'].fillna(value=np.mean(df10['K2']))
```

Out[208]:
```
0    15.0
1    20.0
2    25.0
Name: K2, dtype: float64
```

In [209]:
```
df10['K2'].fillna(value=np.mean(df10['K2']),inplace=True)
```

In [210]:
```
df10
```

Out[210]:

	K1	K2	K3
0	10	15.0	100.0
1	20	20.0	NaN
2	30	25.0	50.0

In [212]:
```
df10['K3'].fillna(value=np.mean(df10['K3']),inplace=True)
```

In [213]:
```
df10.isnull()
```

Out[213]:

	K1	K2	K3
0	False	False	False
1	False	False	False
2	False	False	False

In [214]:
```
df10.isnull().sum()
```

Out[214]:
```
K1    0
K2    0
K3    0
dtype: int64
```

In [ ]:

In [216]:
```
df6 = pd.read_excel('E:\\mycourse.xlsx',sheet_name=['One','products'])
df6['products']
```

Out[216]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2
0	NaN	product	price
1	NaN	prodA	1000
2	NaN	prodB	NaN
3	NaN	prodC	3000
4	NaN	prodD	3000
5	NaN	prodA	NaN
6	NaN	prodB	6000
7	NaN	prodE	5412
8	NaN	prodB	8000
9	NaN	prodD	3000

In [219]:
```
pd.read_excel('E://mycourse.xlsx',sheet_name='products')
```

Out[219]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2
0	NaN	product	price
1	NaN	prodA	1000
2	NaN	prodB	NaN
3	NaN	prodC	3000
4	NaN	prodD	3000
5	NaN	prodA	NaN
6	NaN	prodB	6000
7	NaN	prodE	5412
8	NaN	prodB	8000
9	NaN	prodD	3000

In [220]:
```
pd.read_excel('E://mycourse.xlsx',sheet_name='products',header=1)
```

Out[220]:

	Unnamed: 0	product	price
0	NaN	prodA	1000.0
1	NaN	prodB	NaN
2	NaN	prodC	3000.0
3	NaN	prodD	3000.0
4	NaN	prodA	NaN
5	NaN	prodB	6000.0
6	NaN	prodE	5412.0
7	NaN	prodB	8000.0
8	NaN	prodD	3000.0

In [221]:
```
df11 = pd.read_excel('E://mycourse.xlsx',sheet_name='products',header=1)
df11
```

Out[221]:

	Unnamed: 0	product	price
0	NaN	prodA	1000.0
1	NaN	prodB	NaN
2	NaN	prodC	3000.0
3	NaN	prodD	3000.0
4	NaN	prodA	NaN
5	NaN	prodB	6000.0
6	NaN	prodE	5412.0
7	NaN	prodB	8000.0
8	NaN	prodD	3000.0

In [222]:
```
df11.duplicated()
```

Out[222]:
```
0    False
1    False
2    False
3    False
4    False
5    False
6    False
7    False
8     True
dtype: bool
```

In [223]:
```
df11.duplicated().sum()
```

Out[223]:
```
1
```

In [224]:
```
df11.duplicated().any()
```

Out[224]:
```
True
```

In [225]:
```
df11.drop_duplicates(inplace=True)
```

In [226]:
```
df11.duplicated().any()
```

Out[226]:
```
False
```

In [227]:
```
df11.duplicated().sum()
```

Out[227]:
```
0
```

In [228]:
```
df11.isnull()
```

Out[228]:

	Unnamed: 0	product	price
0	True	False	False
1	True	False	True
2	True	False	False
3	True	False	False
4	True	False	True
5	True	False	False
6	True	False	False
7	True	False	False

In [229]:
```
df11.isnull().sum()
```

Out[229]:
```
Unnamed: 0    8
product       0
price         2
dtype: int64
```

In [232]:
```
df11.rename(columns={'Unnamed: 0':'Key1'},inplace=True)
```

In [233]:
```
df11.columns
```

Out[233]:
```
Index(['Key1', 'product', 'price'], dtype='object')
```

In [234]:
```
df11.isnull().sum()
```

Out[234]:
```
Key1       8
product    0
price      2
dtype: int64
```

In [235]:
```
df11
```

Out[235]:

	Key1	product	price
0	NaN	prodA	1000.0
1	NaN	prodB	NaN
2	NaN	prodC	3000.0
3	NaN	prodD	3000.0
4	NaN	prodA	NaN
5	NaN	prodB	6000.0
6	NaN	prodE	5412.0
7	NaN	prodB	8000.0

In [237]:
```
df11['Key1'].fillna(value='abc',inplace=True)
df11['price'].fillna(value=np.mean(df11['price']),inplace=True)
```

In [238]:
```
df11
```

Out[238]:

	Key1	product	price
0	abc	prodA	1000.0
1	abc	prodB	4402.0
2	abc	prodC	3000.0
3	abc	prodD	3000.0
4	abc	prodA	4402.0
5	abc	prodB	6000.0
6	abc	prodE	5412.0
7	abc	prodB	8000.0

In [246]:
```
#df11.at['4','price'] = 5000 - add new record
#   Vs
df11.at[4,'price']=555   ## To modify particular cell value
```

In [248]:
```
df11
```

Out[248]:

	Key1	product	price
0	abc	prodA	1000.0
1	abc	prodB	4402.0
2	abc	prodC	3000.0
3	abc	prodD	3000.0
4	abc	prodA	555.0
5	abc	prodB	6000.0
6	abc	prodE	5412.0
7	abc	prodB	8000.0
4	NaN	NaN	5000.0

In [249]:
```
df11['Key1'].replace('abc','XYZ')
```

Out[249]:
```
0    XYZ
1    XYZ
2    XYZ
3    XYZ
4    XYZ
5    XYZ
6    XYZ
7    XYZ
4    NaN
Name: Key1, dtype: object
```

In [250]:
```
df11['Key1'].replace('abc','XYZ',inplace=True)
```

In [251]:
```
df11
```

Out[251]:

	Key1	product	price
0	XYZ	prodA	1000.0
1	XYZ	prodB	4402.0
2	XYZ	prodC	3000.0
3	XYZ	prodD	3000.0
4	XYZ	prodA	555.0
5	XYZ	prodB	6000.0
6	XYZ	prodE	5412.0
7	XYZ	prodB	8000.0
4	NaN	NaN	5000.0

In [254]:
```
df11['product'].replace('prodA','producA',inplace=True)
```

In [255]:
```
df11
```

Out[255]:

	Key1	product	price
0	XYZ	producA	1000.0
1	XYZ	prodB	4402.0
2	XYZ	prodC	3000.0
3	XYZ	prodD	3000.0
4	XYZ	producA	555.0
5	XYZ	prodB	6000.0
6	XYZ	prodE	5412.0
7	XYZ	prodB	8000.0
4	NaN	NaN	5000.0

In [256]:
```
df11['Key1'].fillna(value='XYZ',inplace=True)
df11['product'].fillna(value='prodAB',inplace=True)
```

In [257]:
```
df11
```

Out[257]:

	Key1	product	price
0	XYZ	producA	1000.0
1	XYZ	prodB	4402.0
2	XYZ	prodC	3000.0
3	XYZ	prodD	3000.0
4	XYZ	producA	555.0
5	XYZ	prodB	6000.0
6	XYZ	prodE	5412.0
7	XYZ	prodB	8000.0
4	XYZ	prodAB	5000.0

In [261]:
```
df11['product'].value_counts()
```

Out[261]:
```
product
prodB      3
producA    2
prodC      1
prodD      1
prodE      1
prodAB     1
Name: count, dtype: int64
```

In [258]:
```
df12 = pd.read_csv('E:\\emp.csv')
df12.head()
```

Out[258]:

	eid	ename	edept	eplace	ecost
0	101	raj	sales	pune	1000
1	102	leo	prod	bglore	2000
2	103	paul	HR	chennai	3000
3	104	anu	hr	hyderabad	4000
4	456	kumar	sales	bglore	3000

In [259]:
```
df12.tail()
```

Out[259]:

	eid	ename	edept	eplace	ecost
5	105	zion	Hr	mumbai	5000
6	106	bibu	sales	bglore	1450
7	107	theeb	sales	noida	4590
8	108	bibu	sales	bglore	5000
9	113	kumar	prod	hyderabad	5423

In [260]:
```
df12.sample(n=5) # random sample - take any 5 items
```

Out[260]:

	eid	ename	edept	eplace	ecost
6	106	bibu	sales	bglore	1450
5	105	zion	Hr	mumbai	5000
0	101	raj	sales	pune	1000
8	108	bibu	sales	bglore	5000
4	456	kumar	sales	bglore	3000

In [264]:
```
df12[df12['edept'] == 'sales']
```

Out[264]:

	eid	ename	edept	eplace	ecost
0	101	raj	sales	pune	1000
4	456	kumar	sales	bglore	3000
6	106	bibu	sales	bglore	1450
7	107	theeb	sales	noida	4590
8	108	bibu	sales	bglore	5000

In [265]:
```
df12.query('edept == "sales"')
```

Out[265]:

	eid	ename	edept	eplace	ecost
0	101	raj	sales	pune	1000
4	456	kumar	sales	bglore	3000
6	106	bibu	sales	bglore	1450
7	107	theeb	sales	noida	4590
8	108	bibu	sales	bglore	5000

In [267]:
```
df12.filter(items=['edept'])
```

Out[267]:

	edept
0	sales
1	prod
2	HR
3	hr
4	sales
5	Hr
6	sales
7	sales
8	sales
9	prod

In [268]:
```
df12['edept']
```

Out[268]:
```
0    sales
1     prod
2       HR
3       hr
4    sales
5       Hr
6    sales
7    sales
8    sales
9     prod
Name: edept, dtype: object
```

In [269]:
```
df12['ecost'].aggregate('sum')
```

Out[269]:
```
34463
```

In [270]:
```
df12['ecost'].aggregate('mean')
```

Out[270]:
```
3446.3
```

In [271]:
```
df12['ecost'].aggregate('max')
```

Out[271]:
```
5423
```

In [272]:
```
df12['ecost'].aggregate('min')
```

Out[272]:
```
1000
```

In [ ]:
```
# Load this data
#            |->used_cars_data.csv

1. Check no.of rows and cols
2. using info()
3. get list of columns & index
     |->3.1 select sample -> 20 items
4. check duplicate item is exists <or> not
5.           |->no.of duplicate items
6.           |->drop duplicate items
7. check NaN value is exists ->replace NaN value
8. select few cols from dataset
                        -> rename
9. drop some unwanted Cols.
10.  - display 1st 7lines & last 7lines
11.  - use query()
-------------------------------------------------
```

In [276]:
```
df = pd.read_csv('C:\\users\\Raja\\downloads\\used_cars_data.csv')
df.shape
```

Out[276]:
```
(7253, 14)
```

In [277]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7253 entries, 0 to 7252
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   S.No.              7253 non-null   int64  
 1   Name               7253 non-null   object 
 2   Location           7253 non-null   object 
 3   Year               7253 non-null   int64  
 4   Kilometers_Driven  7253 non-null   int64  
 5   Fuel_Type          7253 non-null   object 
 6   Transmission       7253 non-null   object 
 7   Owner_Type         7253 non-null   object 
 8   Mileage            7251 non-null   object 
 9   Engine             7207 non-null   object 
 10  Power              7207 non-null   object 
 11  Seats              7200 non-null   float64
 12  New_Price          1006 non-null   object 
 13  Price              6019 non-null   float64
dtypes: float64(2), int64(3), object(9)
memory usage: 793.4+ KB
```

In [278]:
```
new_df = df.sample(n=20)
new_df.shape
```

Out[278]:
```
(20, 14)
```

In [279]:
```
new_df.index
```

Out[279]:
```
Index([  94, 5320, 3139, 3883, 5109, 5407,  176, 6604, 3731, 3874, 2419, 1405,
       3051, 5944, 4744, 4606,  732, 1672, 3820, 3202],
      dtype='int64')
```

In [280]:
```
new_df.columns
```

Out[280]:
```
Index(['S.No.', 'Name', 'Location', 'Year', 'Kilometers_Driven', 'Fuel_Type',
       'Transmission', 'Owner_Type', 'Mileage', 'Engine', 'Power', 'Seats',
       'New_Price', 'Price'],
      dtype='object')
```

In [281]:
```
new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 20 entries, 94 to 3202
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   S.No.              20 non-null     int64  
 1   Name               20 non-null     object 
 2   Location           20 non-null     object 
 3   Year               20 non-null     int64  
 4   Kilometers_Driven  20 non-null     int64  
 5   Fuel_Type          20 non-null     object 
 6   Transmission       20 non-null     object 
 7   Owner_Type         20 non-null     object 
 8   Mileage            20 non-null     object 
 9   Engine             20 non-null     object 
 10  Power              20 non-null     object 
 11  Seats              20 non-null     float64
 12  New_Price          3 non-null      object 
 13  Price              19 non-null     float64
dtypes: float64(2), int64(3), object(9)
memory usage: 2.3+ KB
```

In [283]:
```
new_df.duplicated().sum()
```

Out[283]:
```
0
```

In [284]:
```
new_df.duplicated().any()
```

Out[284]:
```
False
```

In [286]:
```
new_df.isna().sum()
```

Out[286]:
```
S.No.                0
Name                 0
Location             0
Year                 0
Kilometers_Driven    0
Fuel_Type            0
Transmission         0
Owner_Type           0
Mileage              0
Engine               0
Power                0
Seats                0
New_Price           17
Price                1
dtype: int64
```

In [293]:
```
new_df.isnull().sum()/len(new_df)*100
```

Out[293]:
```
S.No.                0.0
Name                 0.0
Location             0.0
Year                 0.0
Kilometers_Driven    0.0
Fuel_Type            0.0
Transmission         0.0
Owner_Type           0.0
Mileage              0.0
Engine               0.0
Power                0.0
Seats                0.0
New_Price           85.0
Price                5.0
dtype: float64
```

In [294]:
```
new_df.fillna(value=new_df.isnull().sum()/len(new_df)*100,inplace=True)
```

In [295]:
```
new_df.isna().sum()
```

Out[295]:
```
S.No.               0
Name                0
Location            0
Year                0
Kilometers_Driven   0
Fuel_Type           0
Transmission        0
Owner_Type          0
Mileage             0
Engine              0
Power               0
Seats               0
New_Price           0
Price               0
dtype: int64
```

In [296]:
```
new_df.columns
```

Out[296]:
```
Index(['S.No.', 'Name', 'Location', 'Year', 'Kilometers_Driven', 'Fuel_Type',
       'Transmission', 'Owner_Type', 'Mileage', 'Engine', 'Power', 'Seats',
       'New_Price', 'Price'],
      dtype='object')
```

In [297]:
```
new_df.drop(['S.No.','Location','Owner_Type','Power','Seats'],axis=1,inplace=True)
```

In [298]:
```
new_df.shape
```

Out[298]:
```
(20, 9)
```

In [299]:
```
new_df.columns
```

Out[299]:
```
Index(['Name', 'Year', 'Kilometers_Driven', 'Fuel_Type', 'Transmission',
       'Mileage', 'Engine', 'New_Price', 'Price'],
      dtype='object')
```

In [304]:
```
new_df.rename(columns={'Kilometers_Driven':'Km'},inplace=True)
```

In [305]:
```
new_df.columns
```

Out[305]:
```
Index(['Name', 'Year', 'Km', 'Fuel_Type', 'Transmission', 'Mileage', 'Engine',
       'New_Price', 'Price'],
      dtype='object')
```

In [306]:
```
new_df.rename(columns={'Transmission':'Trans','Mileage':'Mile'},inplace=True)
```

In [307]:
```
new_df.columns
```

Out[307]:
```
Index(['Name', 'Year', 'Km', 'Fuel_Type', 'Trans', 'Mile', 'Engine',
       'New_Price', 'Price'],
      dtype='object')
```

In [309]:
```
new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 20 entries, 94 to 3202
Data columns (total 9 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Name       20 non-null     object 
 1   Year       20 non-null     int64  
 2   Km         20 non-null     int64  
 3   Fuel_Type  20 non-null     object 
 4   Trans      20 non-null     object 
 5   Mile       20 non-null     object 
 6   Engine     20 non-null     object 
 7   New_Price  20 non-null     object 
 8   Price      20 non-null     float64
dtypes: float64(1), int64(2), object(6)
memory usage: 1.6+ KB
```

In [311]:
```
new_df.isna().sum()
```

Out[311]:
```
Name         0
Year         0
Km           0
Fuel_Type    0
Trans        0
Mile         0
Engine       0
New_Price    0
Price        0
dtype: int64
```

In [313]:
```
new_df.duplicated().sum()
```

Out[313]:
```
0
```

In [322]:
```
new_df.head(7)
```

Out[322]:

	Name	Year	Km	Fuel_Type	Trans	Mile	Engine	New_Price	Price
94	Mahindra XUV500 W8 2WD	2014	58000	Diesel	Manual	15.1 kmpl	2179 CC	85.0	8.10
5320	Volkswagen Polo 1.2 MPI Highline	2016	7523	Petrol	Manual	16.2 kmpl	1199 CC	85.0	6.15
3139	Hyundai i10 Sportz	2013	32776	Petrol	Manual	20.36 kmpl	1197 CC	85.0	4.64
3883	Maruti Ertiga ZXI	2013	82543	Petrol	Manual	16.02 kmpl	1373 CC	85.0	5.50
5109	Hyundai Grand i10 CRDi Magna	2015	63776	Diesel	Manual	24.0 kmpl	1120 CC	85.0	3.96
5407	Nissan Micra Diesel XV	2011	39010	Diesel	Manual	23.08 kmpl	1461 CC	85.0	2.60
176	Mini Countryman Cooper D	2017	8525	Diesel	Automatic	16.6 kmpl	1998 CC	85.0	23.00

In [321]:
```
new_df.tail(7)
```

Out[321]:

	Name	Year	Km	Fuel_Type	Trans	Mile	Engine	New_Price	Price
5944	Ford EcoSport 1.0 Ecoboost Titanium Plus	2015	64000	Petrol	Manual	18.88 kmpl	999 CC	85.0	6.00
4744	Hyundai Santro Xing XL eRLX Euro III	2005	45000	Petrol	Manual	17.0 kmpl	1086 CC	85.0	0.95
4606	Maruti Zen LXi - BS III	2006	80000	Petrol	Manual	17.3 kmpl	993 CC	85.0	1.20
732	Honda City i-VTEC V	2014	55818	Petrol	Manual	17.4 kmpl	1497 CC	11.85 Lakh	5.70
1672	Nissan Micra Diesel	2011	80000	Diesel	Manual	19.5 kmpl	1461 CC	85.0	2.10
3820	Maruti Alto K10 VXI	2011	28300	Petrol	Manual	24.07 kmpl	998 CC	4.5 Lakh	2.15
3202	Maruti Swift VDI	2014	50000	Diesel	Manual	22.9 kmpl	1248 CC	85.0	4.30

In [319]:
```
new_df.query('Km >30000 and Fuel_Type == "Diesel"')
```

Out[319]:

	Name	Year	Km	Fuel_Type	Trans	Mile	Engine	New_Price	Price
94	Mahindra XUV500 W8 2WD	2014	58000	Diesel	Manual	15.1 kmpl	2179 CC	85.0	8.10
5109	Hyundai Grand i10 CRDi Magna	2015	63776	Diesel	Manual	24.0 kmpl	1120 CC	85.0	3.96
5407	Nissan Micra Diesel XV	2011	39010	Diesel	Manual	23.08 kmpl	1461 CC	85.0	2.60
6604	Nissan Sunny 2011-2014 Diesel XL	2012	45000	Diesel	Manual	21.64 kmpl	1461 CC	85.0	5.00
3731	Nissan Terrano XV D Pre	2013	60000	Diesel	Manual	19.64 kmpl	1461 CC	17.21 Lakh	5.45
3874	Ford Ecosport 1.5 DV5 MT Titanium Optional	2015	45000	Diesel	Manual	22.7 kmpl	1498 CC	85.0	8.90
1405	Maruti Ertiga ZDI	2013	70000	Diesel	Manual	20.77 kmpl	1248 CC	85.0	5.45
1672	Nissan Micra Diesel	2011	80000	Diesel	Manual	19.5 kmpl	1461 CC	85.0	2.10
3202	Maruti Swift VDI	2014	50000	Diesel	Manual	22.9 kmpl	1248 CC	85.0	4.30

In [320]:
```
new_df.query('Km >30000 and Km <50000 and Fuel_Type == "Diesel"')
```

Out[320]:

	Name	Year	Km	Fuel_Type	Trans	Mile	Engine	New_Price	Price
5407	Nissan Micra Diesel XV	2011	39010	Diesel	Manual	23.08 kmpl	1461 CC	85.0	2.6
6604	Nissan Sunny 2011-2014 Diesel XL	2012	45000	Diesel	Manual	21.64 kmpl	1461 CC	85.0	5.0
3874	Ford Ecosport 1.5 DV5 MT Titanium Optional	2015	45000	Diesel	Manual	22.7 kmpl	1498 CC	85.0	8.9

In [327]:
```
new_df['Mile'].replace('[a-zA-Z\s]', value='',regex=True)
```

Out[327]:
```
94       15.1
5320     16.2
3139    20.36
3883    16.02
5109     24.0
5407    23.08
176      16.6
6604    21.64
3731    19.64
3874     22.7
2419     18.0
1405    20.77
3051     11.3
5944    18.88
4744     17.0
4606     17.3
732      17.4
1672     19.5
3820    24.07
3202     22.9
Name: Mile, dtype: object
```

In [332]:
```
# delete all the alpha and / chars - this is regx substitute logic
new_df['Mile'].replace('[a-zA-Z\s\/]', value='',regex=True,inplace=True)
```

In [333]:
```
new_df.query('Km >30000 and Km <50000 and Fuel_Type == "Diesel"')
```

Out[333]:

	Name	Year	Km	Fuel_Type	Trans	Mile	Engine	New_Price	Price
5407	Nissan Micra Diesel XV	2011	39010	Diesel	Manual	23.08	1461 CC	85.0	2.6
6604	Nissan Sunny 2011-2014 Diesel XL	2012	45000	Diesel	Manual	21.64	1461 CC	85.0	5.0
3874	Ford Ecosport 1.5 DV5 MT Titanium Optional	2015	45000	Diesel	Manual	22.7	1498 CC	85.0	8.9

In [334]:
```
new_df['Mile']
```

Out[334]:
```
94       15.1
5320     16.2
3139    20.36
3883    16.02
5109     24.0
5407    23.08
176      16.6
6604    21.64
3731    19.64
3874     22.7
2419     18.0
1405    20.77
3051     11.3
5944    18.88
4744     17.0
4606     17.3
732      17.4
1672     19.5
3820    24.07
3202     22.9
Name: Mile, dtype: object
```

In [335]:
```
d={'K1':[10,20,30],'K2':[15,25,35],'K3':[100,200,300]}
df = pd.DataFrame(d)
df
```

Out[335]:

	K1	K2	K3
0	10	15	100
1	20	25	200
2	30	35	300

In [341]:
```
df = pd.DataFrame(d,index=['i1','i2','i3'])
df
```

Out[341]:

	K1	K2	K3
i1	10	15	100
i2	20	25	200
i3	30	35	300

In [343]:
```
pd.read_csv('E:\\emp.csv',index_col='edept')
```

Out[343]:

edept	eid	ename	eplace	ecost
sales	101	raj	pune	1000
prod	102	leo	bglore	2000
HR	103	paul	chennai	3000
hr	104	anu	hyderabad	4000
sales	456	kumar	bglore	3000
Hr	105	zion	mumbai	5000
sales	106	bibu	bglore	1450
sales	107	theeb	noida	4590
sales	108	bibu	bglore	5000
prod	113	kumar	hyderabad	5423

In [345]:
```
df = pd.read_csv('E:\\emp.csv')
df
```

Out[345]:

	eid	ename	edept	eplace	ecost
0	101	raj	sales	pune	1000
1	102	leo	prod	bglore	2000
2	103	paul	HR	chennai	3000
3	104	anu	hr	hyderabad	4000
4	456	kumar	sales	bglore	3000
5	105	zion	Hr	mumbai	5000
6	106	bibu	sales	bglore	1450
7	107	theeb	sales	noida	4590
8	108	bibu	sales	bglore	5000
9	113	kumar	prod	hyderabad	5423

In [346]:
```
df.index
```

Out[346]:
```
RangeIndex(start=0, stop=10, step=1)
```

In [347]:
```
df['eplace']
```

Out[347]:
```
0         pune
1       bglore
2      chennai
3    hyderabad
4       bglore
5       mumbai
6       bglore
7        noida
8       bglore
9    hyderabad
Name: eplace, dtype: object
```

In [348]:
```
df.index = df['eplace'] # set the index
```

In [349]:
```
df
```

Out[349]:

eplace	eid	ename	edept	eplace	ecost
pune	101	raj	sales	pune	1000
bglore	102	leo	prod	bglore	2000
chennai	103	paul	HR	chennai	3000
hyderabad	104	anu	hr	hyderabad	4000
bglore	456	kumar	sales	bglore	3000
mumbai	105	zion	Hr	mumbai	5000
bglore	106	bibu	sales	bglore	1450
noida	107	theeb	sales	noida	4590
bglore	108	bibu	sales	bglore	5000
hyderabad	113	kumar	prod	hyderabad	5423

In [350]:
```
new_df.index
```

Out[350]:
```
Index([  94, 5320, 3139, 3883, 5109, 5407,  176, 6604, 3731, 3874, 2419, 1405,
       3051, 5944, 4744, 4606,  732, 1672, 3820, 3202],
      dtype='int64')
```

In [351]:
```
new_df['Year']
```

Out[351]:
```
94      2014
5320    2016
3139    2013
3883    2013
5109    2015
5407    2011
176     2017
6604    2012
3731    2013
3874    2015
2419    2014
1405    2013
3051    2016
5944    2015
4744    2005
4606    2006
732     2014
1672    2011
3820    2011
3202    2014
Name: Year, dtype: int64
```

In [352]:
```
new_df.index = new_df['Year']
```

In [353]:
```
new_df
```

Out[353]:

Year	Name	Year	Km	Fuel_Type	Trans	Mile	Engine	New_Price	Price
2014	Mahindra XUV500 W8 2WD	2014	58000	Diesel	Manual	15.1	2179 CC	85.0	8.10
2016	Volkswagen Polo 1.2 MPI Highline	2016	7523	Petrol	Manual	16.2	1199 CC	85.0	6.15
2013	Hyundai i10 Sportz	2013	32776	Petrol	Manual	20.36	1197 CC	85.0	4.64
2013	Maruti Ertiga ZXI	2013	82543	Petrol	Manual	16.02	1373 CC	85.0	5.50
2015	Hyundai Grand i10 CRDi Magna	2015	63776	Diesel	Manual	24.0	1120 CC	85.0	3.96
2011	Nissan Micra Diesel XV	2011	39010	Diesel	Manual	23.08	1461 CC	85.0	2.60
2017	Mini Countryman Cooper D	2017	8525	Diesel	Automatic	16.6	1998 CC	85.0	23.00
2012	Nissan Sunny 2011-2014 Diesel XL	2012	45000	Diesel	Manual	21.64	1461 CC	85.0	5.00
2013	Nissan Terrano XV D Pre	2013	60000	Diesel	Manual	19.64	1461 CC	17.21 Lakh	5.45
2015	Ford Ecosport 1.5 DV5 MT Titanium Optional	2015	45000	Diesel	Manual	22.7	1498 CC	85.0	8.90
2014	Honda Amaze S i-Vtech	2014	69876	Petrol	Manual	18.0	1198 CC	85.0	3.90
2013	Maruti Ertiga ZDI	2013	70000	Diesel	Manual	20.77	1248 CC	85.0	5.45
2016	Mercedes-Benz GL-Class 350 CDI Blue Efficiency	2016	16000	Diesel	Automatic	11.3	2987 CC	85.0	56.00
2015	Ford EcoSport 1.0 Ecoboost Titanium Plus	2015	64000	Petrol	Manual	18.88	999 CC	85.0	6.00
2005	Hyundai Santro Xing XL eRLX Euro III	2005	45000	Petrol	Manual	17.0	1086 CC	85.0	0.95
2006	Maruti Zen LXi - BS III	2006	80000	Petrol	Manual	17.3	993 CC	85.0	1.20
2014	Honda City i-VTEC V	2014	55818	Petrol	Manual	17.4	1497 CC	11.85 Lakh	5.70
2011	Nissan Micra Diesel	2011	80000	Diesel	Manual	19.5	1461 CC	85.0	2.10
2011	Maruti Alto K10 VXI	2011	28300	Petrol	Manual	24.07	998 CC	4.5 Lakh	2.15
2014	Maruti Swift VDI	2014	50000	Diesel	Manual	22.9	1248 CC	85.0	4.30

In [354]:
```
new_df.loc[2016]
```

Out[354]:

Year	Name	Year	Km	Fuel_Type	Trans	Mile	Engine	New_Price	Price
2016	Volkswagen Polo 1.2 MPI Highline	2016	7523	Petrol	Manual	16.2	1199 CC	85.0	6.15
2016	Mercedes-Benz GL-Class 350 CDI Blue Efficiency	2016	16000	Diesel	Automatic	11.3	2987 CC	85.0	56.00

In [355]:
```
new_df.to_excel('E:\\test1.xlsx')
# Write to Excel file
```

In [ ]:
```

```

In [377]:
```
import requests
import bs4
```

In [378]:
```
def download_url(url):
    r=requests.get(url)
    if(r.status_code != 200):
        print('url download is failed')
        return False
    if('text/html' in r.headers['Content-Type']):
        ol_web = r.text
        webparsing(ol_web)
    else:
        return 'Given url is not a webpage content'
```

In [379]:
```
Linux_rpm=[]
def webparsing(webpage):
    print(webpage)
    ol_obj = bs4.BeautifulSoup(webpage)
    for var in ol_obj.find_all('a'):
        if(re.search('rpm$',var.get('href'))):
            Linux_rpm.append(var.string)
```

In [380]:
```
ol_rpm={}
```

In [381]:
```
ol7_url='https://yum.oracle.com/repo/OracleLinux/OL7/latest/x86_64/index.html'
ol8_url='https://yum.oracle.com/repo/OracleLinux/OL8/baseos/latest/x86_64/index.html'
ol9_url='https://yum.oracle.com/repo/OracleLinux/OL9/baseos/latest/x86_64/index.html'
```

In [382]:
```
download_url(ol7_url)
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

In [383]:
```
ol_rpm['OL7']=Linux_rpm
```

In [385]:
```
download_url(ol8_url)
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

In [386]:
```
ol_rpm['OL8']=Linux_rpm
```

In [388]:
```
download_url(ol9_url)
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

In [389]:
```
ol_rpm['OL9']=Linux_rpm
```

In [390]:
```
ol_df = pd.DataFrame(ol_rpm)
ol_df.to_excel('E:\\ol_rpm.xlsx') # write ol rpm df ->excel
```

In [362]:
```
new_df[new_df.axes[1]]
```

Out[362]:

Year	Name	Year	Km	Fuel_Type	Trans	Mile	Engine	New_Price	Price
2014	Mahindra XUV500 W8 2WD	2014	58000	Diesel	Manual	15.1	2179 CC	85.0	8.10
2016	Volkswagen Polo 1.2 MPI Highline	2016	7523	Petrol	Manual	16.2	1199 CC	85.0	6.15
2013	Hyundai i10 Sportz	2013	32776	Petrol	Manual	20.36	1197 CC	85.0	4.64
2013	Maruti Ertiga ZXI	2013	82543	Petrol	Manual	16.02	1373 CC	85.0	5.50
2015	Hyundai Grand i10 CRDi Magna	2015	63776	Diesel	Manual	24.0	1120 CC	85.0	3.96
2011	Nissan Micra Diesel XV	2011	39010	Diesel	Manual	23.08	1461 CC	85.0	2.60
2017	Mini Countryman Cooper D	2017	8525	Diesel	Automatic	16.6	1998 CC	85.0	23.00
2012	Nissan Sunny 2011-2014 Diesel XL	2012	45000	Diesel	Manual	21.64	1461 CC	85.0	5.00
2013	Nissan Terrano XV D Pre	2013	60000	Diesel	Manual	19.64	1461 CC	17.21 Lakh	5.45
2015	Ford Ecosport 1.5 DV5 MT Titanium Optional	2015	45000	Diesel	Manual	22.7	1498 CC	85.0	8.90
2014	Honda Amaze S i-Vtech	2014	69876	Petrol	Manual	18.0	1198 CC	85.0	3.90
2013	Maruti Ertiga ZDI	2013	70000	Diesel	Manual	20.77	1248 CC	85.0	5.45
2016	Mercedes-Benz GL-Class 350 CDI Blue Efficiency	2016	16000	Diesel	Automatic	11.3	2987 CC	85.0	56.00
2015	Ford EcoSport 1.0 Ecoboost Titanium Plus	2015	64000	Petrol	Manual	18.88	999 CC	85.0	6.00
2005	Hyundai Santro Xing XL eRLX Euro III	2005	45000	Petrol	Manual	17.0	1086 CC	85.0	0.95
2006	Maruti Zen LXi - BS III	2006	80000	Petrol	Manual	17.3	993 CC	85.0	1.20
2014	Honda City i-VTEC V	2014	55818	Petrol	Manual	17.4	1497 CC	11.85 Lakh	5.70
2011	Nissan Micra Diesel	2011	80000	Diesel	Manual	19.5	1461 CC	85.0	2.10
2011	Maruti Alto K10 VXI	2011	28300	Petrol	Manual	24.07	998 CC	4.5 Lakh	2.15
2014	Maruti Swift VDI	2014	50000	Diesel	Manual	22.9	1248 CC	85.0	4.30

In [391]:
```
# matplotlib -> Visualization

import matplotlib
```

In [392]:
```
matplotlib.__version__
```

Out[392]: '3.7.2'

In [393]:
```
matplotlib.__file__
```

Out[393]:
```
'C:\\ProgramData\\anaconda3\\Lib\\site-packages\\matplotlib\\__init__.py'
```

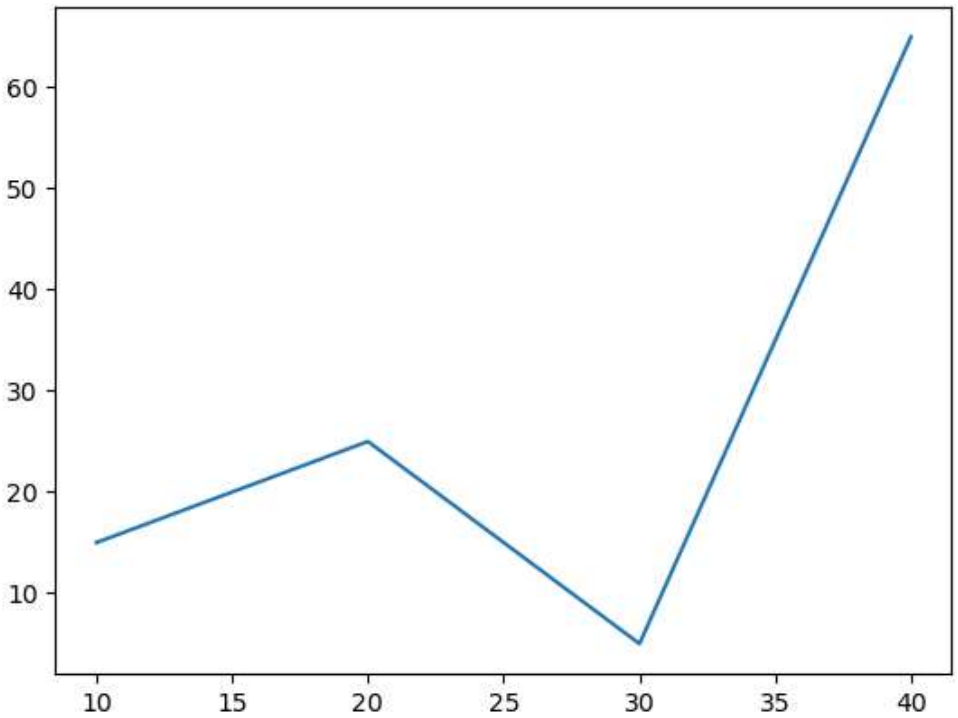
In [ ]:
```
matplotlib/ 1st
        |--->pyplot.py 2nd
               |--------->plot(),scatter(),....
                          ........................


import matplotlib.pyplot
matplotlib.pyplot.plot()
Vs

import matplotlib.pyplot as plt
plt.plot()
plt.scatter()
...
```

In [394]:
```
import matplotlib.pyplot
print(matplotlib.pyplot)
```

```
<module 'matplotlib.pyplot' from 'C:\\ProgramData\\anaconda3\\Lib\\site-packages\\matplotl
ib\\pyplot.py'>
```

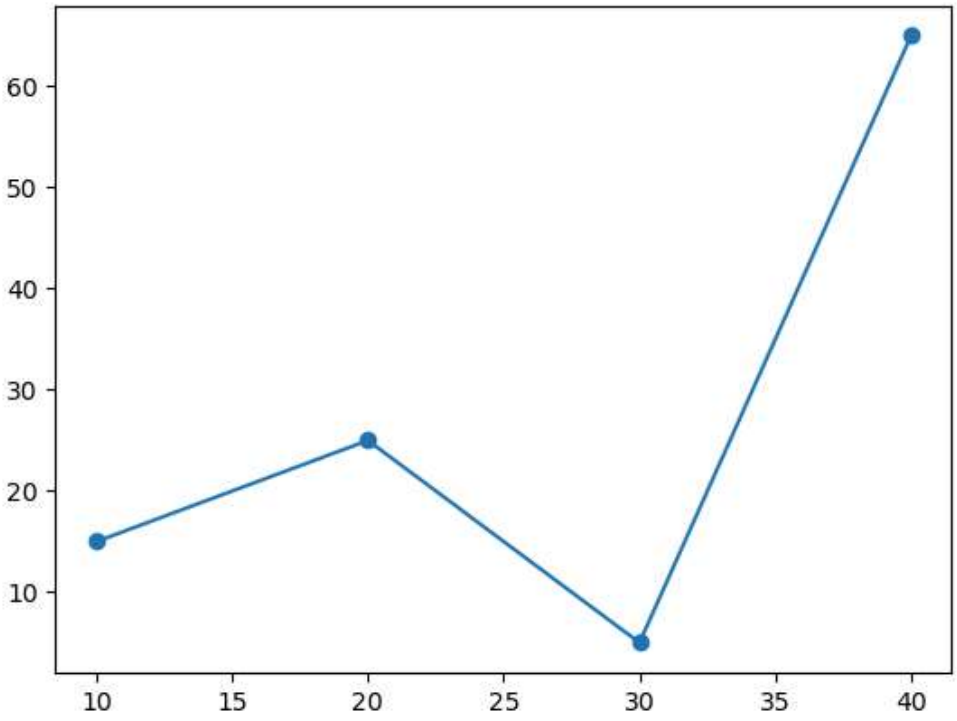
In [397]:
```
import matplotlib.pyplot as plt
#help(plt.plot)

plt.plot([10,20,30,40],[15,25,5,65])
```

Out[397]:
```
[<matplotlib.lines.Line2D at 0x1c297cf1010>]
```



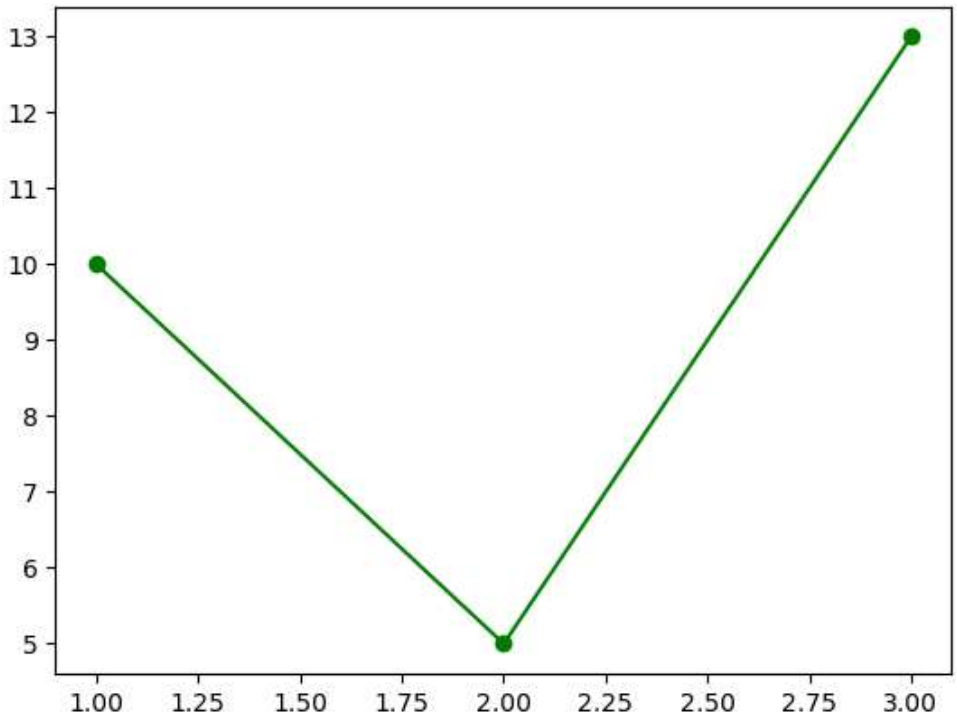
In [398]:
```
plt.plot([10,20,30,40],[15,25,5,65],marker='o')
```

Out[398]:
```
[<matplotlib.lines.Line2D at 0x1c29855f910>]
```



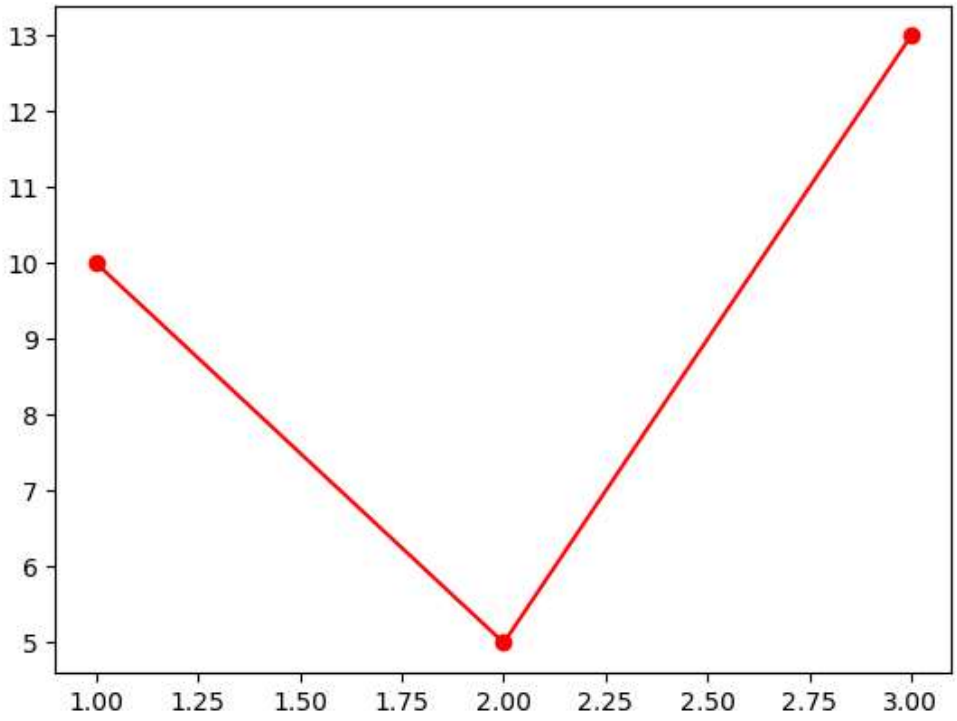
In [405]:
```
plt.plot([1, 2, 3], [10, 5, 13], 'go-')
```

Out[405]:
```
[<matplotlib.lines.Line2D at 0x1c297c37f90>]
```



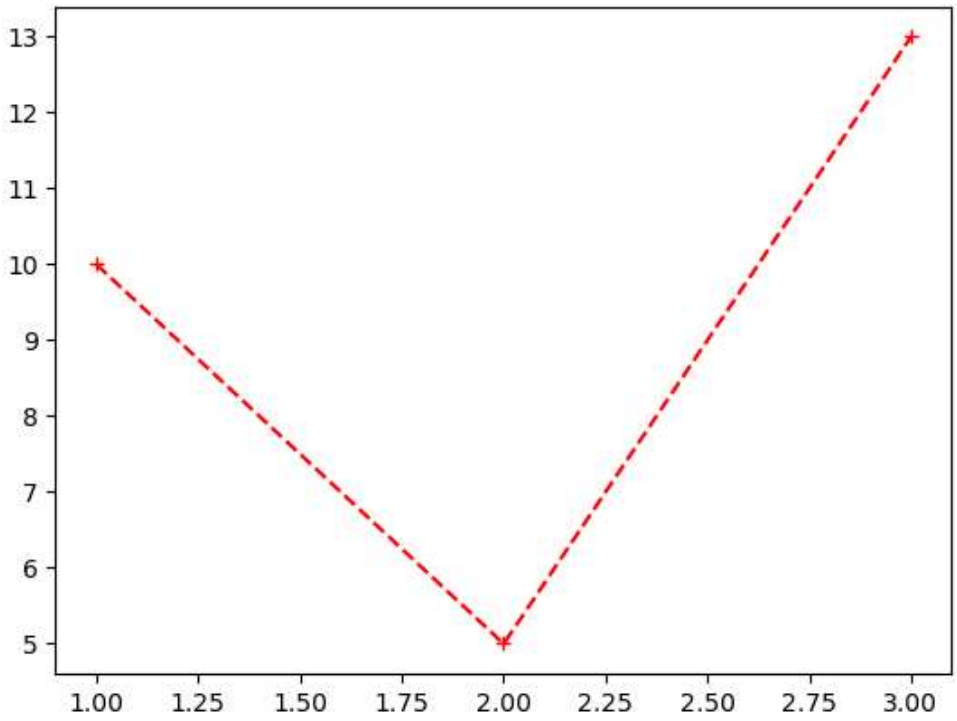
In [406]:
```
plt.plot([1, 2, 3], [10, 5, 13], 'ro-')
```

Out[406]:
```
[<matplotlib.lines.Line2D at 0x1c298624c10>]
```



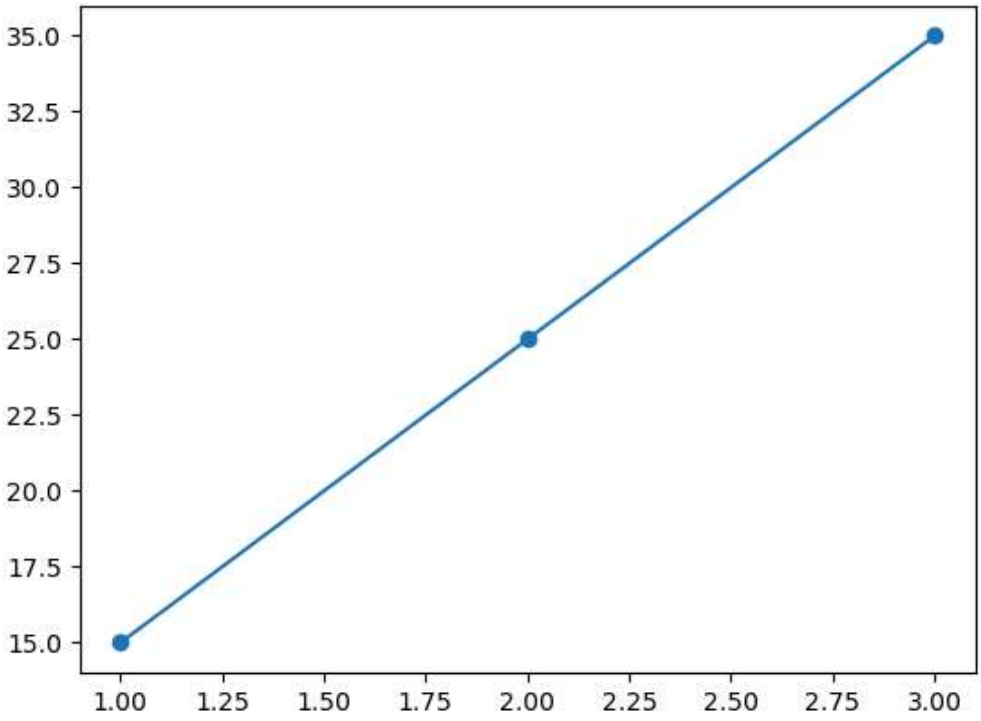
In [409]:
```
plt.plot([1, 2, 3], [10, 5, 13], 'r+--')
```

Out[409]:
```
[<matplotlib.lines.Line2D at 0x1c298bc3610>]
```



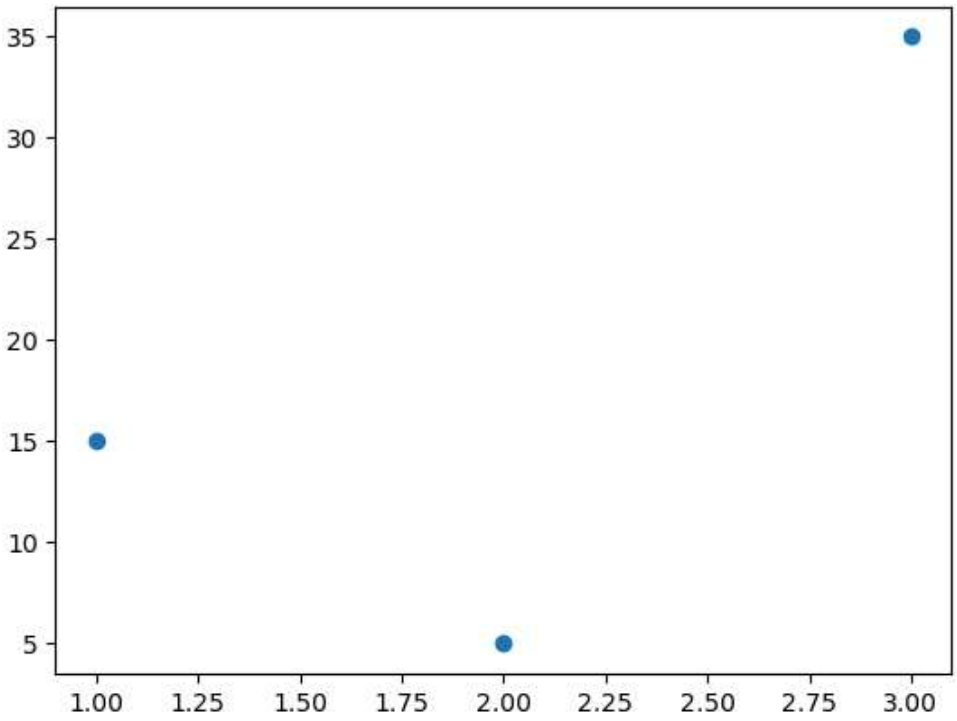
In [411]:
```
plt.plot([1,2,3],[15,25,35],marker='o')
```

Out[411]:
```
[<matplotlib.lines.Line2D at 0x1c298cc2590>]
```



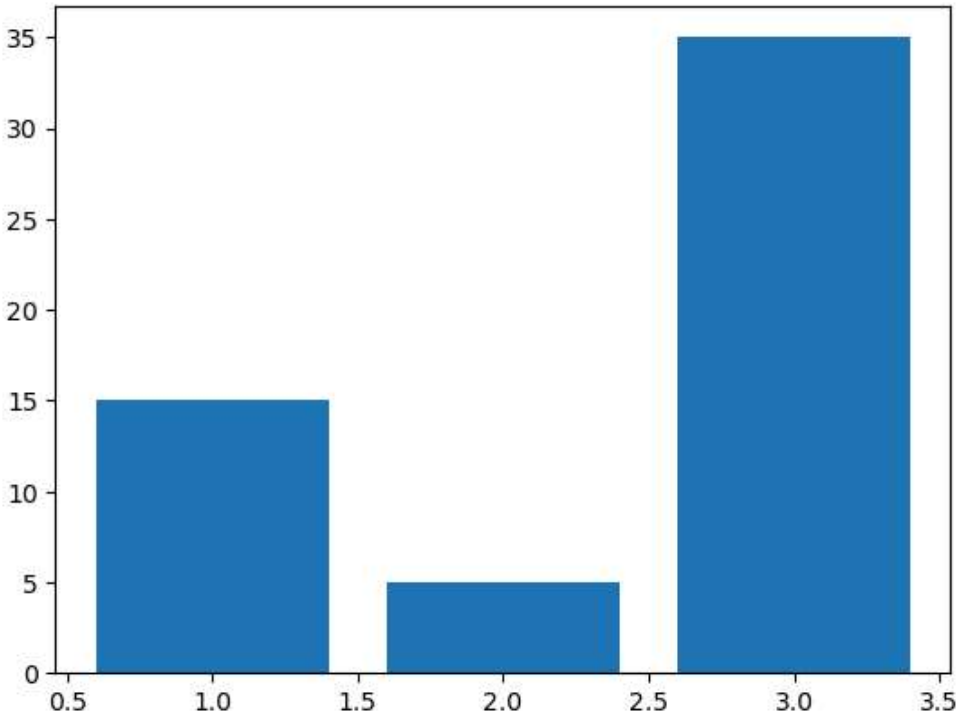
In [413]:
```
plt.scatter([1,2,3],[15,5,35])
```

Out[413]:
```
<matplotlib.collections.PathCollection at 0x1c298d8bf10>
```



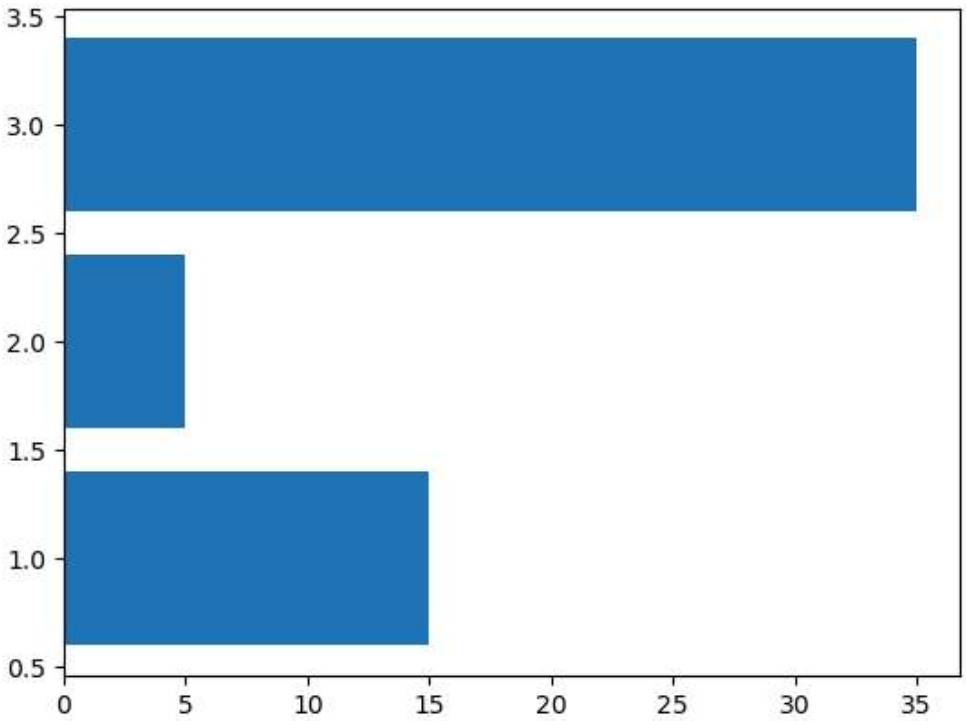
In [414]:
```
plt.bar([1,2,3],[15,5,35])
```

Out[414]:
```
<BarContainer object of 3 artists>
```



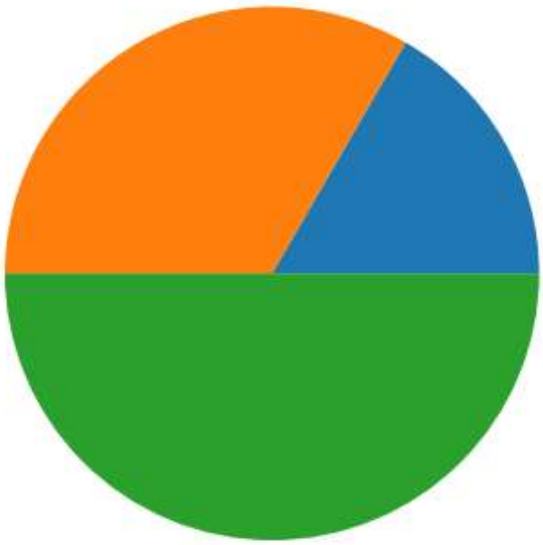
In [415]:
```
plt.barh([1,2,3],[15,5,35])
```

Out[415]:
```
<BarContainer object of 3 artists>
```



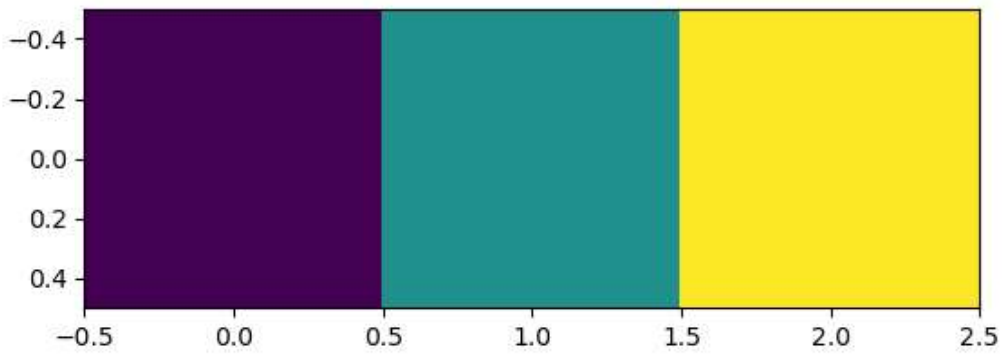
In [417]:
```
plt.pie([1,2,3])
```

Out[417]:
```
([<matplotlib.patches.Wedge at 0x1c29894c950>,
  <matplotlib.patches.Wedge at 0x1c29894d950>,
  <matplotlib.patches.Wedge at 0x1c29894eb10>],
 [Text(0.9526279355804298, 0.5500000148652441, ''),
  Text(-0.5500000594609755, 0.9526279098330699, ''),
  Text(1.0298943251329445e-07, -1.0999999999999954, '')])
```



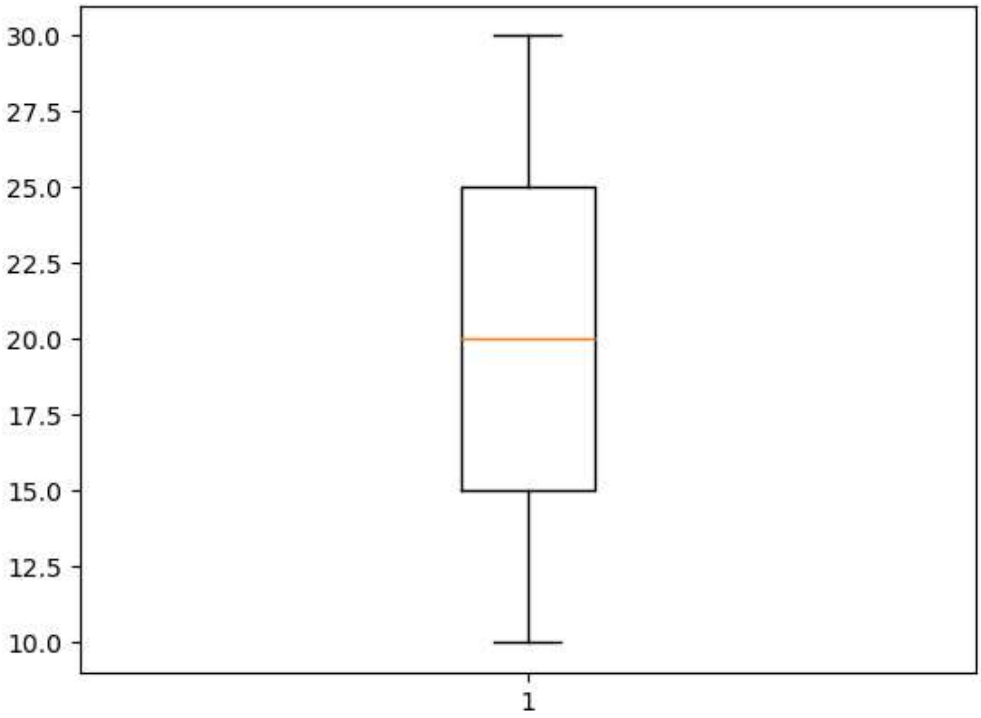
In [418]:
```
plt.imshow([[10,20,30]])
```

Out[418]:
```
<matplotlib.image.AxesImage at 0x1c298973b50>
```



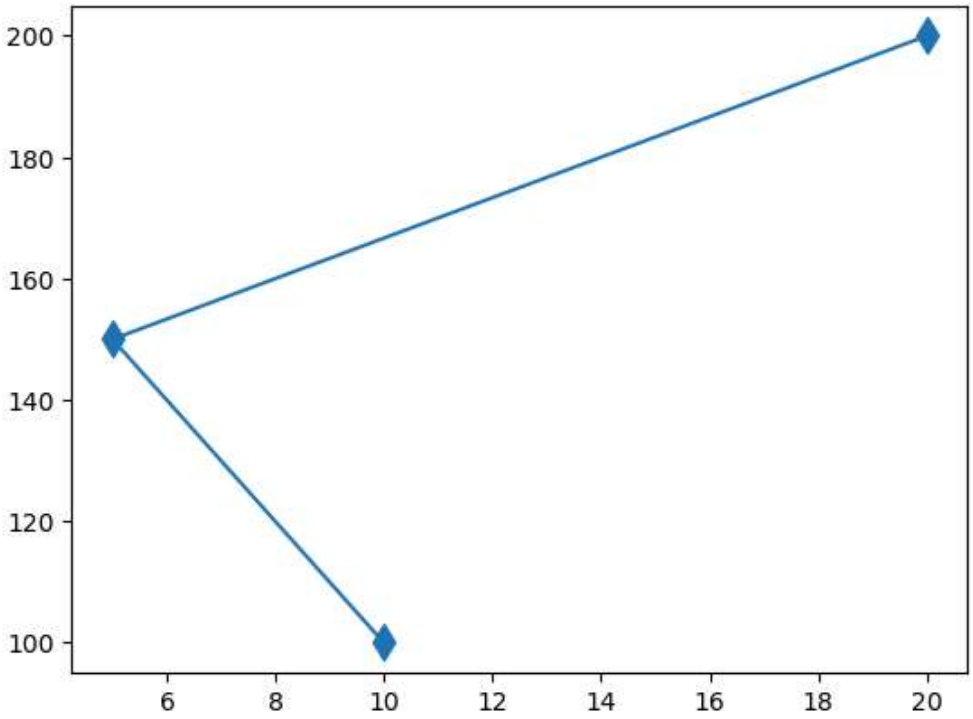
In [419]:
```
plt.boxplot([10,20,30])
```

Out[419]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1c298a07210>,
  <matplotlib.lines.Line2D at 0x1c298a07e50>],
 'caps': [<matplotlib.lines.Line2D at 0x1c2989e1f90>,
  <matplotlib.lines.Line2D at 0x1c298e15590>],
 'boxes': [<matplotlib.lines.Line2D at 0x1c298a06450>],
 'medians': [<matplotlib.lines.Line2D at 0x1c298e16090>],
 'fliers': [<matplotlib.lines.Line2D at 0x1c298e16a90>],
 'means': []}
```



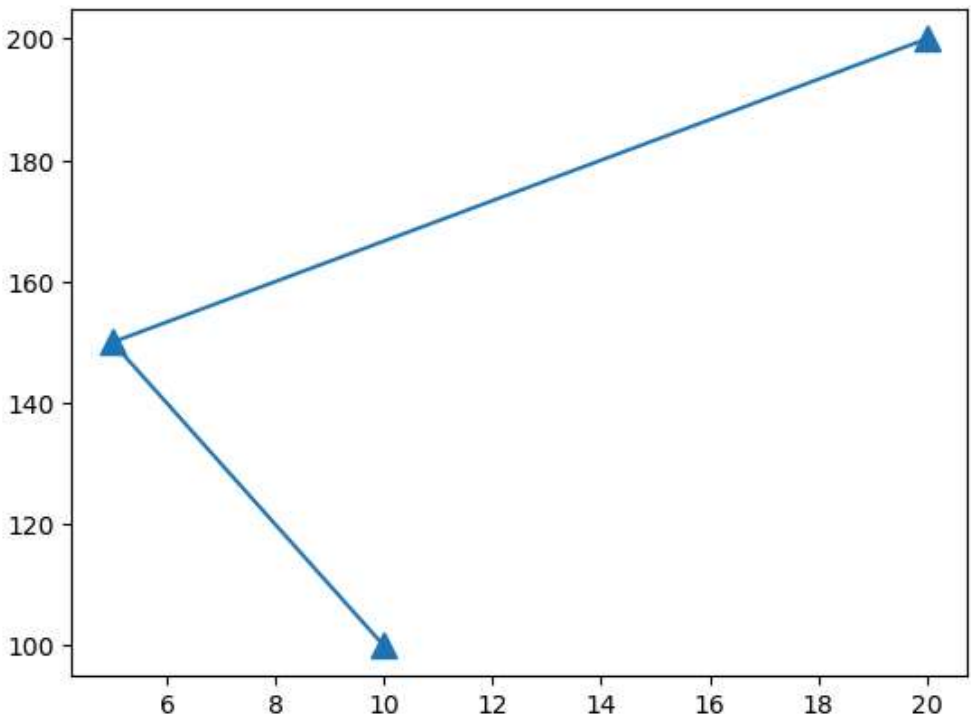
In [422]:
```
plt.plot([10,5,20],[100,150,200],marker='d',ms=10)
```

Out[422]:
```
[<matplotlib.lines.Line2D at 0x1c2a18d09d0>]
```



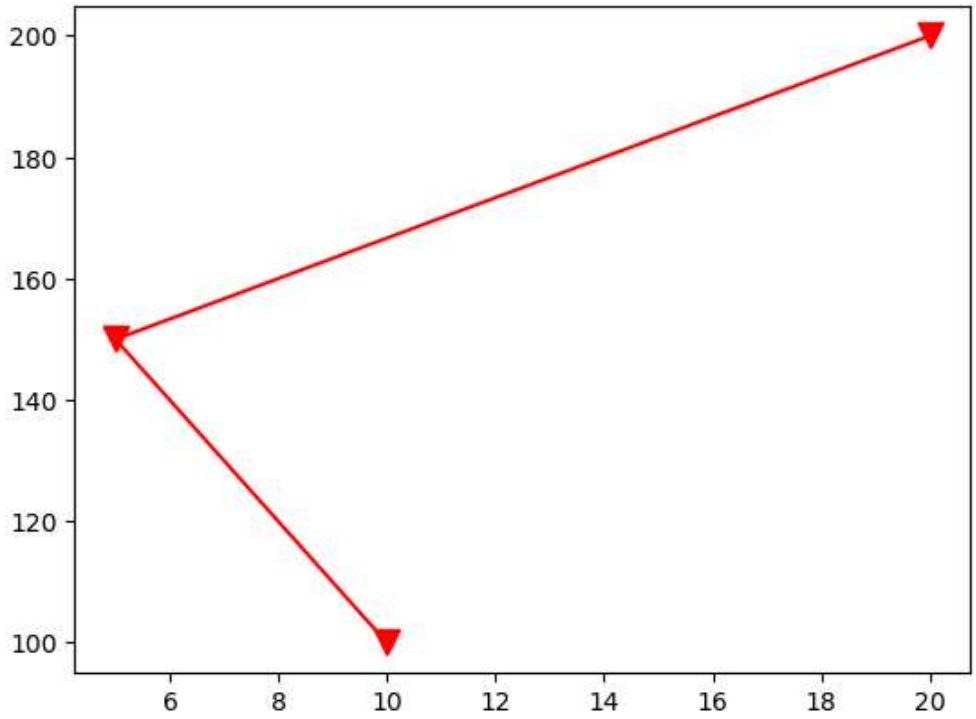
In [423]:
```
plt.plot([10,5,20],[100,150,200],marker='^',ms=10)
```

Out[423]:
```
[<matplotlib.lines.Line2D at 0x1c2a195f010>]
```



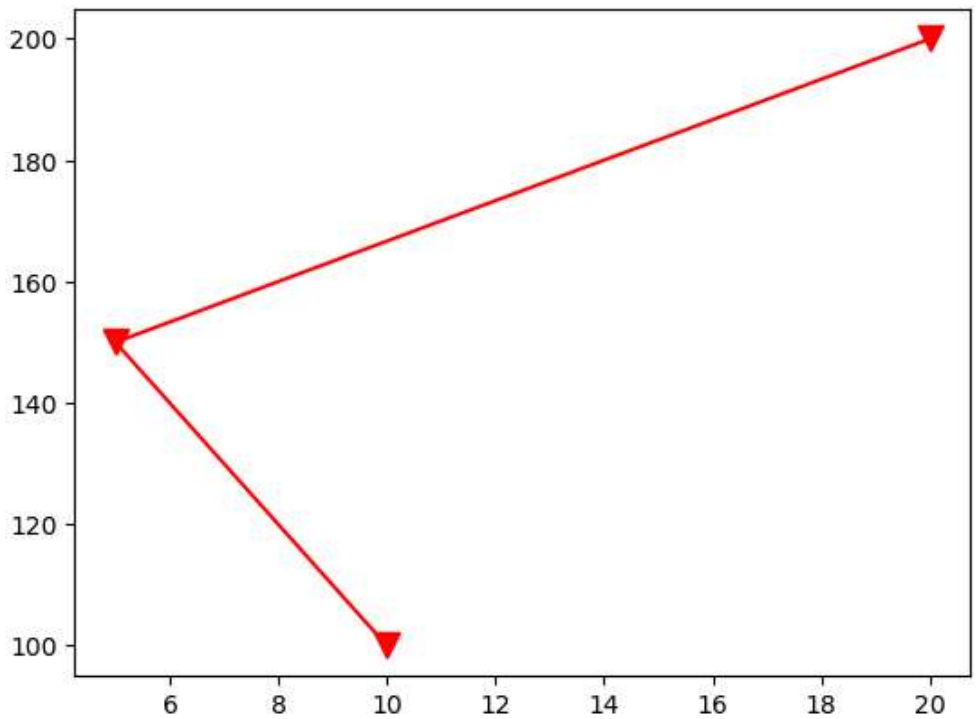
In [425]:
```
plt.plot([10,5,20],[100,150,200],marker='v',ms=10,color='r')
```

Out[425]:
```
[<matplotlib.lines.Line2D at 0x1c298f4c6d0>]
```



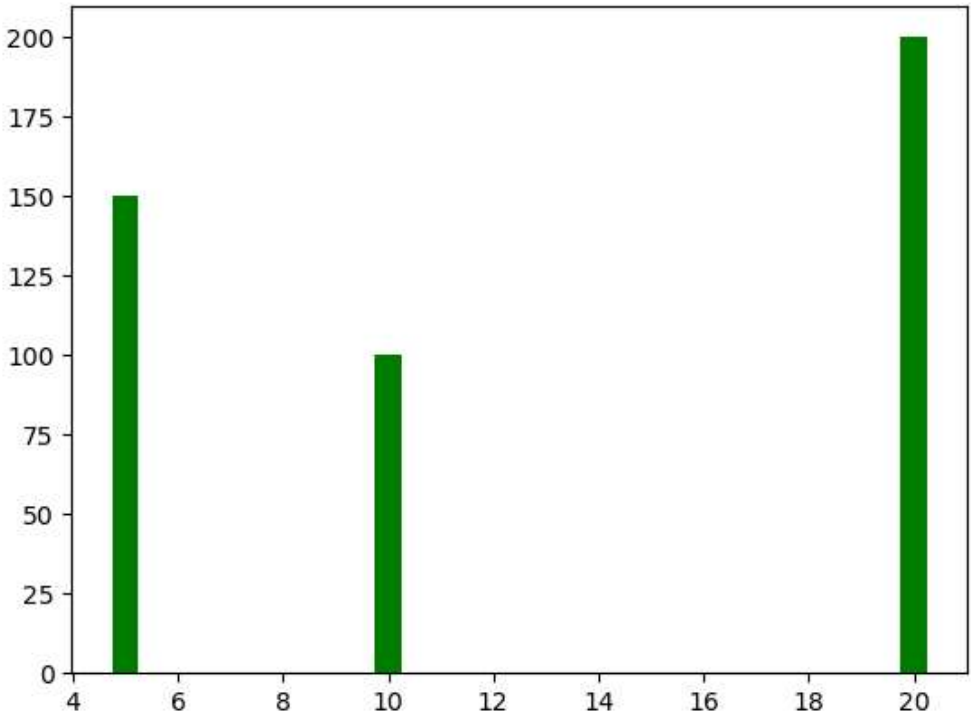
In [425]:
```
plt.plot([10,5,20],[100,150,200],marker='v',ms=10,color='r')
```

Out[425]:
```
[<matplotlib.lines.Line2D at 0x1c298f4c6d0>]
```



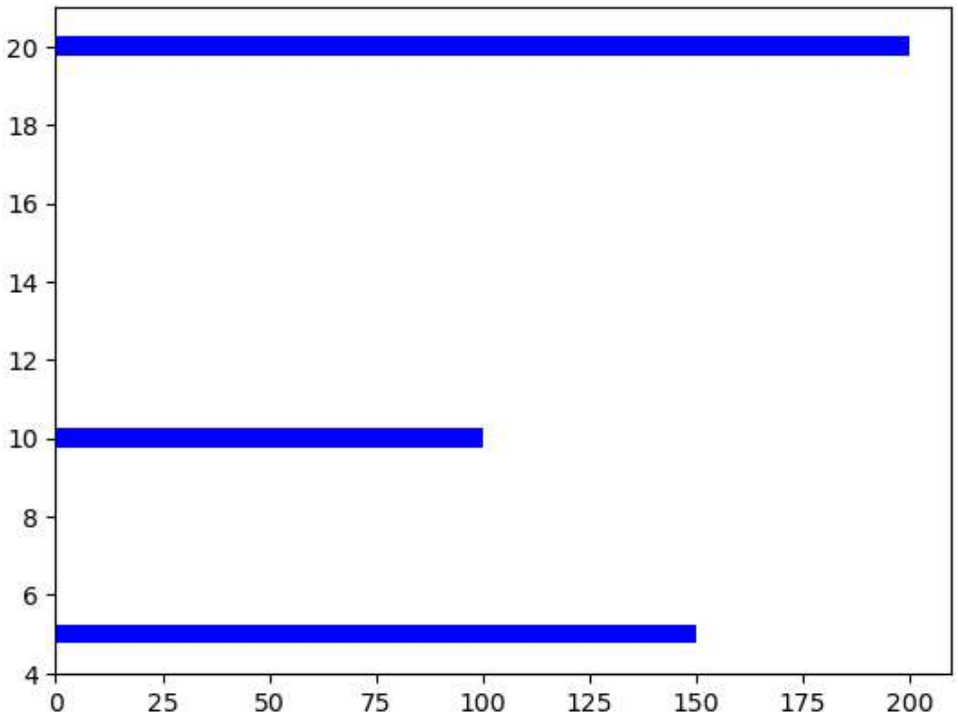
In [430]:
```
plt.bar([10,5,20],[100,150,200],width=0.5,color='g')
```

Out[430]:
```
<BarContainer object of 3 artists>
```



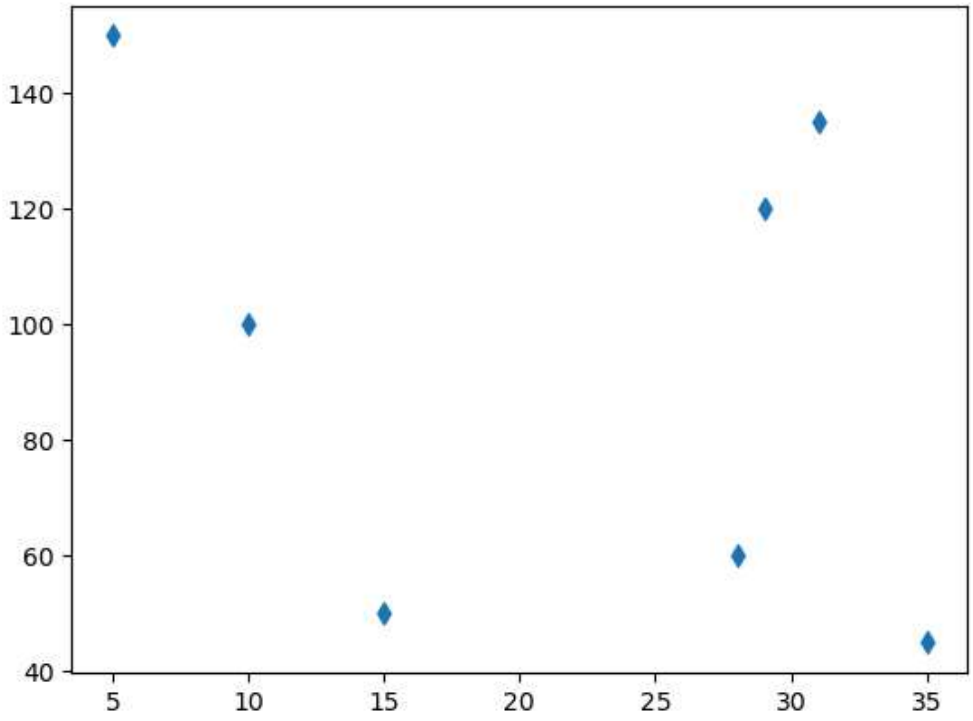
In [433]:
```
plt.barh([10,5,20],[100,150,200],height=0.5,color='b')
```

Out[433]:
```
<BarContainer object of 3 artists>
```



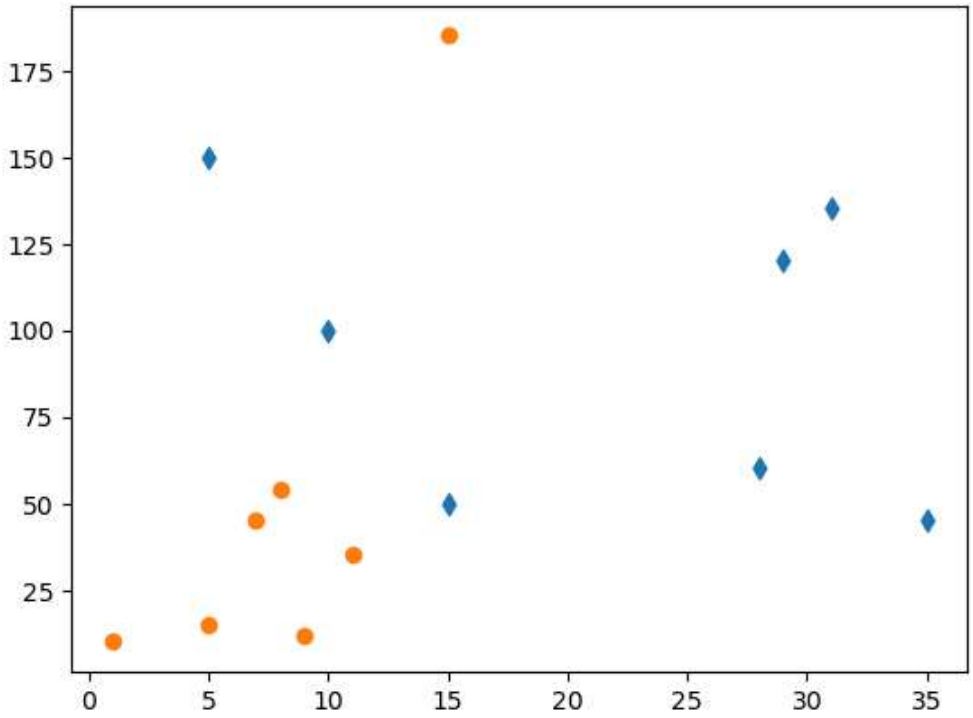
In [435]:
```
plt.scatter([10,5,15,28,29,31,35],[100,150,50,60,120,135,45],marker='d')
```

Out[435]:
```
<matplotlib.collections.PathCollection at 0x1c2a2101d50>
```



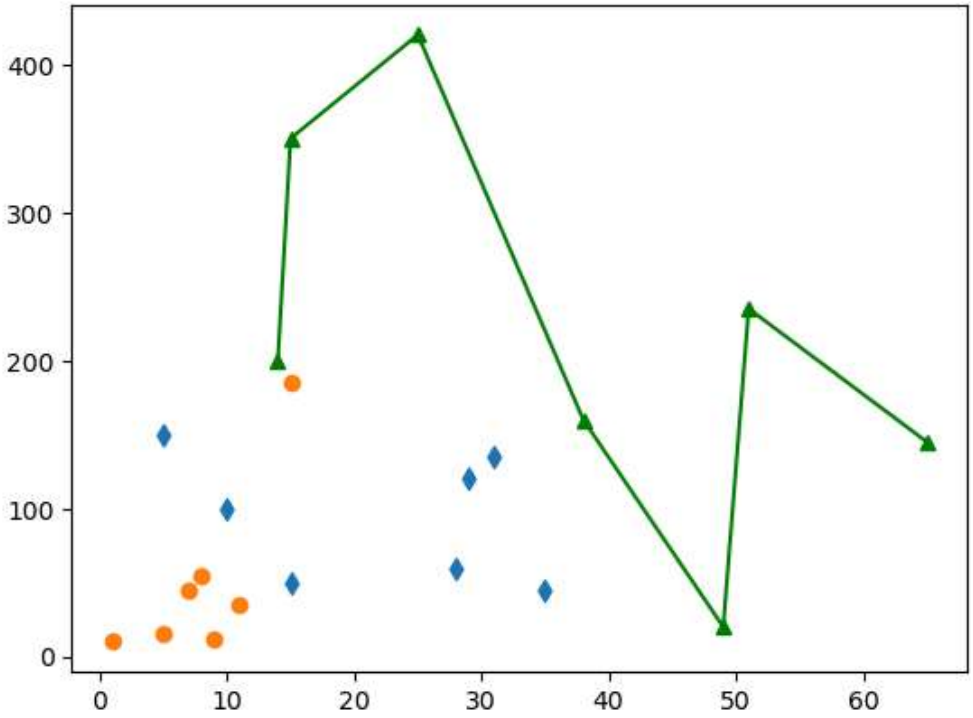
In [437]:
```
plt.scatter([10,5,15,28,29,31,35],[100,150,50,60,120,135,45],marker='d')
plt.scatter([1,5,7,8,9,11,15],[10,15,45,54,12,35,185],marker='o')
```

Out[437]:
```
<matplotlib.collections.PathCollection at 0x1c2a21c43d0>
```



In [440]:
```
plt.scatter([10,5,15,28,29,31,35],[100,150,50,60,120,135,45],marker='d')
plt.scatter([1,5,7,8,9,11,15],[10,15,45,54,12,35,185],marker='o')
plt.plot([14,15,25,38,49,51,65],[200,350,420,160,20,235,145],marker='^',color='g')
```

Out[440]:
```
[<matplotlib.lines.Line2D at 0x1c2a2841510>]
```



In [ ]:
```
## Task
    |
load home.csv file
  |
  |-> info about home.csv
      - duplicated ->drop duplicates
      - replace null values
      - select any two cols - X , Y
                              |
                        matplotlib - Visualization
```

In [441]:
```
pd.read_csv('E:\\home.csv')
```

Out[441]:

	month	water	telephone	eb	gas	provision	fuel	internet	academic	external
0	jan	200	1300	505	930	8900	1600	1700	35000	5000
1	feb	200	1300	125	930	4000	1800	1710	55000	9000
2	mar	200	1350	505	930	5000	1600	1700	500	5000
3	apr	600	1365	125	930	3000	1800	1710	15000	9000
4	may	700	1234	65	850	10000	1800	1700	6000	5000
5	jun	800	1700	88	890	7000	1600	1710	200	15000
6	jul	200	1783	0	930	8900	1700	1700	100	5000
7	aug	150	1100	1235	780	6000	1960	1710	500	9000
8	sep	200	1230	1500	760	4900	1650	1700	600	5000
9	oct	200	1323	70	789	4600	1500	1710	700	20000
10	nov	200	1231	656	880	6900	1900	1800	500	5000
11	dec	200	1100	100	930	8000	1200	1710	1000	10000

In [442]:
```
home_df = pd.read_csv('E:\\home.csv')
home_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   month      12 non-null     object
 1   water      12 non-null     int64 
 2   telephone  12 non-null     int64 
 3   eb         12 non-null     int64 
 4   gas        12 non-null     int64 
 5   provision  12 non-null     int64 
 6   fuel       12 non-null     int64 
 7   internet   12 non-null     int64 
 8   academic   12 non-null     int64 
 9   external   12 non-null     int64 
dtypes: int64(9), object(1)
memory usage: 1.1+ KB
```

In [443]:
```
home_df.index
```

Out[443]:
```
RangeIndex(start=0, stop=12, step=1)
```

In [444]:
```
home_df.columns
```

Out[444]:
```
Index(['month', 'water', 'telephone', 'eb', 'gas', 'provision', 'fuel',
       'internet', 'academic', 'external'],
      dtype='object')
```

In [446]:
```
home_df.head(3)
```

Out[446]:

	month	water	telephone	eb	gas	provision	fuel	internet	academic	external
0	jan	200	1300	505	930	8900	1600	1700	35000	5000
1	feb	200	1300	125	930	4000	1800	1710	55000	9000
2	mar	200	1350	505	930	5000	1600	1700	500	5000

In [447]:
```
home_df.duplicated().any()
```

Out[447]:
```
False
```

In [448]:
```
home_df.isnull().any()
```

Out[448]:
```
month        False
water        False
telephone    False
eb           False
gas          False
provision    False
fuel         False
internet     False
academic     False
external     False
dtype: bool
```

In [449]:
```
home_df.rename(columns={'telephone':'tele','provision':'pro','internet':'net'},inplace=True
```

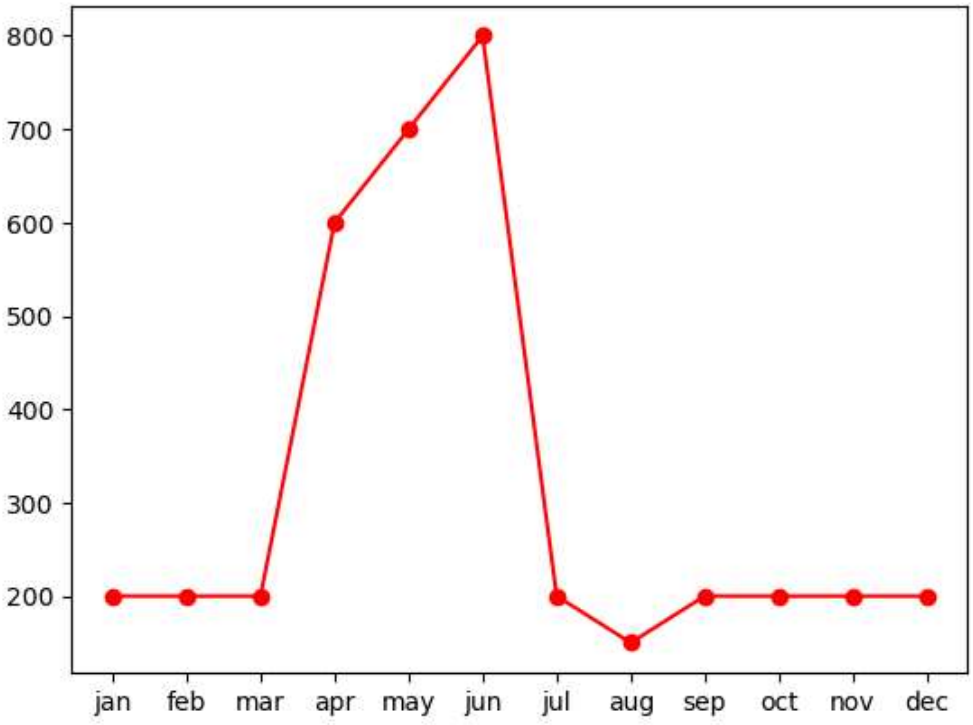
In [450]:
```
home_df.columns
```

Out[450]:
```
Index(['month', 'water', 'tele', 'eb', 'gas', 'pro', 'fuel', 'net', 'academic',
       'external'],
      dtype='object')
```

In [452]:
```
X = home_df['month']
Y = home_df['water']
```

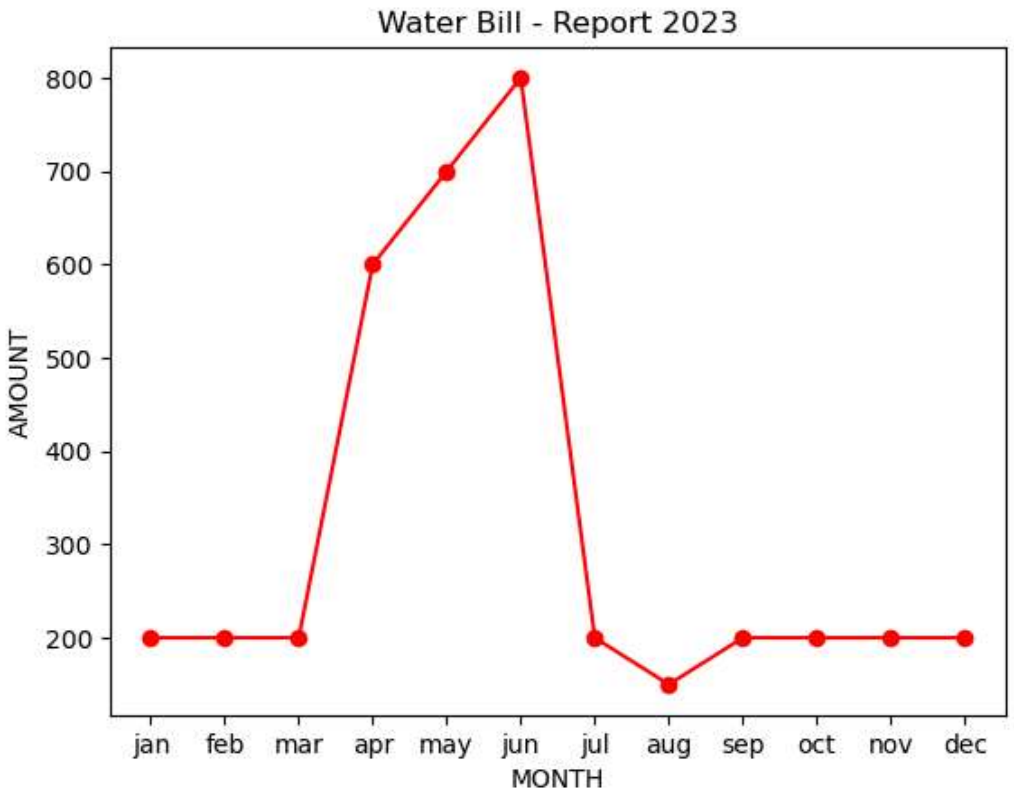
In [453]:
```
plt.plot(X,Y,marker='o',color='r')
```

Out[453]:
```
[<matplotlib.lines.Line2D at 0x1c2a29452d0>]
```



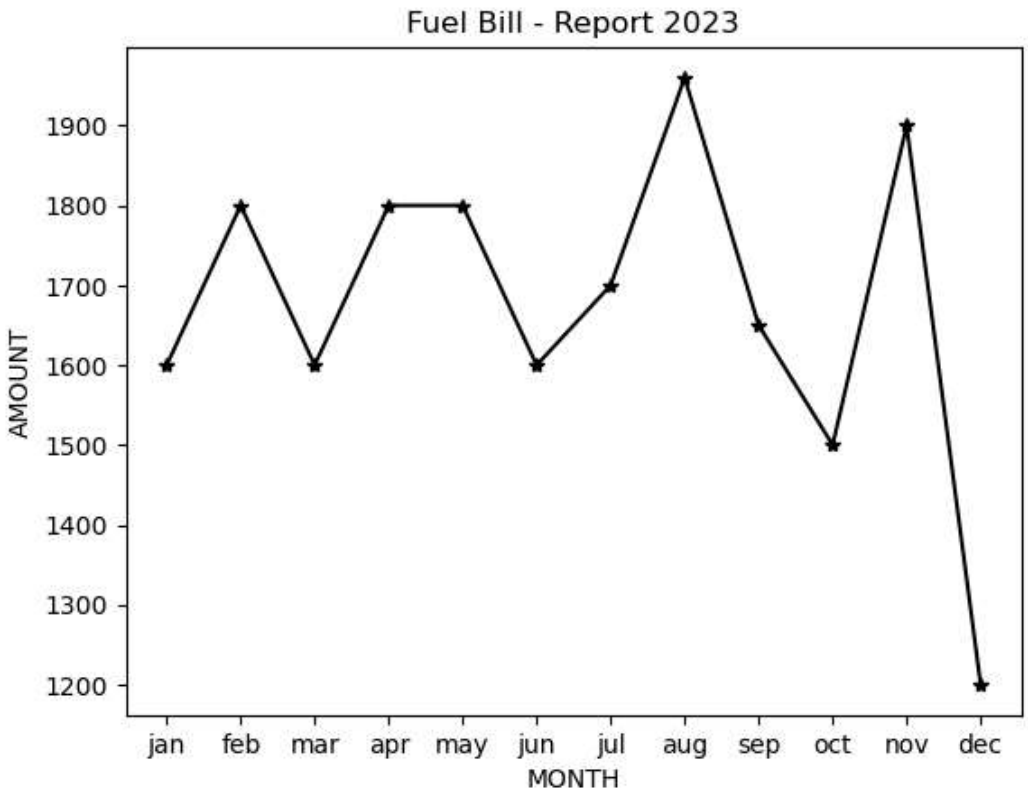
In [454]:
```
plt.plot(X,Y,marker='o',color='r')
plt.xlabel('MONTH')
plt.ylabel('AMOUNT')
plt.title('Water Bill - Report 2023')
```

Out[454]: Text(0.5, 1.0, 'Water Bill - Report 2023')



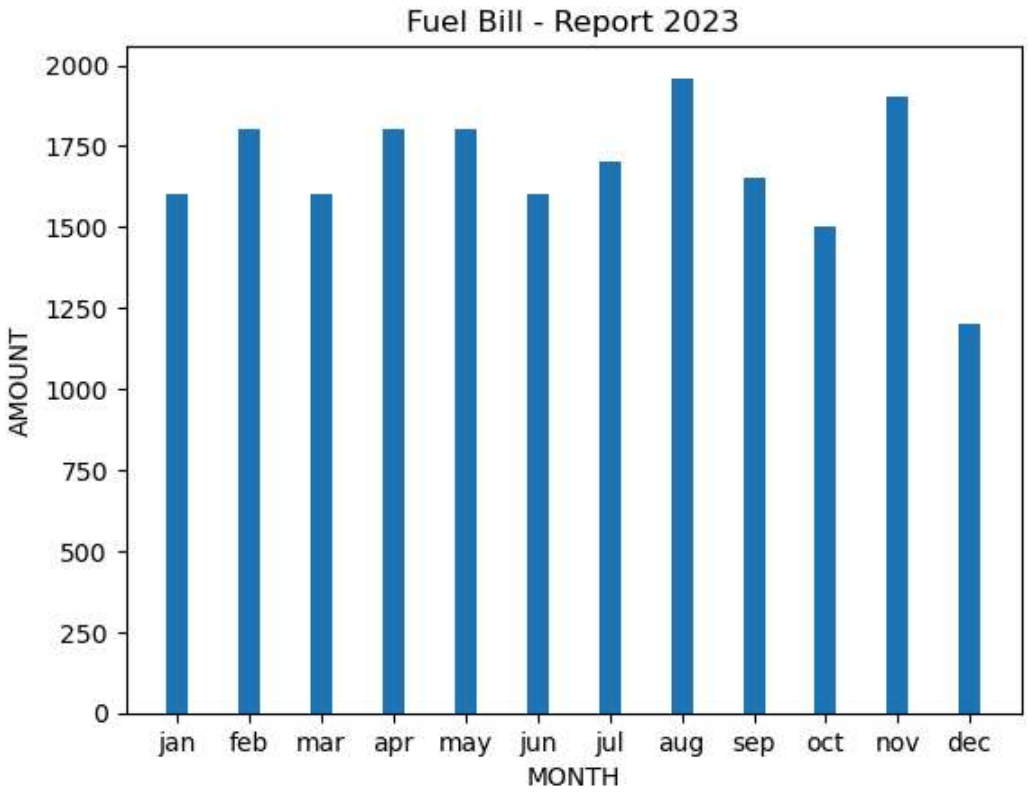
In [455]:
```
X = home_df['month']
Y = home_df['fuel']
plt.plot(X,Y,marker='*',color='black')
plt.xlabel('MONTH')
plt.ylabel('AMOUNT')
plt.title('Fuel Bill - Report 2023')
```

Out[455]: Text(0.5, 1.0, 'Fuel Bill - Report 2023')



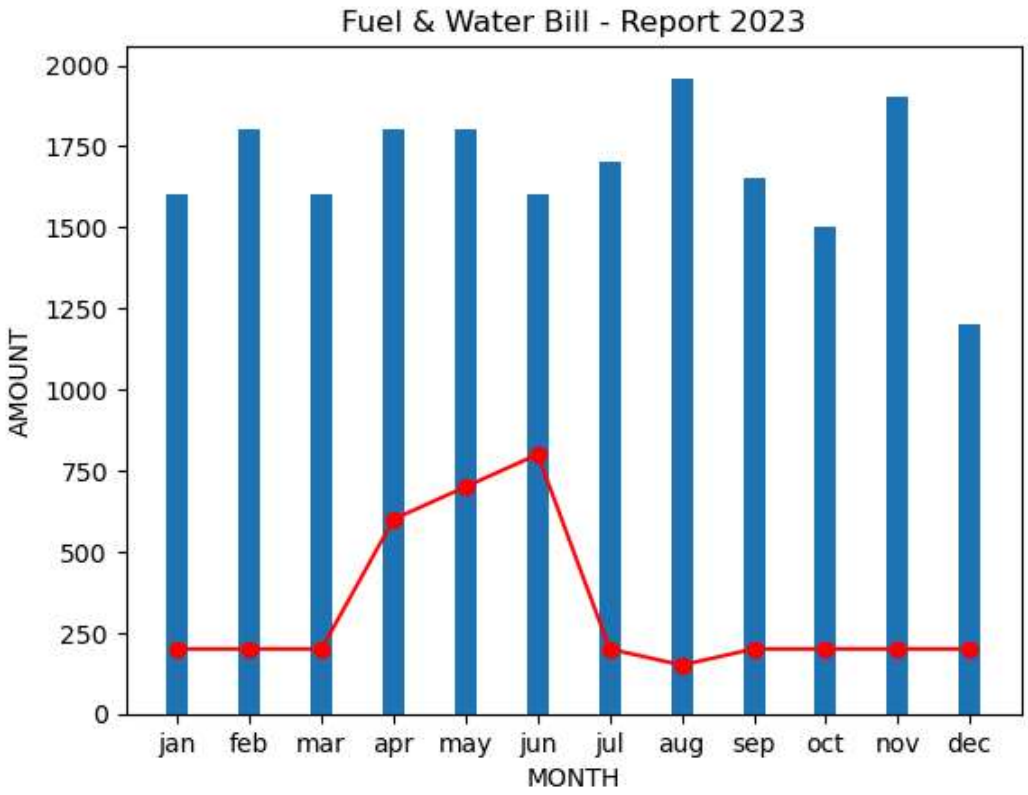
In [456]:
```
X = home_df['month']
Y = home_df['fuel']

plt.xlabel('MONTH')
plt.ylabel('AMOUNT')
plt.title('Fuel Bill - Report 2023')
plt.bar(X,Y,width=0.3)
```

Out[456]: <BarContainer object of 12 artists>



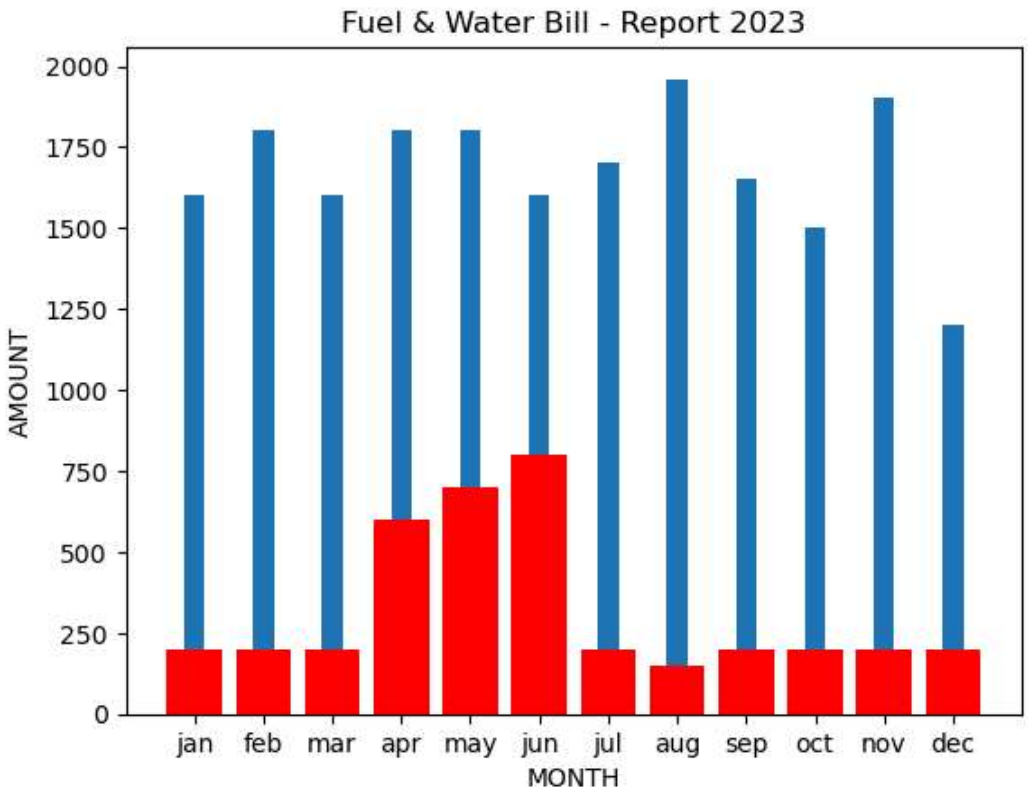
In [458]:
```
X1 = home_df['month']
Y1 = home_df['fuel']

Y2 = home_df['water']
plt.xlabel('MONTH')
plt.ylabel('AMOUNT')
plt.title('Fuel & Water Bill - Report 2023')
plt.bar(X1,Y1,width=0.3)
plt.plot(X1,Y2,marker='o',color='r')
```

Out[458]: [<matplotlib.lines.Line2D at 0x1c2a29f5110>]



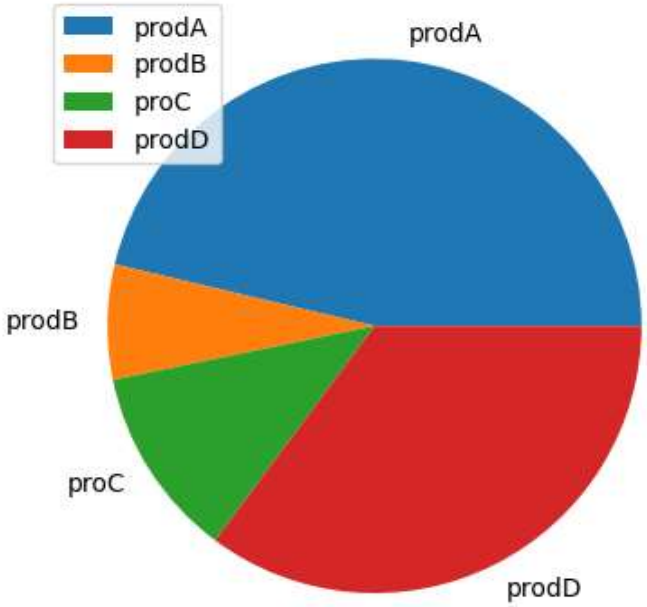
In [466]:
```
X1 = home_df['month']
Y1 = home_df['fuel']

Y2 = home_df['water']
plt.xlabel('MONTH')
plt.ylabel('AMOUNT')
plt.title('Fuel & Water Bill - Report 2023')
plt.bar(X1,Y1,width=0.3)
plt.bar(X1,Y2,color='r')
#plt.legend('Fuel','water')
```

Out[466]: <BarContainer object of 12 artists>



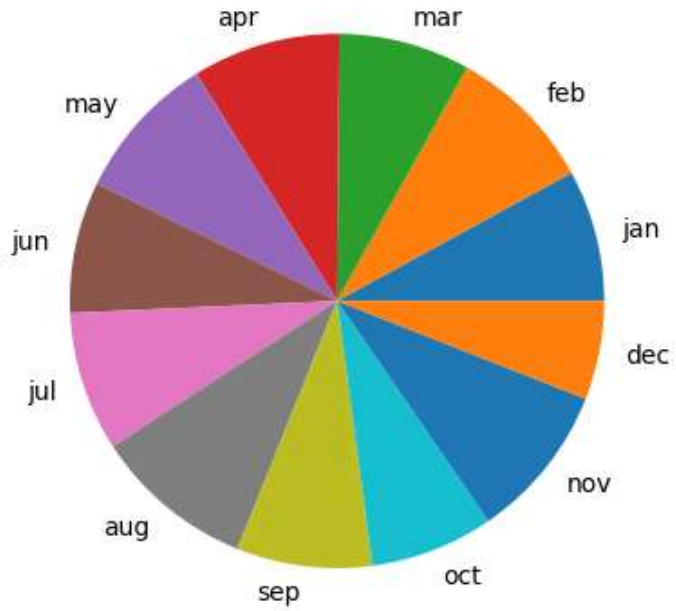
In [469]:
```
plt.pie([100,15,25,76],labels=['prodA','prodB','proC','prodD'])
plt.legend()
```

Out[469]: <matplotlib.legend.Legend at 0x1c2a3365f50>



In [475]:
```
plt.pie(home_df['fuel'],labels=home_df['month'])
```

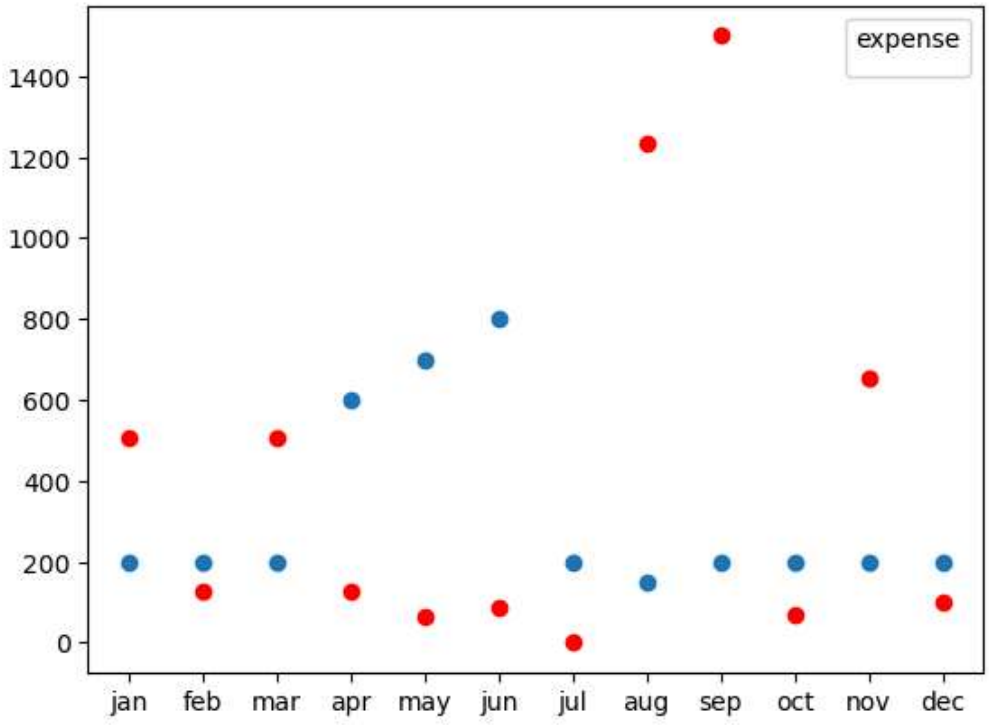
Out[475]:
```
([<matplotlib.patches.Wedge at 0x1c2a5152490>,
  <matplotlib.patches.Wedge at 0x1c2a5153f50>,
  <matplotlib.patches.Wedge at 0x1c2a5160e50>,
  <matplotlib.patches.Wedge at 0x1c2a5150f50>,
  <matplotlib.patches.Wedge at 0x1c2a5162b10>,
  <matplotlib.patches.Wedge at 0x1c2a5163ad0>,
  <matplotlib.patches.Wedge at 0x1c2a5170b50>,
  <matplotlib.patches.Wedge at 0x1c2a5171c10>,
  <matplotlib.patches.Wedge at 0x1c2a5162c10>,
  <matplotlib.patches.Wedge at 0x1c2a5173810>,
  <matplotlib.patches.Wedge at 0x1c2a517c710>,
  <matplotlib.patches.Wedge at 0x1c2a517d050>],
 [Text(1.0658165455689068, 0.27209390142295065, 'jan'),
  Text(0.7811518427918157, 0.774468720157858, 'feb'),
  Text(0.2812414266201684, 1.0634393541489109, 'mar'),
  Text(-0.29616482095425317, 1.0593801955998305, 'apr'),
  Text(-0.8154247489778412, 0.7382970125595963, 'may'),
  Text(-1.0770456525246412, 0.22354566061046668, 'jun'),
  Text(-1.0472717152449498, -0.3364847016550691, 'jul'),
  Text(-0.6986211069955461, -0.8496637857766551, 'aug'),
  Text(-0.13626393318651675, -1.0915274346128643, 'sep'),
  Text(0.3956213890977273, -1.0263935485418763, 'oct'),
  Text(0.8610085548318517, -0.6845905845878735, 'nov'),
  Text(1.0807279031662669, -0.20500536411968395, 'dec')])
```



In [513]:
```
plt.scatter(home_df['month'],home_df['eb'],c='r')
plt.scatter(home_df['month'],home_df['water'])
plt.legend(title='expense')
```

```
No artists with labels found to put in legend.  Note that artists whose label start with a
n underscore are ignored when legend() is called with no argument.
```

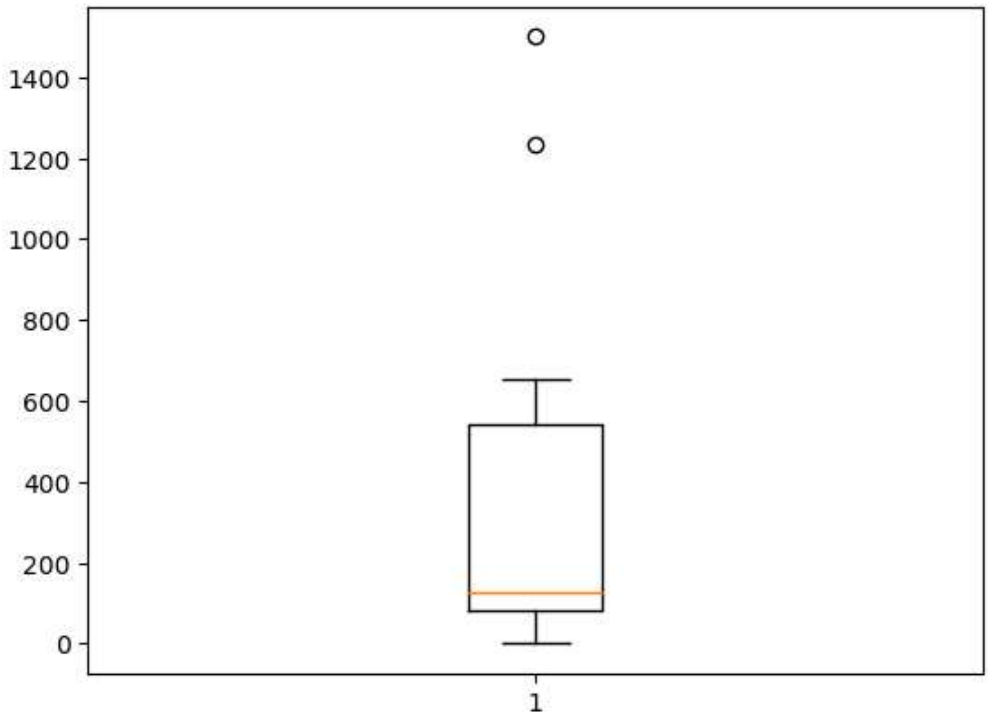
Out[513]:
```
<matplotlib.legend.Legend at 0x1c2aa1b7390>
```



In [482]:
```
print(home_df['eb'])
plt.boxplot(home_df['eb'])
```

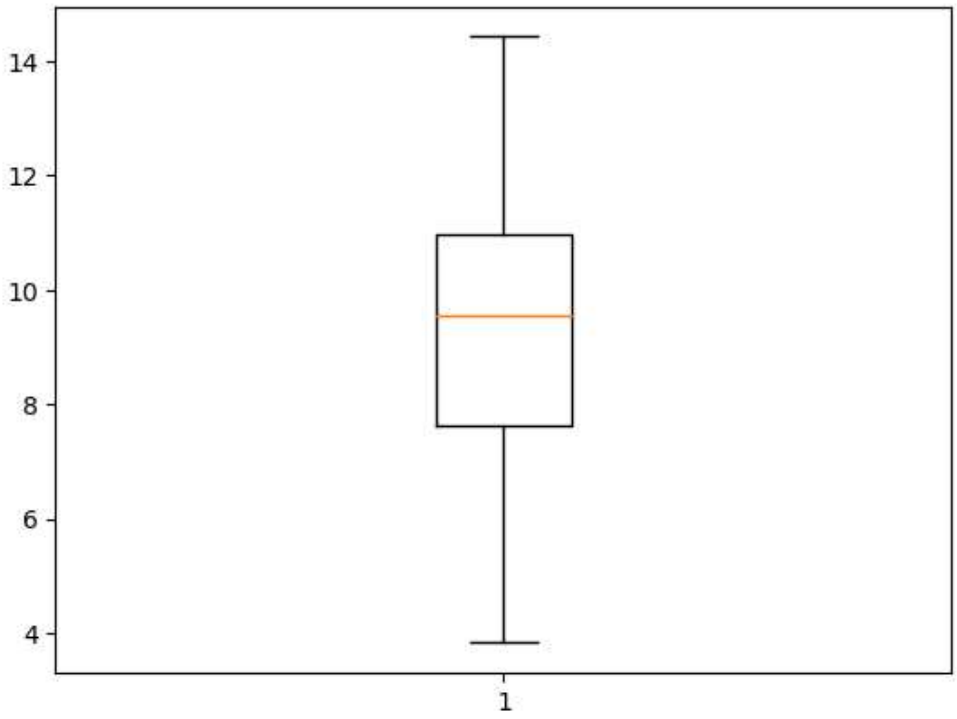
```
0      505
1      125
2      505
3      125
4       65
5       88
6        0
7     1235
8     1500
9       70
10     656
11     100
Name: eb, dtype: int64
```

Out[482]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1c2a53c7150>,
  <matplotlib.lines.Line2D at 0x1c2a53c7b90>],
 'caps': [<matplotlib.lines.Line2D at 0x1c2a53d4510>,
  <matplotlib.lines.Line2D at 0x1c2a53d4c90>],
 'boxes': [<matplotlib.lines.Line2D at 0x1c2a53c6850>],
 'medians': [<matplotlib.lines.Line2D at 0x1c2a53d5490>],
 'fliers': [<matplotlib.lines.Line2D at 0x1c2a5123010>],
 'means': []}
```



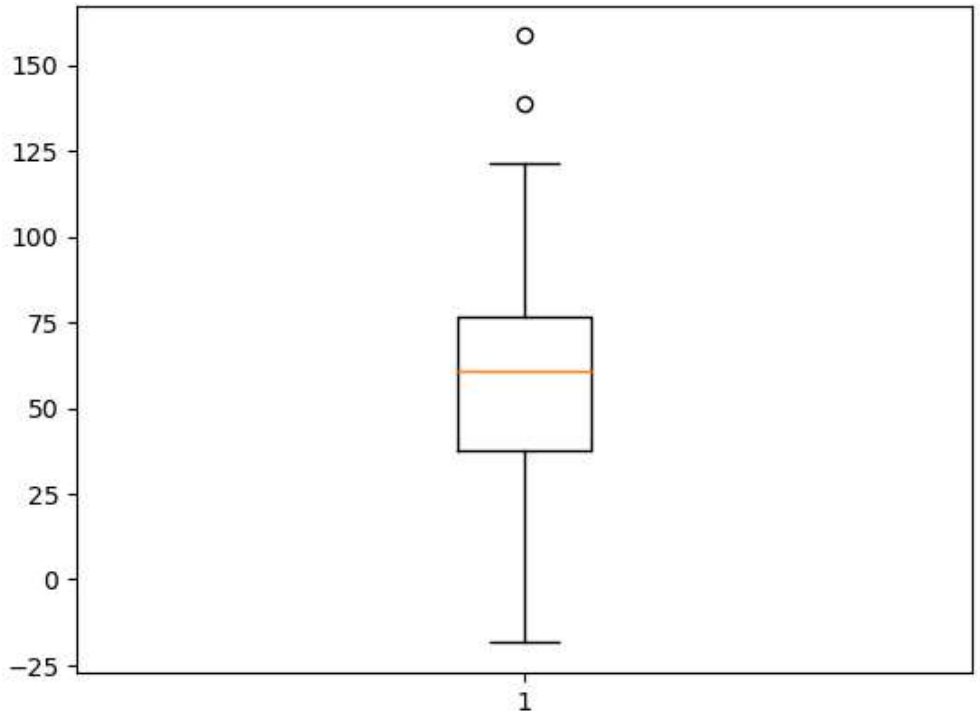
In [486]:
```
#help(np.random.normal)
data_set = np.random.normal(10,3,20)
plt.boxplot(data_set)
```

Out[486]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1c2a54f60d0>,
  <matplotlib.lines.Line2D at 0x1c2a55f2910>],
 'caps': [<matplotlib.lines.Line2D at 0x1c2a55f1050>,
  <matplotlib.lines.Line2D at 0x1c2a55f3e90>],
 'boxes': [<matplotlib.lines.Line2D at 0x1c2a55f1250>],
 'medians': [<matplotlib.lines.Line2D at 0x1c2a5600ad0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1c2a55e1510>],
 'means': []}
```



In [487]:
```
data_set = np.random.normal(60,30,200)
plt.boxplot(data_set)
```

Out[487]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1c2a5660b50>,
  <matplotlib.lines.Line2D at 0x1c2a56617d0>],
 'caps': [<matplotlib.lines.Line2D at 0x1c2a5662310>,
  <matplotlib.lines.Line2D at 0x1c2a5662e10>],
 'boxes': [<matplotlib.lines.Line2D at 0x1c2a5657e90>],
 'medians': [<matplotlib.lines.Line2D at 0x1c2a56639d0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1c2a547f090>],
 'means': []}
```

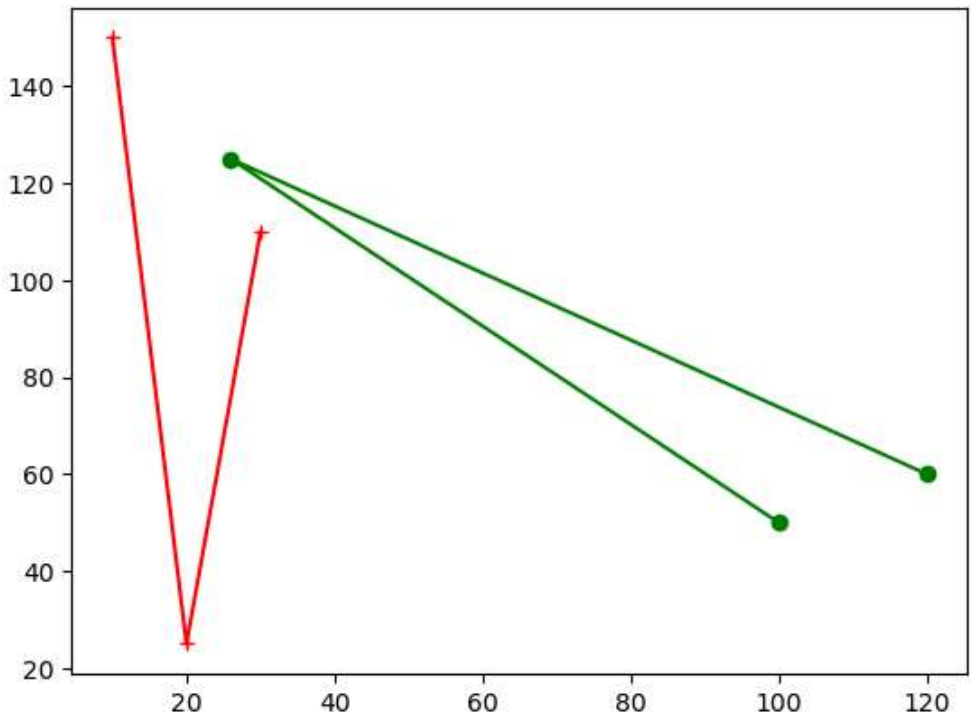


In [ ]:
```
10 20 30 40 50 600 70
                |->outlier
```

In [ ]:
```
plt.show()
```

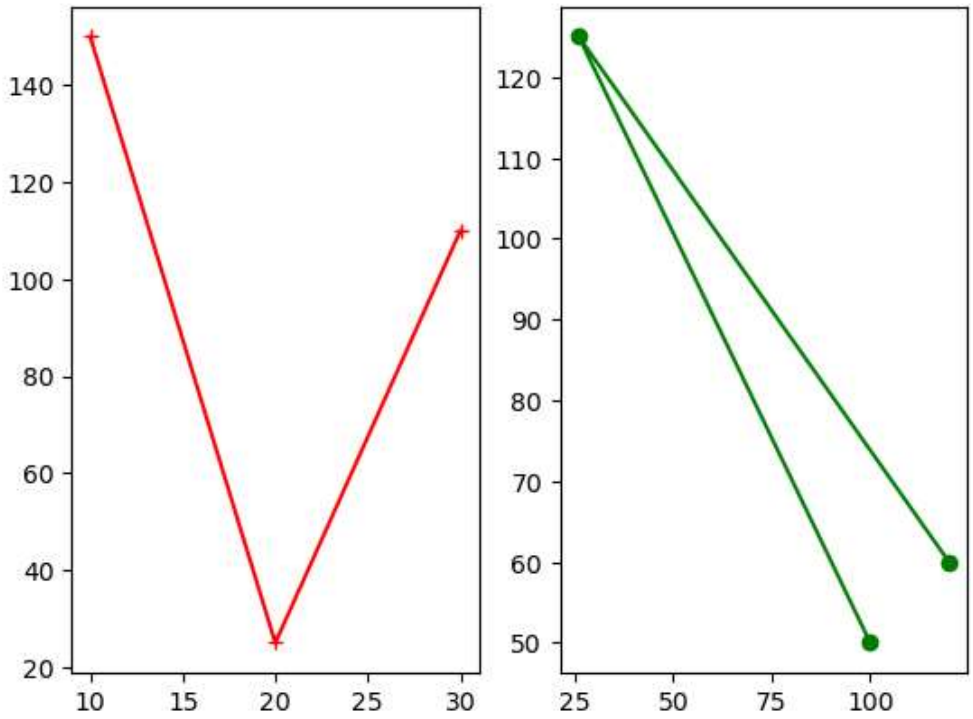
In [489]:
```
X1 = np.array([10,20,30])
Y1 = np.array([150,25,110])
X2 = np.array([100,26,120])
Y2 = np.array([50,125,60])

plt.plot(X1,Y1,marker='+',c='r')
plt.plot(X2,Y2,marker='o',c='g')
```

Out[489]: [<matplotlib.lines.Line2D at 0x1c2a686de90>]



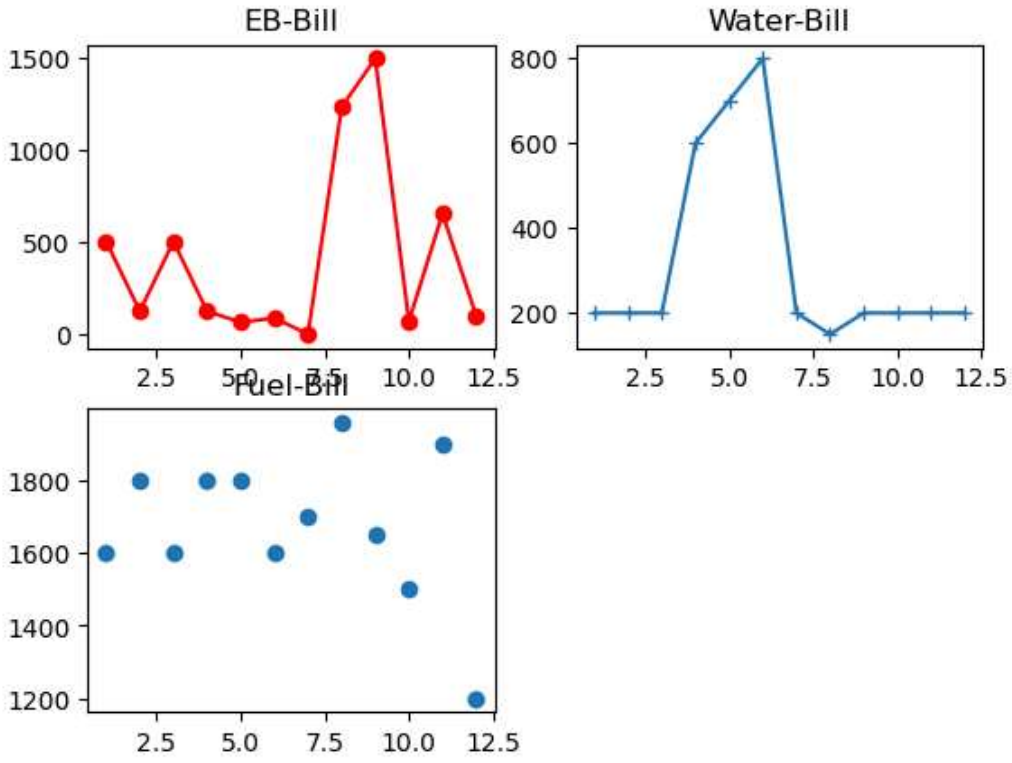
In [492]:
```
X1 = np.array([10,20,30])
Y1 = np.array([150,25,110])

plt.subplot(1,2,1) # plt.subplot(121)
plt.plot(X1,Y1,marker='+',c='r')

X2 = np.array([100,26,120])
Y2 = np.array([50,125,60])
plt.subplot(1,2,2)
plt.plot(X2,Y2,marker='o',c='g')
```

Out[492]: [<matplotlib.lines.Line2D at 0x1c2a6ae8290>]



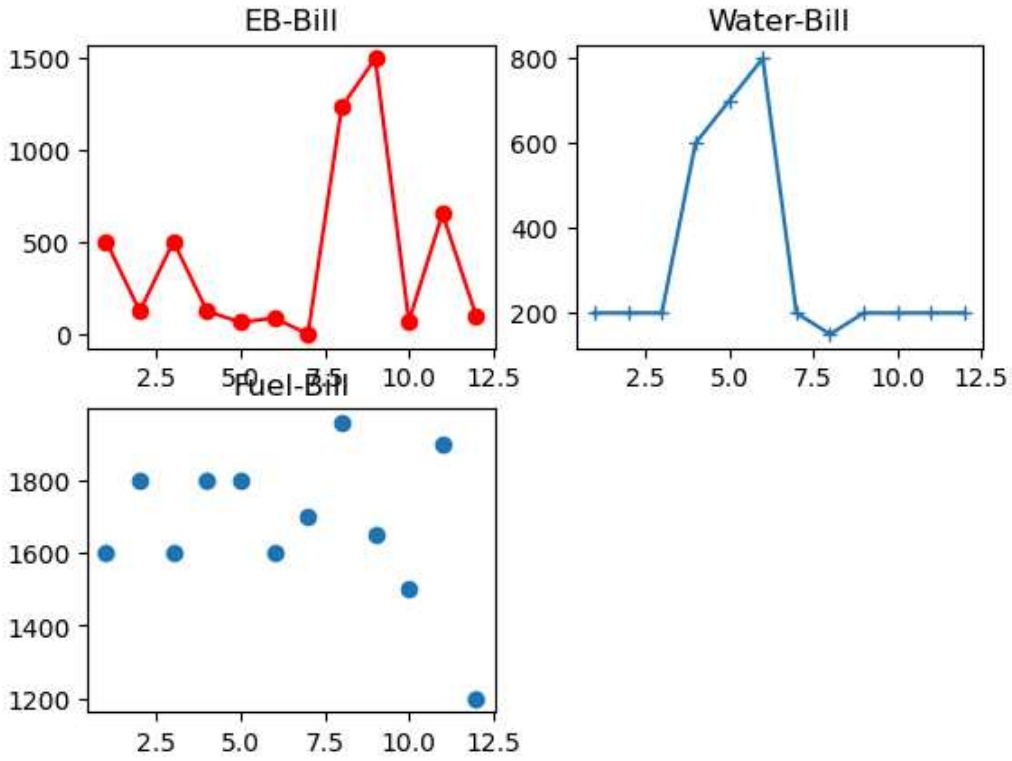
In [502]:
```
L=list(range(1,13))
plt.subplot(2,2,1) # 221
plt.plot(L,home_df['eb'],marker='o',c='r')
plt.title('EB-Bill')

plt.subplot(2,2,2)
plt.plot(L,home_df['water'],marker='+')
plt.title('Water-Bill')

plt.subplot(2,2,3)
plt.scatter(L,home_df['fuel'])
plt.title('Fuel-Bill')
```

Out[502]:
```
Text(0.5, 1.0, 'Fuel-Bill')
```



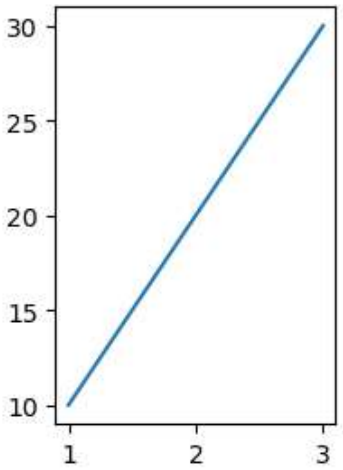
In [503]:
```
L=list(range(1,13))
plt.subplot(2,2,1) # 221
plt.plot(L,home_df['eb'],marker='o',c='r')
plt.title('EB-Bill')

plt.subplot(2,2,2)
plt.plot(L,home_df['water'],marker='+')
plt.title('Water-Bill')

plt.subplot(2,2,3)
plt.scatter(L,home_df['fuel'])
plt.title('Fuel-Bill')
```

Out[503]:
```
Text(0.5, 1.0, 'Fuel-Bill')
```



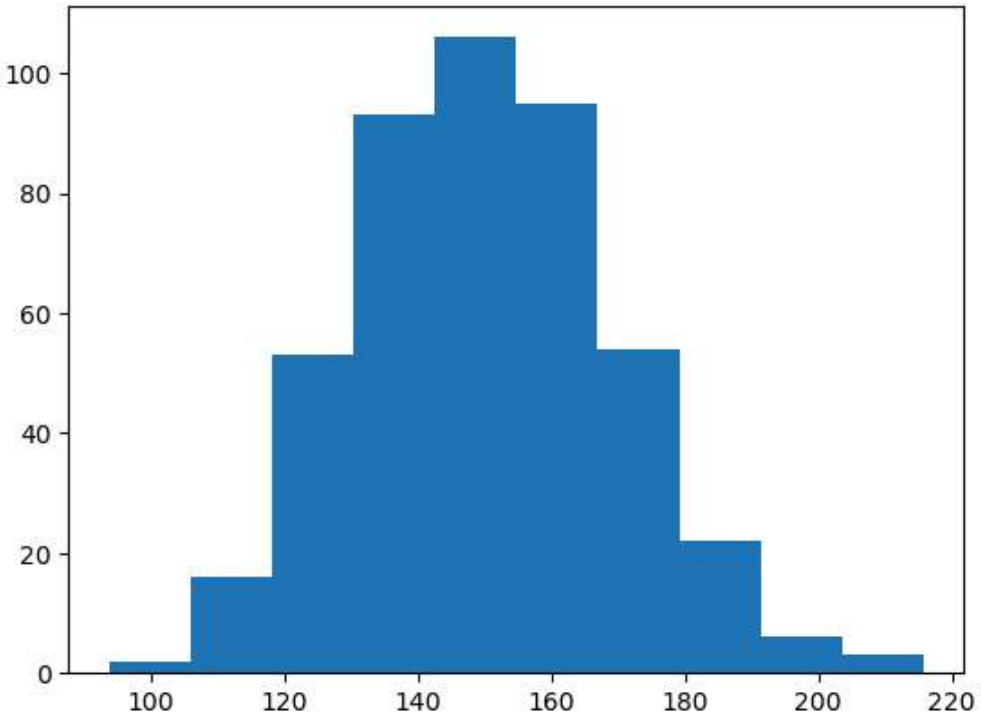
In [507]:
```
plt.figure(figsize=(2,3))
plt.plot([1,2,3],[10,20,30])
```

Out[507]:
```
[<matplotlib.lines.Line2D at 0x1c2a8db53d0>]
```



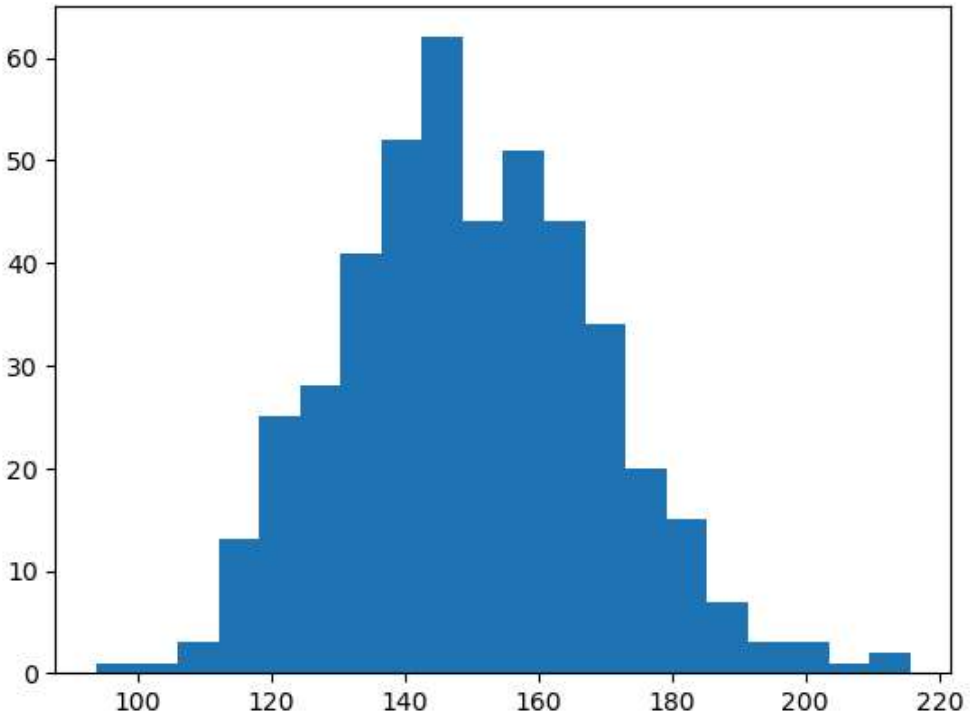
In [509]:
```
# Histogram
http_request_duration = np.random.normal(150,20,450)

plt.hist(http_request_duration)
```

Out[509]:
```
(array([  2.,  16.,  53.,  93., 106.,  95.,  54.,  22.,   6.,   3.]),
 array([ 93.88618345, 106.06804907, 118.24991469, 130.43178031,
        142.61364593, 154.79551155, 166.97737717, 179.15924279,
        191.34110841, 203.52297403, 215.70483965]),
 <BarContainer object of 10 artists>)
```



In [510]:
```
plt.hist(http_request_duration,bins=20)
```

Out[510]:
```
(array([ 1.,  1.,  3., 13., 25., 28., 41., 52., 62., 44., 51., 44., 34.,
        20., 15.,  7.,  3.,  3.,  1.,  2.]),
 array([ 93.88618345,  99.97711626, 106.06804907, 112.15898188,
        118.24991469, 124.3408475 , 130.43178031, 136.52271312,
        142.61364593, 148.70457874, 154.79551155, 160.88644436,
        166.97737717, 173.06830998, 179.15924279, 185.2501756 ,
        191.34110841, 197.43204122, 203.52297403, 209.61390684,
        215.70483965]),
 <BarContainer object of 20 artists>)
```



In [512]:
```
plt.hist(http_request_duration,bins=20)
plt.xlabel('req-values')
plt.ylabel('density-req')
```

Out[512]:
```
Text(0, 0.5, 'density-req')
```