```
[ ] #Gọi thư viện
    import pandas as pd
    import numpy as np
```

```
[ ] #đọc dữ liệu vào data frame df
    df = pd.read_csv("/content/drive/MyDrive/Data Analys/sinhvien.csv")
```

```
#Show file
df
```

	id	name	mark	gender
0	1	A	5	M
1	2	B	6	F
2	3	A	7	M
3	4	B	6	M
4	5	A	5	F
5	6	B	4	M
6	7	A	3	M
7	8	B	8	F
8	9	A	9	M

```
#Ghi dữ liệu vào file csv khác => tự sinh ra 1 file mới, copy nội dung file cũ
df.to_csv("/content/drive/MyDrive/Data Analys/newStudent.csv")
```

```
[ ] #xem thông tin file csv
    df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      30 non-null     int64 
 1   name    30 non-null     object
 2   mark    30 non-null     int64 
 3   gender  30 non-null     object
dtypes: int64(2), object(2)
memory usage: 1.1+ KB
```

```
[ ] #đọc dữ liệu trên các Series
    df["name"]
    df["mark"]
    df["gender"]
```

```
0     M
1     F
2     M
3     M
```

```
[ ] #đọc dữ liệu trên 1 tập series
    df[["name","mark"]]
```

	name	mark
0	A	5
1	B	6
2	A	7
3	B	6
4	A	5
5	B	4

```
#Lấy dữ liệu trên từng hàng (Row)
df.head(0)   #Dòng tiêu đề
```

id	name	mark	gender

```
df.head(1)  #dòng đầu tiên là 1
```

	id	name	mark	gender
0	1	A	5	M

```
#Lấy 1 số bản ghi cuối cùng trong frame, ví dụ lấy 5 bản ghi cuối
df.tail(5)
```

	id	name	mark	gender
25	26	B	9	M
26	27	A	8	M
27	28	B	8	M
28	29	A	8	M
29	30	B	7	M

```
[ ]  #Lấy 1 số bản ghi từ a -> b trong data frame
     df[0:5]
```

	id	name	mark	gender
0	1	A	5	M
1	2	B	6	F
2	3	A	7	M
3	4	B	6	M
4	5	A	5	F

```
[ ]  #Lọc dữ liệu theo điều kiện
     df[df.gender == 'M']
```

	id	name	mark	gender
0	1	A	5	M
2	3	A	7	M
3	4	B	6	M
5	6	B	4	M
6	7	A	3	M

```
df[df.mark<5]
```

	id	name	mark	gender
5	6	B	4	M
6	7	A	3	M
14	15	A	4	M
15	16	B	3	M
16	17	A	3	F
17	18	B	2	F
18	19	A	2	F
19	20	B	3	F
20	21	A	4	M

```
#other
df[df['gender']=='M']
```

	id	name	mark	gender
0	1	A	5	M
2	3	A	7	M

```
#Kết hợp nhiều điều kiện
df[(df.gender=='M')&(df.mark>5)]
```

	id	name	mark	gender
2	3	A	7	M
3	4	B	6	M
8	9	A	9	M
9	10	B	9	M

```
[ ] #Muốn lấy kết quả lọc lưu vào 1 data frame khác để xử lý sau này, có thể lưu vào 1 file
    malestudents = df[(df.gender=='M')&(df.mark>5)]
```

```
[ ] malestudents
```

	id	name	mark	gender
2	3	A	7	M
3	4	B	6	M
8	9	A	9	M
9	10	B	9	M

```
[ ] #Thêm sửa xóa cột trên data frame
    #Thêm 1 cột doublemark = mark*2 trên data frame
    #dữ liệu này nếu muốn thay đổi thực sự thì cần lưu đè lên data frame student.csv
    df['doublemark'] = df['mark']*2
```

+ Mã + Văn bản

```
[ ] df
```

	id	name	mark	gender	doublemark
0	1	A	5	M	10
1	2	B	6	F	12
2	3	A	7	M	14
3	4	B	6	M	12
4	5	A	5	F	10
5	6	B	4	M	8

```
[ ] #Thêm 1 cột pass, nếu mark >=5 thì 1 ngược lại 0
    df['pass']=np.where(df['mark']>=5,1,0)
```

```
[ ] df
```

	id	name	mark	gender	doublemark	pass
0	1	A	5	M	10	1
1	2	B	6	F	12	1

```
[ ]  #Xóa 1 cột, ví dụ xóa cột doubleMark
     mydf = df.drop(columns='doublemark')
```

```
[ ]  mydf
```

	id	name	mark	gender	pass
0	1	A	5	M	1
1	2	B	6	F	1
2	3	A	7	M	1
3	4	B	6	M	1
4	5	A	5	F	1
5	6	B	4	M	0
6	7	A	3	M	0

```
[ ]  #Xóa luôn không cần gán lại, sử dụng
     df.drop(columns='doublemark',inplace=True)
```

```
#Xóa hàng, xóa theo index, nhưng phải gán vào 1 biến, nếu không gán thì sẽ không lưu xóa
df.drop(index=[2,3,4])
```

	id	name	mark	gender	pass
0	1	A	5	M	1
1	2	B	6	F	1
5	6	B	4	M	0
6	7	A	3	M	0
7	8	B	8	F	1

```
[ ]  df
```

	id	name	mark	gender	pass
0	1	A	5	M	1
1	2	B	6	F	1
2	3	A	7	M	1
3	4	B	6	M	1
4	5	A	5	F	1
5	6	B	4	M	0
6	7	A	3	M	0
7	8	B	8	F	1

```
[ ]  df = df.drop(index=[2,3,4])
```

```
[ ]  df
```

	id	name	mark	gender	pass
0	1	A	5	M	1
1	2	B	6	F	1
5	6	B	4	M	0
6	7	A	3	M	0
7	8	B	8	F	1
8	9	A	9	M	1

```
#Sắp xếp, muốn thay đổi trên bảng gốc thì phải gán lại
#mặc định tăng dần
df.sort_values(by=['mark'])
```

	id	name	mark	gender	pass
18	19	A	2	F	0
17	18	B	2	F	0
16	17	A	3	F	0
6	7	A	3	M	0
19	20	B	3	F	0

```
#Sắp xếp theo chiều giảm gần
df.sort_values(by=['mark'],ascending=False)
```

	id	name	mark	gender	pass
8	9	A	9	M	1
9	10	B	9	M	1
25	26	B	9	M	1
28	29	A	8	M	1
27	28	B	8	M	1
7	8	B	8	F	1
10	11	A	8	M	1

```
#Sắp xếp nhiều cột
df.sort_values(by=['mark','gender'],ascending=False)
```

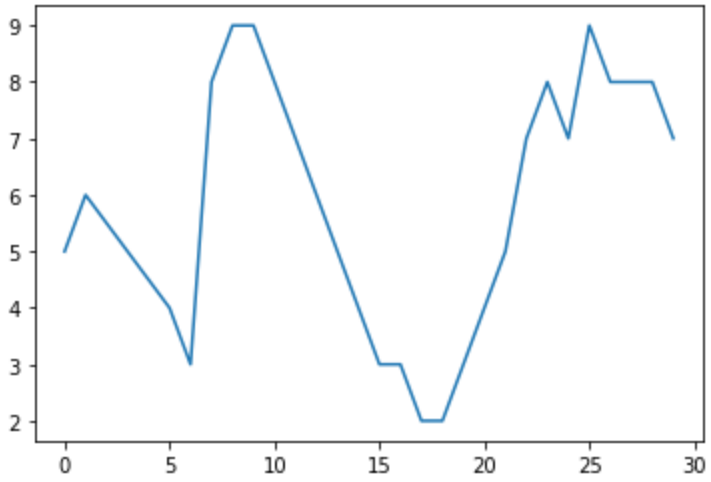
	id	name	mark	gender	pass
8	9	A	9	M	1
9	10	B	9	M	1
25	26	B	9	M	1
10	11	A	8	M	1
23	24	B	8	M	1
26	27	A	8	M	1

```
#Thống kê cho các cột số, std: độ lệch chuẩn, 25%,50%,75%: phân vị, ...
df.describe()
```

	id	mark	pass
count	27.000000	27.000000	27.000000
mean	16.777778	5.851852	0.666667
std	8.331282	2.298891	0.480384
min	1.000000	2.000000	0.000000
25%	10.500000	4.000000	0.000000
50%	17.000000	6.000000	1.000000
75%	23.500000	8.000000	1.000000
max	30.000000	9.000000	1.000000

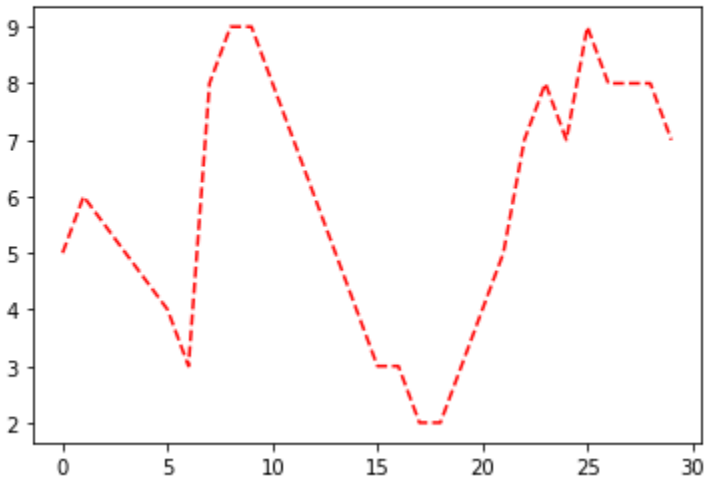
```
#Vẽ biểu đồ thống kê
df['mark'].plot.line()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7c43126e10>
```



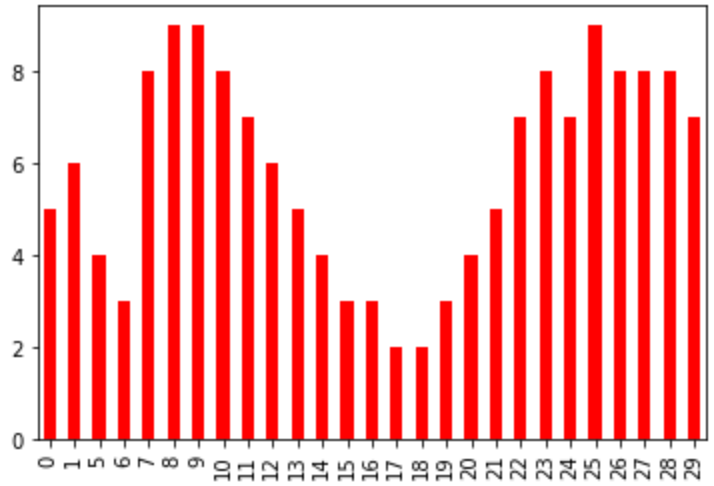

```
[ ] #có thể truyền màu, style cho biểu đồ
df['mark'].plot.line(color='red',linestyle='--')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7c423ba510>
```



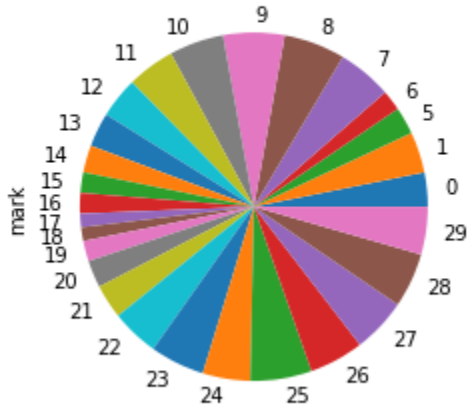
```
[ ]  df['mark'].plot.bar(color='red',linestyle='--')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7c423b09d0>
```




```
df['mark'].plot.pie()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7c422a86d0>
```

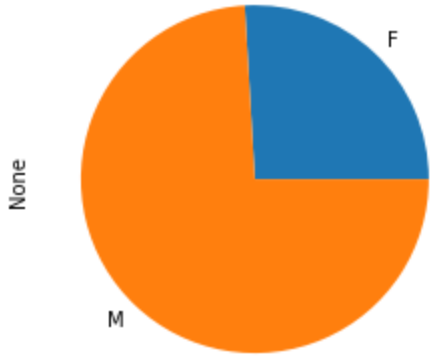


```
[ ]  #Thong ke du lieu
     dulieu = df.groupby('gender').size()
     dulieu
```

```
gender
F     7
M    20
dtype: int64
```

```
[ ]  #ve bieu do thong ke
     dulieu.plot.pie()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7c42176510>
```



```
[ ]  #Làm sạch dữ liệu
     #1.Kiểm tra dữ liệu có bị null hay không? Nếu false là không bị
     df.isna()
```

	id	name	mark	gender	pass
0	False	False	False	False	False
1	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False
10	False	False	False	False	False
11	False	False	False	False	False

```
[ ]  #đọc thông tin từng dòng
     df.iloc[3]
```

```
id        7
name      A
mark      3
gender    M
pass      0
Name: 6, dtype: object
```

```
[ ]  #Đọc thông tin từng thành phần trong dòng
     df['name'].iloc[3]
```

```
'A'
```

```
[ ]  #Lấy nhiều dòng, tương tự df[0:4]
     df.iloc[0:4]
```

	id	name	mark	gender	pass
0	1	A	5	M	1
1	2	B	6	F	1
5	6	B	4	M	0
6	7	A	3	M	0

```
[ ]  #Đọc thông tin từng dòng
     for index,row in df.iterrows():
       print(index,row)
```

```
0 id        1
name        A
mark        5
gender      M
pass        1
Name: 0, dtype: object
1 id        2
name        B
mark        6
gender      F
pass        1
Name: 1, dtype: object
5 id        6
name        B
mark        4
gender      M
pass        0
Name: 5, dtype: object
6 id        7
name        A
```

```
[ ]  #Tìm chuỗi con trong chuỗi, trả về các dòng mà trường name chứa chuỗi con abc
     df.loc[df['name'].str.contains('abc')]
```

	id	name	mark	gender

```
[ ]  #Thống kê cho từng thành phần trong nhóm
     grouped = df.groupby('gender').agg({
         "mark":'min',
         "mark":'mean',
         "mark":'max'
     })
     grouped
```

	mark
gender	
F	8
M	9