pandas 常见用法

班级：_____　　姓名：_____

(可以附在书本 P123)

Series

```
import    pandas as pd
pds1 =pd.Series([45, 30, 35, 28], index=['学生甲', '学生乙', '学生丙', '学生丁'])
data1 = {"学生 A":45,"学生 B":30,"学生 C":35,"学生 D":28}
pds2 =pd.Series(data1)
data2 = {"i1":1,"i2":2,"i3":3,"i4":4}
pds3 = pd.Series(data2,index=['i1','i2','i3'])
pds4 = pd.Series(data2,index=['i1','i2','i3','i4','i5'])
```

程序块	运行结果	程序块	运行结果
print(pds1)	学生甲　45 学生乙　30 学生丙　35 学生丁　28 dtype: int64	for i in pds1.index: 　print(i)	学生甲 学生乙 学生丙 学生丁
print(pds2)	学生 A　45 学生 B　30 学生 C　35 学生 D　28 dtype: int64	for i in pds1.values: 　print(i)	45 30 35 28
print(pds3)	i1　1 i2　2 i3　3 dtype: int64	for i in pds1: 　print(i)	45 30 35 28
print(pds4)	i1　1.0 i2　2.0 i3　3.0 i4　4.0 i5　NaN dtype: float64	print(pds1.index)	Index(['学生甲', '学生乙', '学生丙', '学生丁'], dtype='object')
print(pds1.values)	[45 30 35 28]	pds1['学生甲']=42 print(pds1['学生甲'])	42
print(pds1[['学生甲', '学生乙']])	学生甲　45 学生乙　30 dtype: int64		

DataFrame

如从 Excel 文件中读取二维数据数据使用 df=pd.readexcel(文件名)，注意文件名要包含扩展名，同时读取的文件需要和.py 文件位于同一个位置（如都在桌面上）。

```
import pandas as pd
data={'name':['学生甲','学生乙','学生丙', '学生丁'],'语文':[100,89,110,105],'数学':[110,120,125,135],'信息':[45, 30, 35, 28]}
df1=pd.DataFrame(data)
df2=pd.DataFrame(data,index=['20200817','20200818','20200819','20200819'])
df3=pd.DataFrame(data,columns=['name','信息'])
```

程序块	运行结果
print(df1)	name 语文 数学 信息 0 学生甲 100 110 45 1 学生乙 89 120 30 2 学生丙 110 125 35 3 学生丁 105 135 28
print(df2)	name 语文 数学 信息 20200817 学生甲 100 110 45 20200818 学生乙 89 120 30 20200819 学生丙 110 125 35 20200819 学生丁 105 135 28
print(df3)	name 信息 0 学生甲 45 1 学生乙 30 2 学生丙 35 3 学生丁 28
for i in df3.index: print(i)	0 1 2 3
for i in df3.columns: print(i)	name 信息
for i in df3: print(i)	name 信息
print(df3['name'])	0 学生甲 1 学生乙 2 学生丙 3 学生丁 Name: name, dtype: object
print(df3.name)	0 学生甲 1 学生乙 2 学生丙 3 学生丁 Name: name, dtype: object

print(df3[1:3])	name 信息 1 学生乙 30 2 学生丙 35
print(df3[1::2])	name 信息 1 学生乙 30 3 学生丁 28

DataFrame 常用函数		
函数	说明	备注
axis=0 代表**跨行（down)**对每一列都要操作，axis=1 代表**跨列（across)**对每一行都要操作		
大多数函数都是将原数据取出做操作后另外赋值，不会改变原数据。若加上参数 inplace=True 则说明要取代原数据，相当于直接修改原数据（不是所有的函数都有该属性）。		
df.count()	返回 df 中每一列的非空数据项的数量	某一列数据的非空可通过 df[“列名”].count()或 df.列名.count()获取
df.sum(),df.mean()	求和、求平均值，通过 axis=0/1 确定行列	同上
df.max()、df.min()	返回最大、最小值	同上
df. describe()	返回各列的基本描述统计值，包含计数、平均数、标准差、最大值、最小值及 4 分位差	同上
df.head(n) 、df.tail(n)	返回 df1 的前 n 个、后 n 个数据记录（省略 n 为 5）	

df.groupby(n,as_index=True/False)	对各列或各行中的数据按 n(列标题)进行分组，并通过 as_index 参数确定是否以分组依据为每组索引（如该组分组值为 69，则该组索引为 69）。分组后可对其中每一组数据进行不同的操作（如获取每一组的平均分等）。	（另外赋值，以 for 形式依次输出），如
		`g=df.groupby("语文")`
		`for i in g:`
		`  print(i)`

df.sort_values(n,ascending=True/False)	按 n 列进行升/降序排列，省略 ascending 则默认升序	

df.drop(n,axis=0/1)	删除索引为 n 的行或删除列标题为 n 的列	
df.append(n,ignore_index=True)	将字典 n 添加到 df 最后	ignore_index=True 表示自动重新分配索引，False 表示保留原索引
df.insert(i,j,k)	在 i 列插入列标题为 j 的列，默认值为 k	i 为列号而非列名。直接改变原数据 df
df.rename(index/columns={i:j})	将原本的索引/列标题 i 改为 j	
pd.concat([df1,df2])	合并 DataFrame 数据 df1 与 df2	
df.set_value(i,j,k)	将索引为 i，列标题为 j 的数据内容改为 k （新版已弃用，使用 df.at[i,j]=k 代替）	
df.plot()	根据 df 的数据绘制图像	df 的数据从 plot.xlsx 获取，结果可以在绘图里查看

其他不清楚待验证的内容: