Pandas

pandas是Python最常用的資料分析工具，常用於取得資料、加工資料以及各種資料處理時使用

pandas的概要

1. 何謂pandas

- pandas是以NumPy為雛型，提供序列 (Series)與資料框架(DataFrame)的資料類型
- 要使用pandas必須先執行下列的程式匯入

```
#如NumPy的語法，使用as就能呼叫pd
import pandas as pd
```

2. 何謂 Series

Series 是一維資料，要建立Series物件可以使用Series

```
ser = pd.Series([10,20,30,40])
ser
```

```
0    10
1    20
2    30
3    40
dtype: int64
```

```
#建立了4*3矩陣的DataFrame
#此DataFrame的第一行元素為整數
#此DataFrame的第二行元素為字串
#此DataFrame的第三行元素為bool
#每一行的元素都已經指派了資料類型，所以也能輕易計算每一行的資料。如一行之中，同時出現整數與字串的資料，該行的資料類型就會是物件，此時無法進行數值的運算
df = pd.DataFrame([[10,"a",True],
          [20,"b",False,],
          [30,"c",False],
          [40,"d",True]])
df
```

	0	1	2
0	10	a	True
1	20	b	False
2	30	c	False
3	40	d	True

3. DataFrame的概要

```
#第一步利用NumPy的arrange函數建立25*4矩陣的資料，建立DataFrame物件
import numpy as np
df = pd.DataFrame(np.arange(100).reshape((25,4)))

#直接呼叫df，即可輸出DataFrame的所有資訊
df
```

	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15
4	16	17	18	19
5	20	21	22	23
6	24	25	26	27
7	28	29	30	31
8	32	33	34	35
9	36	37	38	39
10	40	41	42	43
11	44	45	46	47
12	48	49	50	51
13	52	53	54	55
14	56	57	58	59
15	60	61	62	63
16	64	65	66	67
17	68	69	70	71
18	72	73	74	75
19	76	77	78	79
20	80	81	82	83
21	84	85	86	87
22	88	89	90	91
23	92	93	94	95
24	96	97	98	99

```
#使用head方法可只輸出DataFrame開頭的5列資料
df.head()
```

	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15
4	16	17	18	19

```
#使用tail方法輸出後面的5列資料
df.tail()
```

	0	1	2	3
20	80	81	82	83
21	84	85	86	87
22	88	89	90	91
23	92	93	94	95
24	96	97	98	99

```
#使用shape屬性了解DataFrame的大小
df.shape
```

```
(25, 4)
```

4. 索引名稱、欄位名稱

- DatatFrame可指定簡單易懂的索引名稱(列的名稱)與欄位名稱(行的名稱)
- 先依照前面的方法建立DataFrame物件

```
#此時的索引名稱、欄位名稱將自動指定為從0開始的數字
df = pd.DataFrame(np.arange(6).reshape((3,2)))
df
```

	0	1
0	0	1
1	2	3
2	4	5

```
#將索引名稱轉換成字串，依序從01開始命名。欄位名稱則依序指派從A開始的英文字母
#換句話，索引名稱與欄位名稱定義為有順序的數值與英文字母。索引名稱與欄位名稱可指定為任何字串與數值，不一定非得具有順序
df.index = ["01","02","03"]
df.columns = ["A","B"]
df
```

	A	B
01	0	1
02	2	3
03	4	5

- 以上都是在DataFrame建立之後，標記索引名稱與欄位名稱。若想在建立DataFrame時直接設定索引名稱與欄位名稱，可使用下面的程式

```
name_df = pd.DataFrame(np.arange(6).reshape((3,2)),columns = ["A行","B行"],
index=["第一列","第二列","第三列"])
name_df
```

	A行	B行
第一列	0	1
第二列	2	3
第三列	4	5

```
#以字典(dic)格式建立DataFrame的方法也常見。假設每個欄位都有整理好的資料，就可以使用這個方法建立
#此範例只指定欄位名稱，索引名稱則指派為從0開始的編號
pd.DataFrame({"A行": [0,2,4], "B行": [1,3,5]})
```

	A行	B行
0	0	1
1	2	3
2	4	5

5. 篩選資料

- 介紹重新製作資料與篩選資料的方法

```
import numpy as np
import pandas as pd
df = pd.DataFrame(np.arange(12).reshape((4,3)),columns=["A","B","C"],
                          index=["第一列","第二列","第三列","第四列"])
df
```

	A	B	C
第一列	0	1	2
第二列	3	4	5
第三列	6	7	8
第四列	9	10	11

```
#介紹指定欄位名稱再篩選資料
df["A"]
```

```
第一列    0
第二列    3
第三列    6
第四列    9
Name: A, dtype: int64
```

```
#取得多個欄位的資料
df[["A","B"]]
```

	A	B
第一列	0	1
第二列	3	4
第三列	6	7
第四列	9	10

- 上述的範例是以list指定欄位。以list指定欄位，就能篩選出欄位名稱相同的資料，同時輸出為DataFrame物件

```
#指定索引名稱再篩選資料
df[:2]
```

	A	B	C
第一列	0	1	2
第二列	3	4	5

```
#使用loc與iloc這兩個方法篩選資料
# : 有輸出全部資料的意思，所以輸出整個DataFrame的資料
df.loc[:,:]
```

	A	B	C
第一列	0	1	2
第二列	3	4	5
第三列	6	7	8
第四列	9	10	11

```
#利用loc方法篩選欄位A的資料，並以Series物件輸出
df.loc[:,"A"]
```

```
第一列    0
第二列    3
第三列    6
第四列    9
Name: A, dtype: int64
```

```
#利用loc方法篩選多個欄位的資料
df.loc[:,["A","B"]]
```

	A	B
第一列	0	1
第二列	3	4
第三列	6	7
第四列	9	10

```
#篩選索引方向的資料
df.loc["第一列",:]
```

```
A    0
B    1
C    2
Name: 第一列, dtype: int64
```

```
#指定多個索引名稱，輸出所有的欄位
df.loc[["第一列","第三列"], :]
```

	A	B	C
第一列	0	1	2
第三列	6	7	8

```
#同時指定索引名稱與欄位名稱
df.loc[["第一列"],["A","C"]]
```

	A	C
第一列	0	2

```
#iloc方法篩選資料時，不用指定索引名稱，而是指定索引編號與欄位編號
#編號是從0開始，依序往1、2編號的整數
df.iloc[1,1]
```

```
4
```

```
#以範圍指定索引，再以位置指定欄位
df.iloc[1:,1]
```

```
第二列     4
第三列     7
第四列    10
Name: B, dtype: int64
```

```
#以範圍指定索引與欄位，所以會傳回DataFrame物件
df.iloc[1:,:2]
```

	A	B
第二列	3	4
第三列	6	7
第四列	9	10

讀取與寫入資料

使用pandas讀取與寫入外部檔案

1. 讀取資料 : CSV檔案

```
import pandas as pd
df = pd.read_csv("maskdata.csv",encoding="utf-8")
df
```

maskdata.csv

	醫事機構代碼	醫事機構名稱	醫事機構地址	醫事機構電話	成人口罩剩餘數	兒童口罩剩餘數	來源資料時間
0	0145080011	衛生福利部花蓮醫院豐濱原住民分院	花蓮縣豐濱鄉豐濱村光豐路４１號	8358141	1100	1020	2021/09/27 14:50:37
1	0291010010	連江縣立醫院	連江縣南竿鄉復興村２１７號	623995	8000	1150	2021/09/27 14:50:37
2	2312010014	新竹市東區衛生所	新竹市東區民族路４０之２號	(03)5236158	3420	750	2021/09/27 14:50:37
3	2312041028	新竹市北區衛生所	新竹市北區國華街六十九號一樓	(03)5353969	1660	480	2021/09/27 14:50:37
4	2312050018	新竹市香山衛生所	新竹市香山區牛埔里育德街１８８號２樓	(03)5388109	220	1100	2021/09/27 14:50:37
...	...	...	...	...	...	...	...
4241	5990010533	金志忠藥局	金門縣金城鎮東門里莒光路１１號	(082)325813	6750	1260	2021/09/27 14:50:37
4242	5990010631	大森藥局	金門縣金城鎮民生路２８、３０號１、２樓	(82)325100	830	550	2021/09/27 14:50:37
4243	5990020020	大金藥局	金門縣金沙鎮汶沙里五福街１號	(082)355382	6610	530	2021/09/27 14:50:37

時序資料

處理每月、每週這類時序資料

1. 建立一個月份的資料

```
#替一個月份的日期陣列設定開始日期與結束日期
dates = pd.date_range(start="2017-04-01",end="2017-04-30")
dates
```

```
DatetimeIndex(['2017-04-01', '2017-04-02', '2017-04-03', '2017-04-04',
               '2017-04-05', '2017-04-06', '2017-04-07', '2017-04-08',
               '2017-04-09', '2017-04-10', '2017-04-11', '2017-04-12',
               '2017-04-13', '2017-04-14', '2017-04-15', '2017-04-16',
               '2017-04-17', '2017-04-18', '2017-04-19', '2017-04-20',
               '2017-04-21', '2017-04-22', '2017-04-23', '2017-04-24',
               '2017-04-25', '2017-04-26', '2017-04-27', '2017-04-28',
               '2017-04-29', '2017-04-30'],
              dtype='datetime64[ns]', freq='D')
```

```
#將一個月份的日期陣列轉換成索引，建立成DataFrame。資料本身為亂數
np.random.seed(123)
df = pd.DataFrame(np.random.randint(1,31,30),index=dates,columns=["亂數"])
df
```

	亂數
2017-04-01	14
2017-04-02	3
2017-04-03	29
2017-04-04	3
2017-04-05	7
2017-04-06	18
2017-04-07	20
2017-04-08	11
2017-04-09	28
2017-04-10	26
2017-04-11	23
2017-04-12	2
2017-04-13	1
2017-04-14	18
2017-04-15	16
2017-04-16	10
2017-04-17	1
2017-04-18	15
2017-04-19	1
2017-04-20	16
2017-04-21	26
2017-04-22	20
2017-04-23	15
2017-04-24	30
2017-04-25	5
2017-04-26	1
2017-04-27	17
2017-04-28	5
2017-04-29	18
2017-04-30	24

2. 建立一年份365天的資料

根據開始日期建立一年份365天的日期陣列

```
dates = pd.date_range(start="2017-01-01",periods=365)
dates
```

```
DatetimeIndex(['2017-01-01', '2017-01-02', '2017-01-03', '2017-01-04',
               '2017-01-05', '2017-01-06', '2017-01-07', '2017-01-08',
               '2017-01-09', '2017-01-10',
               ...
               '2017-12-22', '2017-12-23', '2017-12-24', '2017-12-25',
               '2017-12-26', '2017-12-27', '2017-12-28', '2017-12-29',
               '2017-12-30', '2017-12-31'],
              dtype='datetime64[ns]', length=365, freq='D')
```

```
np.random.seed(123)
df = pd.DataFrame(np.random.randint(1,31,365),index=dates,columns=["亂數"])
df
```

	亂數
2017-01-01	14
2017-01-02	3
2017-01-03	29
2017-01-04	3
2017-01-05	7
...	...
2017-12-27	22
2017-12-28	5
2017-12-29	22
2017-12-30	1
2017-12-31	8

365 rows × 1 columns

3. 轉換成每月平均的資料

```
#利用365天的資料計算每月平均值
df.groupby(pd.Grouper(freq='M')).mean()
```

	亂數
2017-01-31	13.774194
2017-02-28	13.428571
2017-03-31	15.612903
2017-04-30	15.533333
2017-05-31	15.322581
2017-06-30	14.300000
2017-07-31	15.258065
2017-08-31	16.129032
2017-09-30	18.433333
2017-10-31	14.580645
2017-11-30	12.633333
2017-12-31	17.483871

- 使用groupby方法加總資料
- 參數指定為freq='M'
- Grouper可進行周期型的分組，設定為freq = 'M' 代表以月分作為分組資料的單位

```
#將參數的欄位固定為亂數，再使用resample方法輸出每月的平均值
df.loc[:,"亂數"].resample('M').mean()
```

```
2017-01-31    13.774194
2017-02-28    13.428571
2017-03-31    15.612903
2017-04-30    15.533333
2017-05-31    15.322581
2017-06-30    14.300000
2017-07-31    15.258065
2017-08-31    16.129032
2017-09-30    18.433333
2017-10-31    14.580645
2017-11-30    12.633333
2017-12-31    17.483871
Freq: M, Name: 亂數, dtype: float64
```

4. 條件複製的索引

```
#先確認如何建立一年份的星期六資料
pd.date_range(start="2017-01-01",end="2017-12-31",freq="W-SAT")
```

```
DatetimeIndex(['2017-01-07', '2017-01-14', '2017-01-21', '2017-01-28',
               '2017-02-04', '2017-02-11', '2017-02-18', '2017-02-25',
               '2017-03-04', '2017-03-11', '2017-03-18', '2017-03-25',
               '2017-04-01', '2017-04-08', '2017-04-15', '2017-04-22',
               '2017-04-29', '2017-05-06', '2017-05-13', '2017-05-20',
               '2017-05-27', '2017-06-03', '2017-06-10', '2017-06-17',
               '2017-06-24', '2017-07-01', '2017-07-08', '2017-07-15',
               '2017-07-22', '2017-07-29', '2017-08-05', '2017-08-12',
               '2017-08-19', '2017-08-26', '2017-09-02', '2017-09-09',
               '2017-09-16', '2017-09-23', '2017-09-30', '2017-10-07',
               '2017-10-14', '2017-10-21', '2017-10-28', '2017-11-04',
               '2017-11-11', '2017-11-18', '2017-11-25', '2017-12-02',
               '2017-12-09', '2017-12-16', '2017-12-23', '2017-12-30'],
              dtype='datetime64[ns]', freq='W-SAT')
```

- date_range函數的參數start與end都指定為freq="W-SAT"，所以能輸出start與end之間的星期六日期

```
#以結尾為星期六的一週為單位，統整一年份的資料
df_year = pd.DataFrame(df.groupby(pd.Grouper(freq='W-SAT')).sum(),columns=['亂數'])
df_year
```

	亂數
2017-01-07	94
2017-01-14	109
2017-01-21	85
2017-01-28	93
2017-02-04	81
2017-02-11	127
2017-02-18	114
2017-02-25	82
2017-03-04	71
2017-03-11	117
2017-03-18	132
2017-03-25	103
2017-04-01	105
2017-04-08	133
2017-04-15	109
2017-04-22	111
2017-04-29	67
2017-05-06	108
2017-05-13	124
2017-05-20	103
2017-05-27	78
2017-06-03	131
2017-06-10	80

缺損值處理

- 缺損值就是會顯示NaN，也就是沒有資料的項目
- 一旦出現缺損值，就有可能得出錯誤、非預期的計算結果，所以才必須先針對缺損值進行處理。

```
#讀取CSV檔案，作為DataFrame使用
import pandas as pd
df_mask = pd.read_csv("maskdatadelete.csv",encoding="utf-8",index_col='醫事機構代碼',parse_dates=True)
df_mask
```

醫事機構代碼	醫事機構名稱	醫事機構地址	醫事機構電話	成人口罩剩餘數	兒童口罩剩餘數	來源資料時間
145080011	衛生福利部花蓮醫院豐濱原住民分院	花蓮縣豐濱鄉豐濱村光豐路４１號	8358141	NaN	NaN	2021/9/29 14:47
291010010	連江縣立醫院	連江縣南竿鄉復興村２１７號	623995	8000.0	1150.0	2021/9/29 14:47
2312010014	新竹市東區衛生所	新竹市東區民族路４０之２號	(03)5236158	3170.0	700.0	2021/9/29 14:47
2312041028	新竹市北區衛生所	新竹市北區國華街六十九號一樓	(03)5353969	NaN	NaN	2021/9/29 14:47
2312050018	新竹市香山衛生所	新竹市香山區牛埔里育德街１８８號２樓	(03)5388109	140.0	1100.0	2021/9/29 14:47
...	...	...	...	...	...	...
5990010533	金志忠藥局	金門縣金城鎮東門里莒光路１１號	(082)325813	6750.0	1260.0	2021/9/29 14:47
5990010631	大森藥局	金門縣金城鎮民生路２８、３０號１、２樓	(82)325100	810.0	550.0	2021/9/29 14:47
5990020020	大金藥局	金門縣金沙鎮汶沙里五福街１號	(082)355382	6610.0	530.0	2021/9/29 14:47
5990030044	仁愛復興藥局	金門縣金湖鎮新市里復興路４０號	(082)332368	0.0	1580.0	2021/9/29 14:47
5990030062	大山藥局	金門縣金湖鎮新市里中正路２號、２－２號	(082)333290	3710.0	1700.0	2021/9/29 14:47

4246 rows × 6 columns

```
#使用dropna方法刪除缺損值的列
df_mask_drop = df_mask.dropna()
df_mask_drop
```

醫事機構代碼	醫事機構名稱	醫事機構地址	醫事機構電話	成人口罩剩餘數	兒童口罩剩餘數	來源資料時間
291010010	連江縣立醫院	連江縣南竿鄉復興村２１７號	623995	8000.0	1150.0	2021/9/29 14:47
2312010014	新竹市東區衛生所	新竹市東區民族路４０之２號	(03)5236158	3170.0	700.0	2021/9/29 14:47
2312050018	新竹市香山衛生所	新竹市香山區牛埔里育德街１８８號２樓	(03)5388109	140.0	1100.0	2021/9/29 14:47
2322020013	嘉義市西區衛生所	嘉義市西區福全里德明路１號	(05)2337355	3900.0	2100.0	2021/9/29 14:47
2331200010	新北市坪林區衛生所	新北市坪林區坪林街１０４號	(02)26656272	230.0	1720.0	2021/9/29 14:47
...	...	...	...	...	...	...
5990010533	金志忠藥局	金門縣金城鎮東門里莒光路１１號	(082)325813	6750.0	1260.0	2021/9/29 14:47
5990010631	大森藥局	金門縣金城鎮民生路２８、３０號１、２樓	(82)325100	810.0	550.0	2021/9/29 14:47
5990020020	大金藥局	金門縣金沙鎮汶沙里五福街１號	(082)355382	6610.0	530.0	2021/9/29 14:47
5990030044	仁愛復興藥局	金門縣金湖鎮新市里復興路４０號	(082)332368	0.0	1580.0	2021/9/29 14:47
5990030062	大山藥局	金門縣金湖鎮新市里中正路２號、２－２號	(082)333290	3710.0	1700.0	2021/9/29 14:47

4244 rows × 6 columns

```
#將0指定給fillna方法，將0代入缺損值
df_mask_fillna = df_mask.fillna(0)
df_mask_fillna
```

醫事機構代碼	醫事機構名稱	醫事機構地址	醫事機構電話	成人口罩剩餘數	兒童口罩剩餘數	來源資料時間
145080011	衛生福利部花蓮醫院豐濱原住民分院	花蓮縣豐濱鄉豐濱村光豐路４１號	8358141	0.0	0.0	2021/9/29 14:47
291010010	連江縣立醫院	連江縣南竿鄉復興村２１７號	623995	8000.0	1150.0	2021/9/29 14:47
2312010014	新竹市東區衛生所	新竹市東區民族路４０之２號	(03)5236158	3170.0	700.0	2021/9/29 14:47
2312041028	新竹市北區衛生所	新竹市北區國華街六十九號一樓	(03)5353969	0.0	0.0	2021/9/29 14:47
2312050018	新竹市香山衛生所	新竹市香山區牛埔里育德街１８８號２樓	(03)5388109	140.0	1100.0	2021/9/29 14:47
...	...	...	...	...	...	...
5990010533	金志忠藥局	金門縣金城鎮東門里莒光路１１號	(082)325813	6750.0	1260.0	2021/9/29 14:47
5990010631	大森藥局	金門縣金城鎮民生路２８、３０號１、２樓	(82)325100	810.0	550.0	2021/9/29 14:47
5990020020	大金藥局	金門縣金沙鎮汶沙里五福街１號	(082)355382	6610.0	530.0	2021/9/29 14:47
5990030044	仁愛復興藥局	金門縣金湖鎮新市里復興路４０號	(082)332368	0.0	1580.0	2021/9/29 14:47
5990030062	大山藥局	金門縣金湖鎮新市里中正路２號、２－２號	(082)333290	3710.0	1700.0	2021/9/29 14:47

4246 rows × 6 columns

```
#將method='ffill'指定給fillna方法，讓缺損值沿用上一個值
df_mask_fill = df_mask.fillna(method='ffill')
df_mask_fill
```

醫事機構代碼	醫事機構名稱	醫事機構地址	醫事機構電話	成人口罩剩餘數	兒童口罩剩餘數	來源資料時間
145080011	衛生福利部花蓮醫院豐濱原住民分院	花蓮縣豐濱鄉豐濱村光豐路４１號	8358141	NaN	NaN	2021/9/29 14:47
291010010	連江縣立醫院	連江縣南竿鄉復興村２１７號	623995	8000.0	1150.0	2021/9/29 14:47
2312010014	新竹市東區衛生所	新竹市東區民族路４０之２號	(03)5236158	3170.0	700.0	2021/9/29 14:47
2312041028	新竹市北區衛生所	新竹市北區國華街六十九號一樓	(03)5353969	3170.0	700.0	2021/9/29 14:47
2312050018	新竹市香山衛生所	新竹市香山區牛埔里育德街１８８號２樓	(03)5388109	140.0	1100.0	2021/9/29 14:47
...	...	...	...	...	...	...
5990010533	金志忠藥局	金門縣金城鎮東門里莒光路１１號	(082)325813	6750.0	1260.0	2021/9/29 14:47
5990010631	大森藥局	金門縣金城鎮民生路２８、３０號１、２樓	(82)325100	810.0	550.0	2021/9/29 14:47
5990020020	大金藥局	金門縣金沙鎮汶沙里五福街１號	(082)355382	6610.0	530.0	2021/9/29 14:47
5990030044	仁愛復興藥局	金門縣金湖鎮新市里復興路４０號	(082)332368	0.0	1580.0	2021/9/29 14:47
5990030062	大山藥局	金門縣金湖鎮新市里中正路２號、２－２號	(082)333290	3710.0	1700.0	2021/9/29 14:47

4246 rows × 6 columns

```
#最後確認以平均值、中位數、眾數保存缺損值的方法
#將 df_mask.mean()指定給fillna方法，就能以其他值的平均值代替缺損值
df_mask_fillmean = df_mask.fillna(df_mask.mean())
df_mask_fillmean
```

醫事機構代碼	醫事機構名稱	醫事機構地址	醫事機構電話	成人口罩剩餘數	兒童口罩剩餘數	來源資料時間
145080011	衛生福利部花蓮醫院豐濱原住民分院	花蓮縣豐濱鄉豐濱村光豐路４１號	8358141	1994.516965	1161.340716	2021/9/29 14:47
291010010	連江縣立醫院	連江縣南竿鄉復興村２１７號	623995	8000.000000	1150.000000	2021/9/29 14:47
2312010014	新竹市東區衛生所	新竹市東區民族路４０之２號	(03)5236158	3170.000000	700.000000	2021/9/29 14:47
2312041028	新竹市北區衛生所	新竹市北區國華街六十九號一樓	(03)5353969	1994.516965	1161.340716	2021/9/29 14:47
2312050018	新竹市香山衛生所	新竹市香山區牛埔里育德街１８８號２樓	(03)5388109	140.000000	1100.000000	2021/9/29 14:47
...	...	...	...	...	...	...
5990010533	金志忠藥局	金門縣金城鎮東門里莒光路１１號	(082)325813	6750.000000	1260.000000	2021/9/29 14:47
5990010631	大森藥局	金門縣金城鎮民生路２８、３０號１、２樓	(82)325100	810.000000	550.000000	2021/9/29 14:47
5990020020	大金藥局	金門縣金沙鎮汶沙里五福街１號	(082)355382	6610.000000	530.000000	2021/9/29 14:47
5990030044	仁愛復興藥局	金門縣金湖鎮新市里復興路４０號	(082)332368	0.000000	1580.000000	2021/9/29 14:47
5990030062	大山藥局	金門縣金湖鎮新市里中正路２號、２－２號	(082)333290	3710.000000	1700.000000	2021/9/29 14:47

4246 rows × 6 columns

```
#以中位數代替缺損值
#median()
df_mask_fillmean = df_mask.fillna(df_mask.median())
df_mask_fillmean
```

醫事機構代碼	醫事機構名稱	醫事機構地址	醫事機構電話	成人口罩剩餘數	兒童口罩剩餘數	來源資料時間
145080011	衛生福利部花蓮醫院豐濱原住民分院	花蓮縣豐濱鄉豐濱村光豐路４１號	8358141	1480.0	1130.0	2021/9/29 14:47
291010010	連江縣立醫院	連江縣南竿鄉復興村２１７號	623995	8000.0	1150.0	2021/9/29 14:47
2312010014	新竹市東區衛生所	新竹市東區民族路４０之２號	(03)5236158	3170.0	700.0	2021/9/29 14:47
2312041028	新竹市北區衛生所	新竹市北區國華街六十九號一樓	(03)5353969	1480.0	1130.0	2021/9/29 14:47
2312050018	新竹市香山衛生所	新竹市香山區牛埔里育德街１８８號２樓	(03)5388109	140.0	1100.0	2021/9/29 14:47
...	...	...	...	...	...	...
5990010533	金志忠藥局	金門縣金城鎮東門里莒光路１１號	(082)325813	6750.0	1260.0	2021/9/29 14:47
5990010631	大森藥局	金門縣金城鎮民生路２８、３０號１、２樓	(82)325100	810.0	550.0	2021/9/29 14:47
5990020020	大金藥局	金門縣金沙鎮汶沙里五福街１號	(082)355382	6610.0	530.0	2021/9/29 14:47
5990030044	仁愛復興藥局	金門縣金湖鎮新市里復興路４０號	(082)332368	0.0	1580.0	2021/9/29 14:47
5990030062	大山藥局	金門縣金湖鎮新市里中正路２號、２－２號	(082)333290	3710.0	1700.0	2021/9/29 14:47

4246 rows × 6 columns

```
#以眾數代替缺損值
df_mask_fillmean = df_mask.fillna(df_mask.mode().iloc[0,:])
df_mask_fillmean
```

醫事機構代碼	醫事機構名稱	醫事機構地址	醫事機構電話	成人口罩剩餘數	兒童口罩剩餘數	來源資料時間
145080011	衛生福利部花蓮醫院豐濱原住民分院	花蓮縣豐濱鄉豐濱村光豐路４１號	8358141	0.0	1200.0	2021/9/29 14:47
291010010	連江縣立醫院	連江縣南竿鄉復興村２１７號	623995	8000.0	1150.0	2021/9/29 14:47
2312010014	新竹市東區衛生所	新竹市東區民族路４０之２號	(03)5236158	3170.0	700.0	2021/9/29 14:47
2312041028	新竹市北區衛生所	新竹市北區國華街六十九號一樓	(03)5353969	0.0	1200.0	2021/9/29 14:47
2312050018	新竹市香山衛生所	新竹市香山區牛埔里育德街１８８號２樓	(03)5388109	140.0	1100.0	2021/9/29 14:47
...	...	...	...	...	...	...
5990010533	金志忠藥局	金門縣金城鎮東門里莒光路１１號	(082)325813	6750.0	1260.0	2021/9/29 14:47
5990010631	大森藥局	金門縣金城鎮民生路２８、３０號１、２樓	(82)325100	810.0	550.0	2021/9/29 14:47
5990020020	大金藥局	金門縣金沙鎮汶沙里五福街１號	(082)355382	6610.0	530.0	2021/9/29 14:47
5990030044	仁愛復興藥局	金門縣金湖鎮新市里復興路４０號	(082)332368	0.0	1580.0	2021/9/29 14:47
5990030062	大山藥局	金門縣金湖鎮新市里中正路２號、２－２號	(082)333290	3710.0	1700.0	2021/9/29 14:47

4246 rows × 6 columns

資料合併

重新呼叫資料，讓DataFrame彼此合併

1. 讀取先前儲存的資料

```
#讀取儲存的資料
df = pd.read_csv("SleepStudyData.csv")
df
```

	Enough	Hours	PhoneReach	PhoneTime	Tired	Breakfast
0	Yes	8.0	Yes	Yes	3	Yes
1	No	6.0	Yes	Yes	3	No
2	Yes	6.0	Yes	Yes	2	Yes
3	No	7.0	Yes	Yes	4	No
4	No	7.0	Yes	Yes	2	Yes
...	...	...	...	...	...	...
99	No	7.0	Yes	Yes	2	Yes
100	No	7.0	No	Yes	3	Yes
101	Yes	8.0	Yes	Yes	3	Yes
102	Yes	7.0	Yes	Yes	2	Yes
103	Yes	6.0	Yes	Yes	3	Yes

104 rows × 6 columns

```
#讀取另一個DataFrame，也顯示其中的內容
df_moved = pd.read_csv("StudentsPerformance.csv")
df_moved
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

1000 rows × 8 columns

2. 行方向的資料合併

```
#讓兩個DataFrame朝行方向(欄方向)合併
#使用concat函數將兩個DataFrame轉換成list再傳遞給參數。
#將axis = 1 指定給參數，就能沿著行方向合併
df_merged = pd.concat([df, df_moved],axis=1)
df_merged
```

	Enough	Hours	PhoneReach	PhoneTime	Tired	Breakfast	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	Yes	8.0	Yes	Yes	3.0	Yes	female	group B	bachelor's degree	standard	none	72	72	74
1	No	6.0	Yes	Yes	3.0	No	female	group C	some college	standard	completed	69	90	88
2	Yes	6.0	Yes	Yes	2.0	Yes	female	group B	master's degree	standard	none	90	95	93
3	No	7.0	Yes	Yes	4.0	No	male	group A	associate's degree	free/reduced	none	47	57	44
4	No	7.0	Yes	Yes	2.0	Yes	male	group C	some college	standard	none	76	78	75
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	NaN	NaN	NaN	NaN	NaN	NaN	female	group E	master's degree	standard	completed	88	99	95
996	NaN	NaN	NaN	NaN	NaN	NaN	male	group C	high school	free/reduced	none	62	55	55
997	NaN	NaN	NaN	NaN	NaN	NaN	female	group C	high school	free/reduced	completed	59	71	65
998	NaN	NaN	NaN	NaN	NaN	NaN	female	group D	some college	standard	completed	68	78	77
999	NaN	NaN	NaN	NaN	NaN	NaN	female	group D	some college	free/reduced	none	77	86	86

1000 rows × 14 columns

3. 列方向的資料合併

```
#讓兩個DataFrame沿著列方向(索引方向)合併
#使用concat函數將兩個DataFrame轉換成list再傳遞給參數。
#將axis = 0 指定給參數，就能沿著列方向合併
df_merged_01 = pd.concat([df_merged,df],axis=0,sort=True)
df_merged_01
```

	Breakfast	Enough	Hours	PhoneReach	PhoneTime	Tired	gender	lunch	math score	parental level of education	race/ethnicity	reading score	test preparation course	writing score
0	Yes	Yes	8.0	Yes	Yes	3.0	female	standard	72.0	bachelor's degree	group B	72.0	none	74.0
1	No	No	6.0	Yes	Yes	3.0	female	standard	69.0	some college	group C	90.0	completed	88.0
2	Yes	Yes	6.0	Yes	Yes	2.0	female	standard	90.0	master's degree	group B	95.0	none	93.0
3	No	No	7.0	Yes	Yes	4.0	male	free/reduced	47.0	associate's degree	group A	57.0	none	44.0
4	Yes	No	7.0	Yes	Yes	2.0	male	standard	76.0	some college	group C	78.0	none	75.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
99	Yes	No	7.0	Yes	Yes	2.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
100	Yes	No	7.0	No	Yes	3.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
101	Yes	Yes	8.0	Yes	Yes	3.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
102	Yes	Yes	7.0	Yes	Yes	2.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
103	Yes	Yes	6.0	Yes	Yes	3.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1104 rows × 14 columns

操作統計資料

```
import pandas as pd
df = pd.read_csv("SleepStudyData.csv")
df.head()
```

	Enough	Hours	PhoneReach	PhoneTime	Tired	Breakfast
0	Yes	8.0	Yes	Yes	3	Yes
1	No	6.0	Yes	Yes	3	No
2	Yes	6.0	Yes	Yes	2	Yes
3	No	7.0	Yes	Yes	4	No
4	No	7.0	Yes	Yes	2	Yes

1. 基本統計量

輸出各種基本統計量

```
#利用max方法確認最大值
df.loc[:,"Tired"].max()
```

```
5
```

```
#以min方法確認最小值
df.loc[:,"Hours"].min()
```

```
2.0
```

```
#使用mode方法確認眾數
df.loc[:,"Tired"].mode()
```

```
0    3
dtype: int64
```

```
#使用mean方法確認算術平均(平均值)
df.loc[:,"Tired"].mean()
```

```
3.076923076923077
```

```
#使用median確認中位數
df.loc[:,"Tired"].median()
```

```
3.0
```

```
#使用std方法確認標準差。
#輸出是樣本差
df.loc[:,"Tired"].std()
```

```
1.0115095677628945
```

```
#使用count方法確認資料筆數
#輸入的是Tired的資料筆數
df[df.loc[:, "Tired"]==3].count()
```

```
Enough       40
Hours        39
PhoneReach   40
PhoneTime    40
Tired        40
Breakfast    40
dtype: int64
```