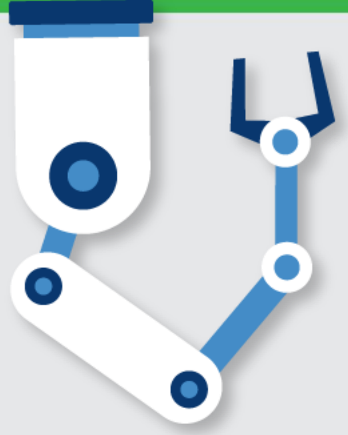
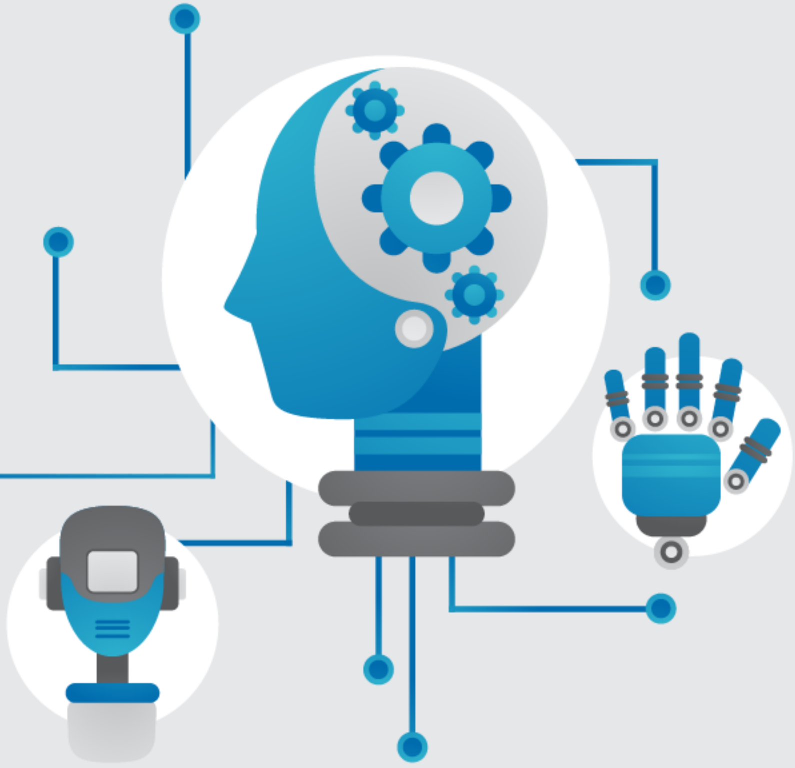


資料處理工具(II)
Pandas



Pandas套件

- Pandas的名稱源自於“Panel data”字首的縮寫。

- 是一套Python套件，即為資料處理和分析的工具，完整包含NumPy、Scipy和Matplotlab等套件功能。

- 可視為是一套Python程式版的Excel試算表工具。透過簡單的Python程式碼，就可針對表格資料執行Excel試算表的功能。

Pandas安裝與使用

- Python安裝套件

```
C:\> pip install pandas
```

- Python 程式匯入套件

```
import pandas as pd
```

Pandas資料結構

› Pandas套件兩種常用資料結構

- **Series物件：是一個類似陣列的物件**

  跟numpy的陣列不同的是，可以定義自己的index（任何資料型態），也可想像成是特殊化的Dictionary。

- **DataFrame物件：類似試算表的表格資料**

  DataFrame跟Series一樣，可以指定index，
  但這邊可以想像成DataFrame是多個Series組成。

建立DataFrame

指令

- pd.read_csv([filename])
- pd.read_json([filename])
- pd.read_html(filename])
- pd.read_excel([filename])

參數

- filename：檔案位置（string）

回傳　DataFrame

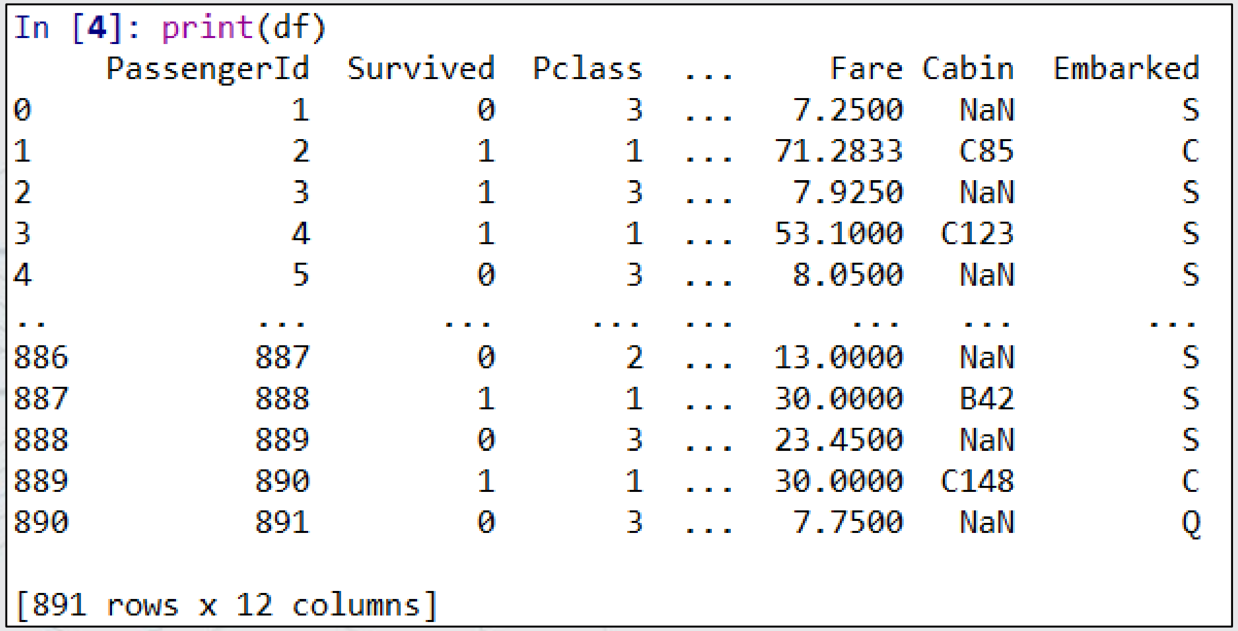
DataFrame範例

- 範例程式

```
import pandas as pd

df=pd.read_csv('train.csv‘)

print(df)
```



```
In [4]: print(df)
     PassengerId  Survived  Pclass  ...     Fare Cabin  Embarked
0              1         0       3  ...   7.2500   NaN         S
1              2         1       1  ...  71.2833   C85         C
2              3         1       3  ...   7.9250   NaN         S
3              4         1       1  ...  53.1000  C123         S
4              5         0       3  ...   8.0500   NaN         S
..           ...       ...     ...  ...      ...   ...       ...
886          887         0       2  ...  13.0000   NaN         S
887          888         1       1  ...  30.0000   B42         S
888          889         0       3  ...  23.4500   NaN         S
889          890         1       1  ...  30.0000  C148         C
890          891         0       3  ...   7.7500   NaN         Q

[891 rows x 12 columns]
```

觀察資料

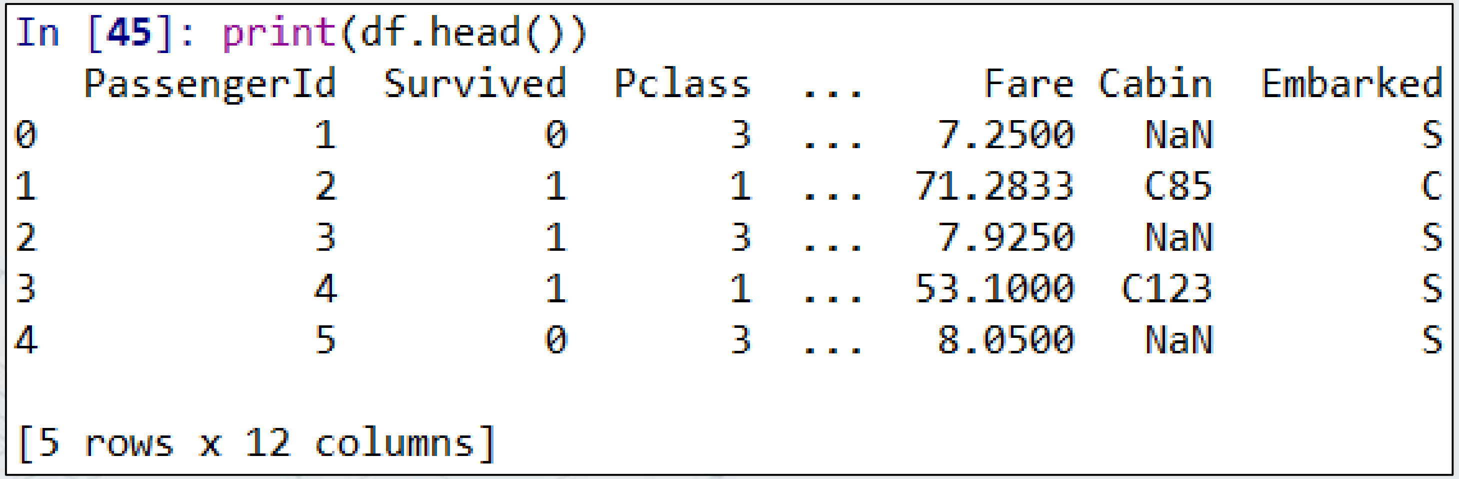
› 指令

- DataFrame.head()，返回前5筆資料
- DataFrame.info()，返回DataFrame相關資訊
- Series.describe()，返回非Nan的統計資料

觀察資料範例 - head

- 範例程式

```
print(df.head())
```



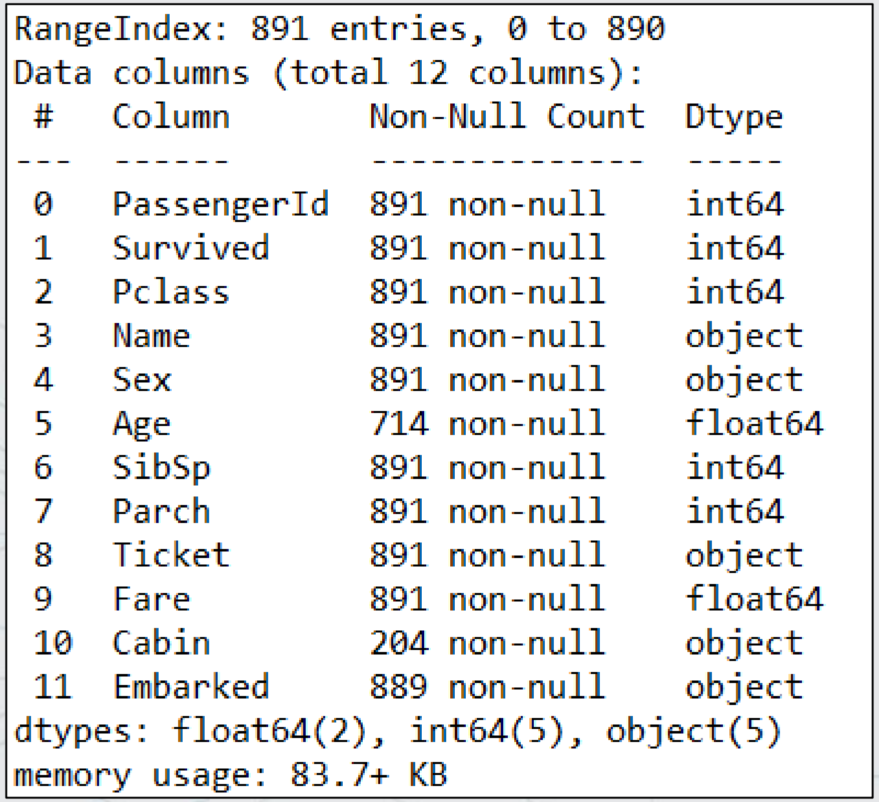
```
In [45]: print(df.head())
   PassengerId  Survived  Pclass  ...     Fare Cabin  Embarked
0            1         0       3  ...   7.2500   NaN         S
1            2         1       1  ...  71.2833   C85         C
2            3         1       3  ...   7.9250   NaN         S
3            4         1       1  ...  53.1000  C123         S
4            5         0       3  ...   8.0500   NaN         S

[5 rows x 12 columns]
```

觀察資料範例 - info

- 範例程式

```
print(df.info())
```



```
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

通常觀察：
有哪些**欄位**

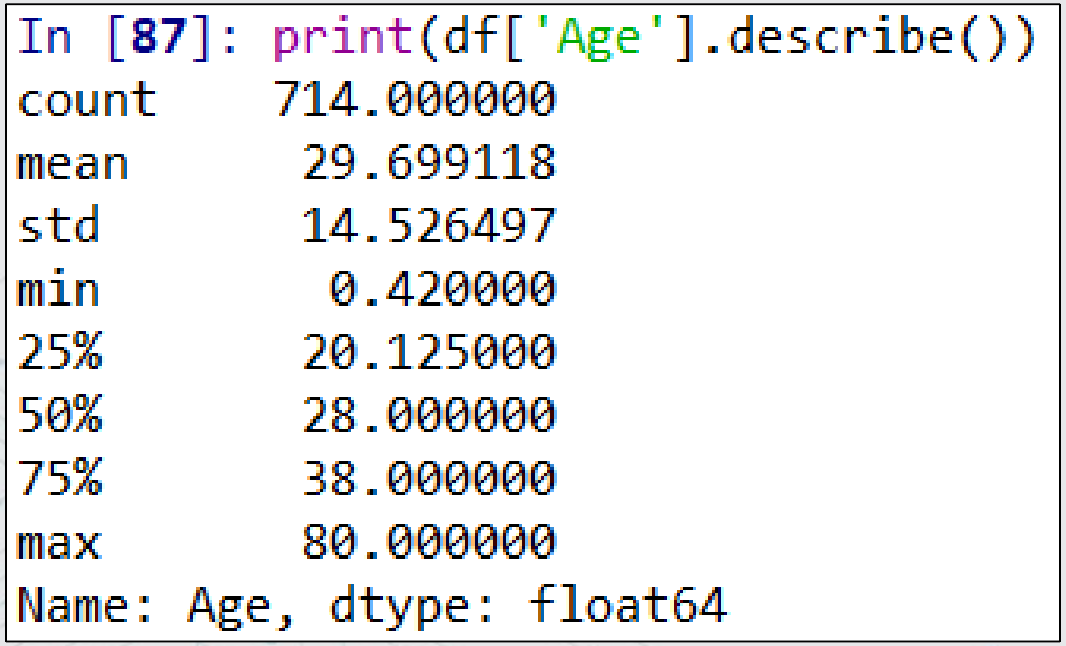
Non-Null Count
不足891代表有Nan

Dtype
為Object代表非數值

觀察資料範例 - describe

- 範例程式

```
print(df['Age'].describe())
```



```
In [87]: print(df['Age'].describe())
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

擷取資料

指令

- DataFrame[[col]]
- DataFrame[slice]
- DataFrame.loc[[index]]
- DataFrame[[boolean mask]]
- DataFrame.pop([col])

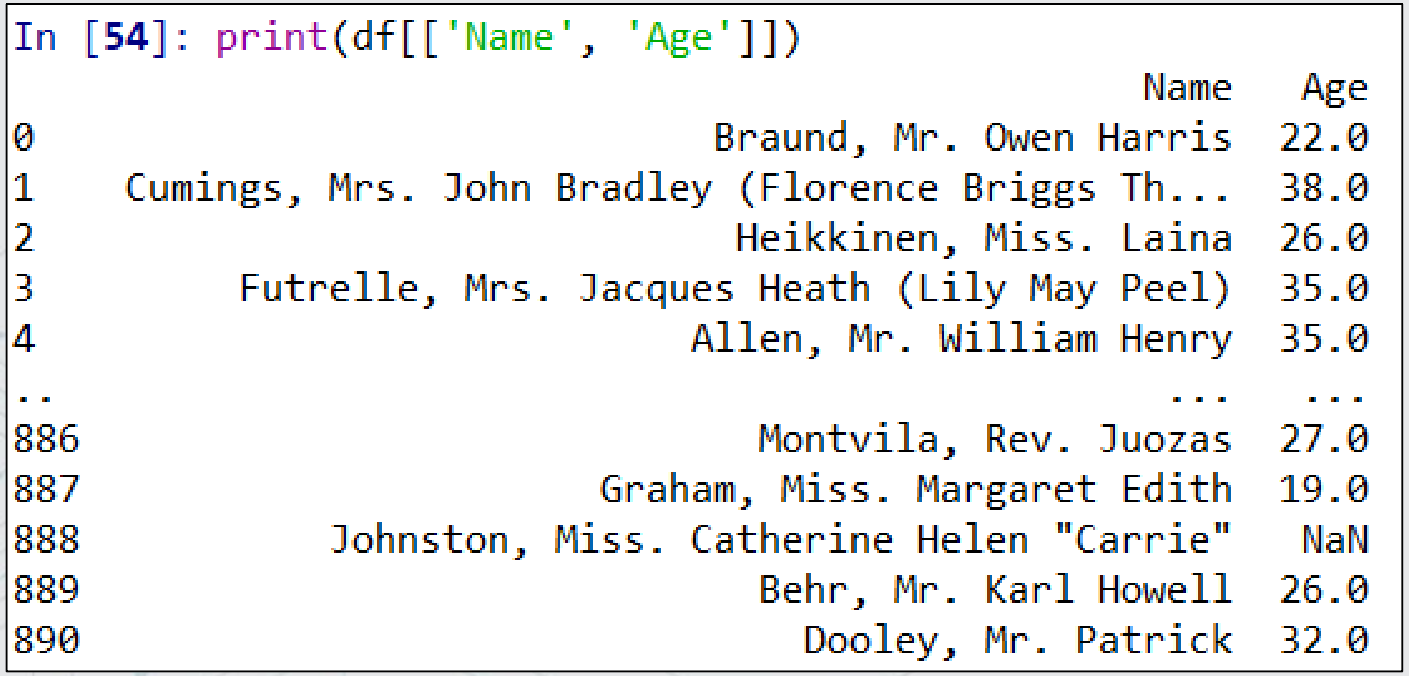
參數

- col：欄位名稱(string)
- slice：同numpy使用slice，僅一維索引，row的index
- index：索引(int or string)，可二維索引
- boolean mask：跟numpy使用boolean mask一樣

擷取資料範例

- 範例程式

```
print(df[['Name', 'Age']])
```

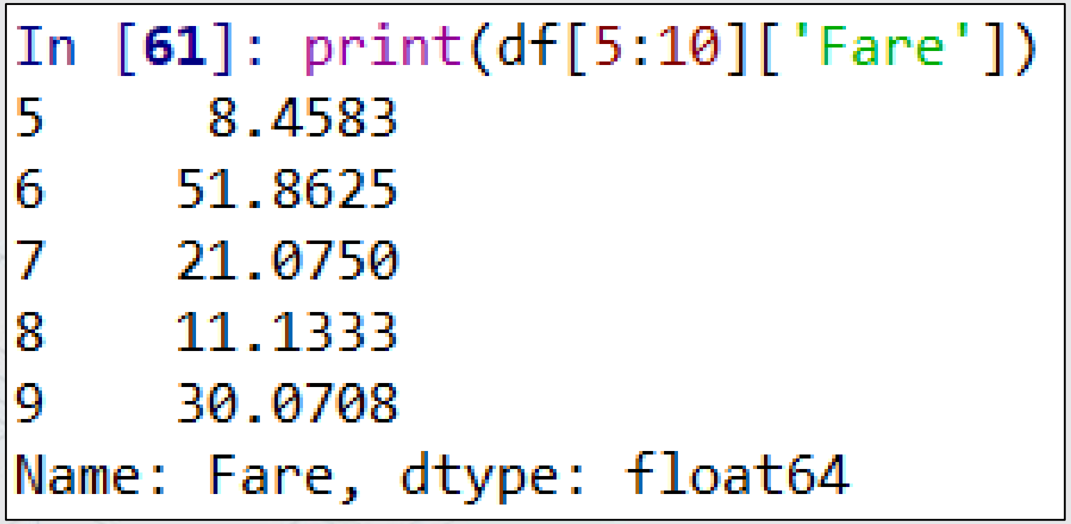


```
In [54]: print(df[['Name', 'Age']])
                                                  Name   Age
0                               Braund, Mr. Owen Harris  22.0
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  38.0
2                                Heikkinen, Miss. Laina  26.0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  35.0
4                              Allen, Mr. William Henry  35.0
..                                                 ...   ...
886                               Montvila, Rev. Juozas  27.0
887                        Graham, Miss. Margaret Edith  19.0
888            Johnston, Miss. Catherine Helen "Carrie"   NaN
889                               Behr, Mr. Karl Howell  26.0
890                                 Dooley, Mr. Patrick  32.0
```

擷取資料範例

- 範例程式

```
print(df[5:10]['Fare'])
```

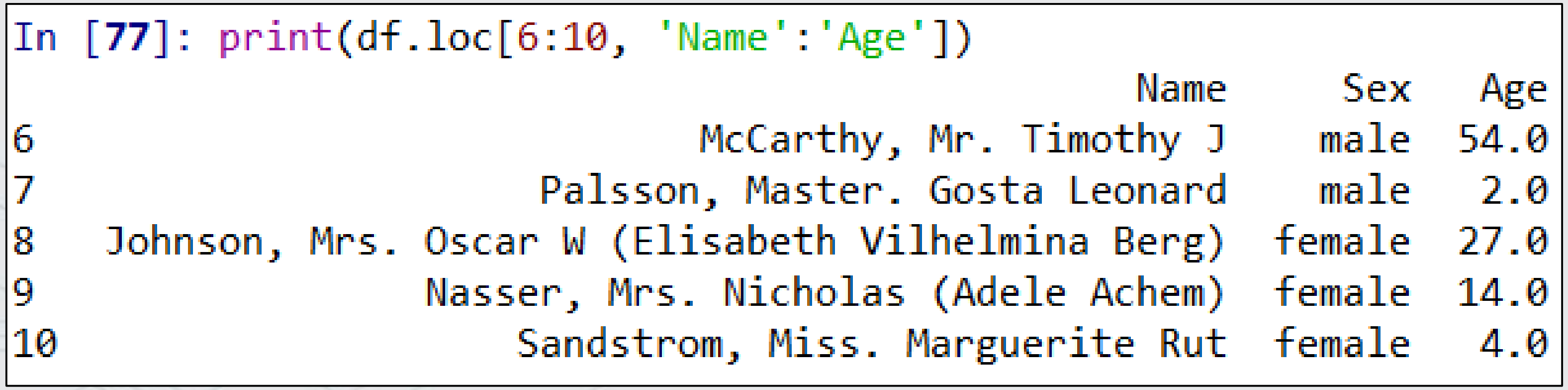


```
In [61]: print(df[5:10]['Fare'])
5     8.4583
6    51.8625
7    21.0750
8    11.1333
9    30.0708
Name: Fare, dtype: float64
```

擷取資料範例

- 範例程式

```
print(df.loc[6:10, 'Name':'Age'])
```

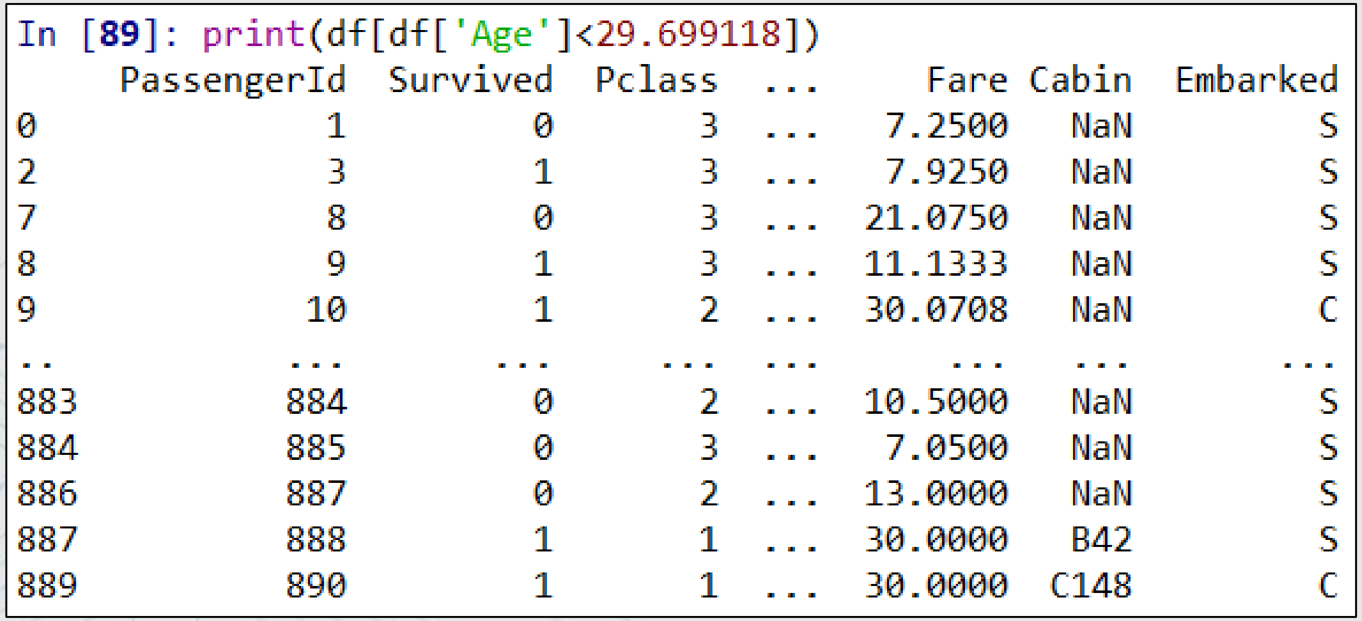


```
In [77]: print(df.loc[6:10, 'Name':'Age'])
                                                  Name     Sex   Age
6                              McCarthy, Mr. Timothy J    male  54.0
7                       Palsson, Master. Gosta Leonard    male   2.0
8    Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)  female  27.0
9                  Nasser, Mrs. Nicholas (Adele Achem)  female  14.0
10                      Sandstrom, Miss. Marguerite Rut  female   4.0
```

擷取資料範例

- 範例程式

print(df[df['Age']<29.699118])

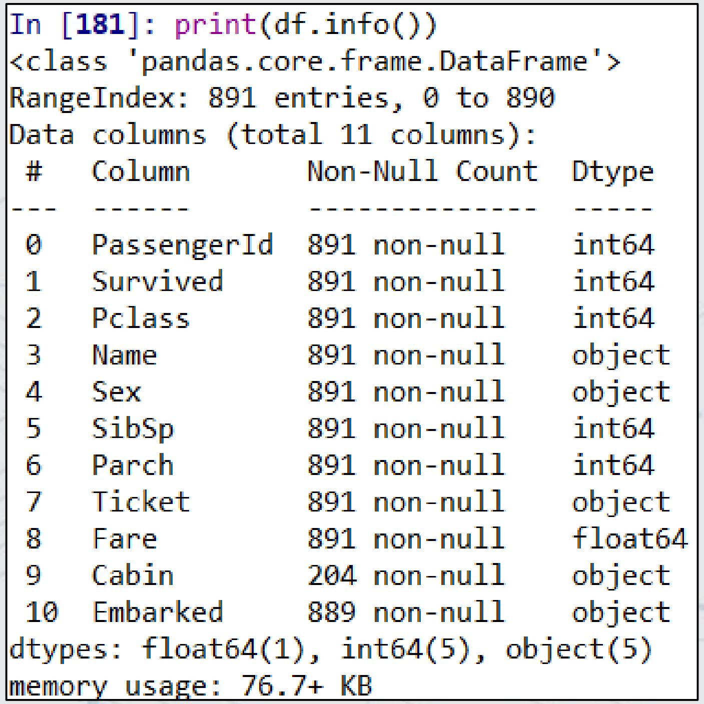


```
In [89]: print(df[df['Age']<29.699118])
     PassengerId  Survived  Pclass  ...     Fare Cabin  Embarked
0              1         0       3  ...   7.2500   NaN         S
2              3         1       3  ...   7.9250   NaN         S
7              8         0       3  ...  21.0750   NaN         S
8              9         1       3  ...  11.1333   NaN         S
9             10         1       2  ...  30.0708   NaN         C
..           ...       ...     ...  ...      ...   ...       ...
883          884         0       2  ...  10.5000   NaN         S
884          885         0       3  ...   7.0500   NaN         S
886          887         0       2  ...  13.0000   NaN         S
887          888         1       1  ...  30.0000   B42         S
889          890         1       1  ...  30.0000  C148         C
```

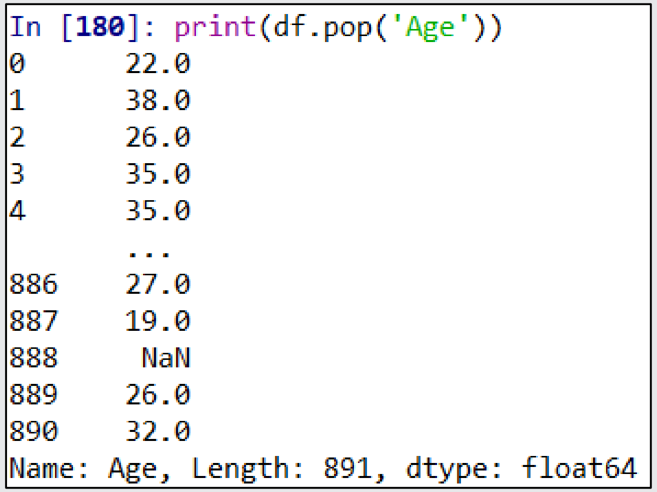
擷取資料範例

- 範例程式

```
print(df.pop('Age'))
print(df.info())
```



```
In [181]: print(df.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   SibSp        891 non-null    int64
 6   Parch        891 non-null    int64
 7   Ticket       891 non-null    object
 8   Fare         891 non-null    float64
 9   Cabin        204 non-null    object
 10  Embarked     889 non-null    object
dtypes: float64(1), int64(5), object(5)
memory usage: 76.7+ KB
```



```
In [180]: print(df.pop('Age'))
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     NaN
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

Age欄位已移除

常用數據處理方法

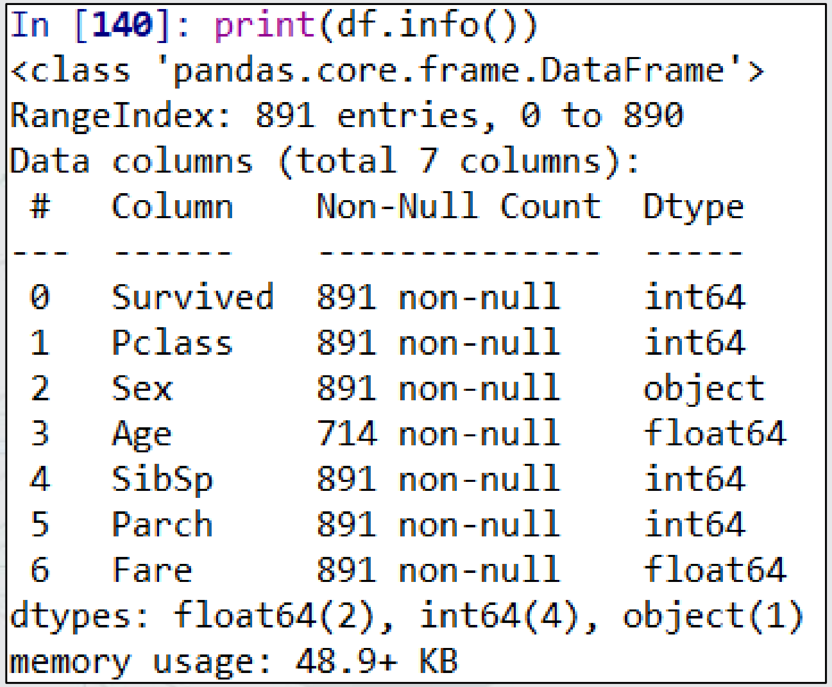
- 指令
  - DataFrame.drop
  - Series.mean、min、max
  - Series.fillna
  - series.map
  - pandas.get_dummies
  - DataFrame.values

常用數據處理方法範例 - drop

- 範例程式

```
df=df.drop(['PassengerId','Name', 'Ticket',
'Cabin', 'Embarked'], axis=1)

print(df.info())
```

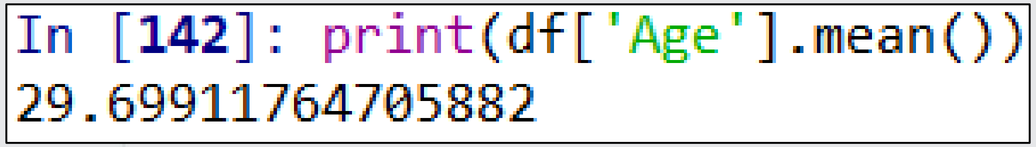


```
In [140]: print(df.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Pclass    891 non-null    int64
 2   Sex       891 non-null    object
 3   Age       714 non-null    float64
 4   SibSp     891 non-null    int64
 5   Parch     891 non-null    int64
 6   Fare      891 non-null    float64
dtypes: float64(2), int64(4), object(1)
memory usage: 48.9+ KB
```

常用數據處理方法範例 - mean

- 範例程式

```
print(df['Age'].mean())
```



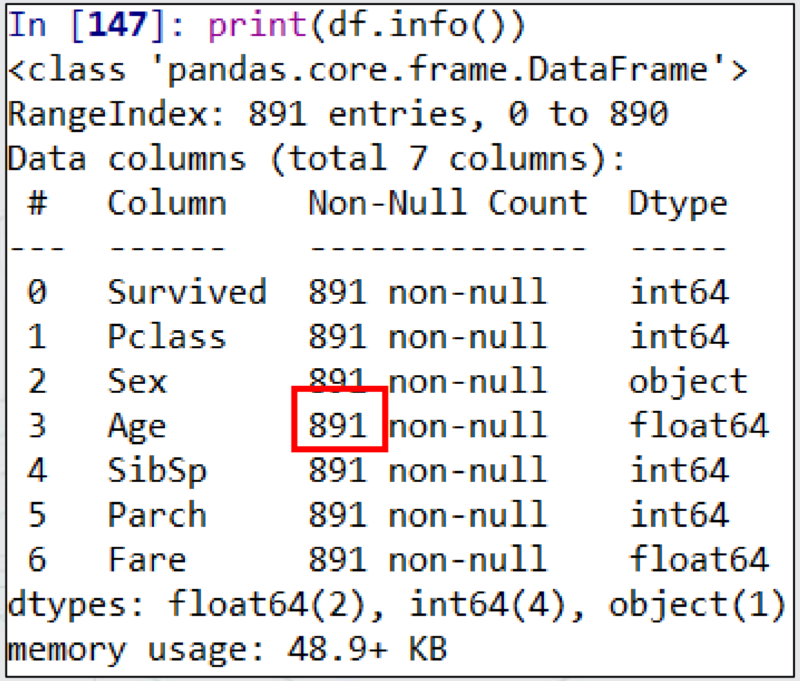
```
In [142]: print(df['Age'].mean())
29.69911764705882
```

常用數據處理理方法範例 - fillna

› 範例程式

df['Age']=df['Age'].fillna(df['Age'].mean())

print(df.info())

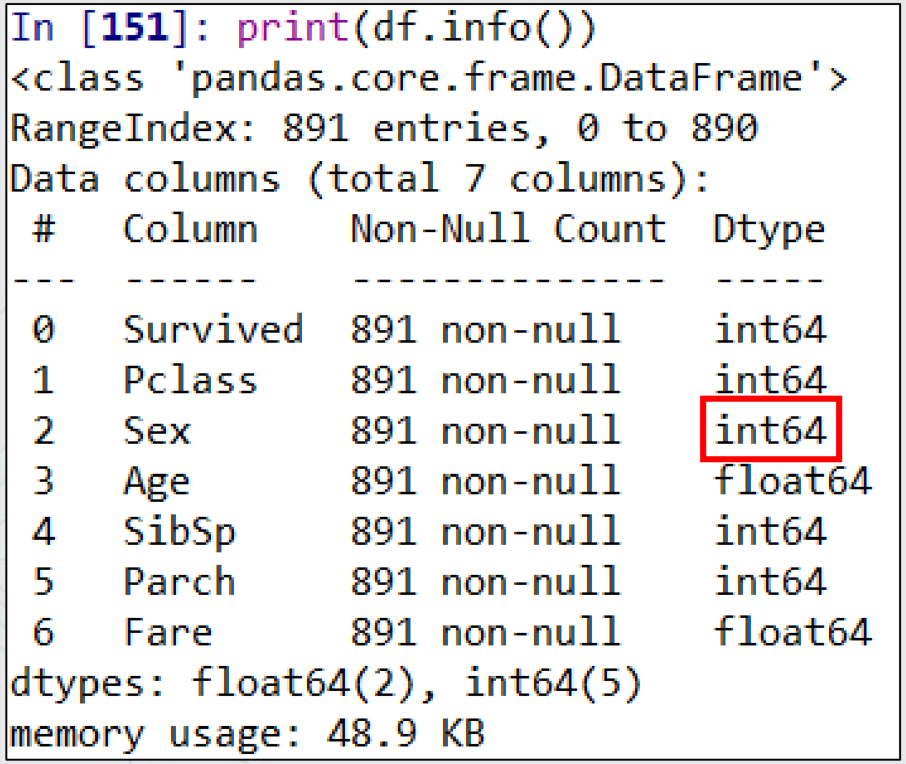


```
In [147]: print(df.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Survived  891 non-null    int64  
 1   Pclass    891 non-null    int64  
 2   Sex       891 non-null    object 
 3   Age       891 non-null    float64
 4   SibSp     891 non-null    int64  
 5   Parch     891 non-null    int64  
 6   Fare      891 non-null    float64
dtypes: float64(2), int64(4), object(1)
memory usage: 48.9+ KB
```

常用數據處理方法範例 - map

› 範例程式

```
df['Sex']=df['Sex'].map({'male':1, 'female':0})
print(df.info())
```

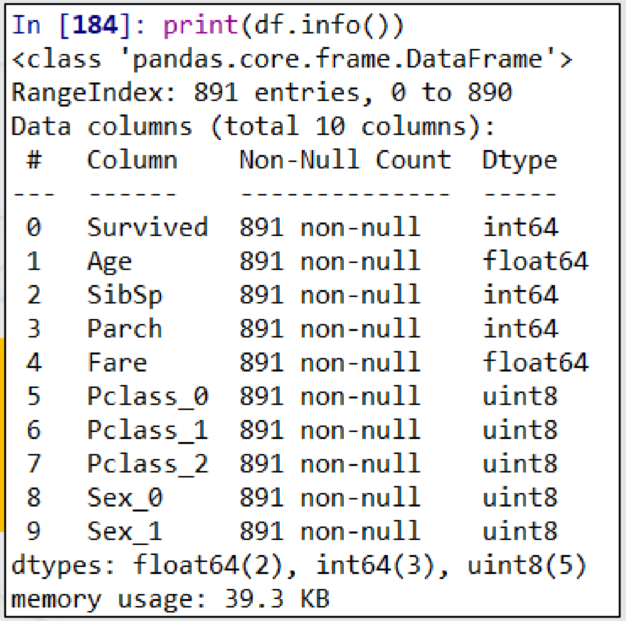


```
In [151]: print(df.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Survived  891 non-null    int64  
 1   Pclass    891 non-null    int64  
 2   Sex       891 non-null    int64  
 3   Age       891 non-null    float64
 4   SibSp     891 non-null    int64  
 5   Parch     891 non-null    int64  
 6   Fare      891 non-null    float64
dtypes: float64(2), int64(5)
memory usage: 48.9 KB
```

常用數據處理方法範例 - get_dummies

› 範例程式

```
classfication_data=['Pclass', 'Sex']
for col in classfication_data:
    pick=df.pop(col)
    df[[col+'_'+str(i) for i in range(len(pick.unique()))]]=pd.get_dummies(pick)
print(df.info())
```



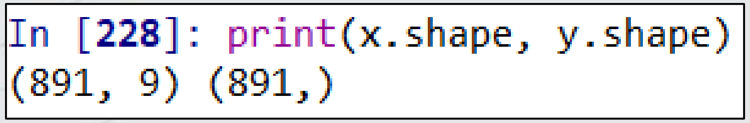
```
In [184]: print(df.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Age       891 non-null    float64
 2   SibSp     891 non-null    int64
 3   Parch     891 non-null    int64
 4   Fare      891 non-null    float64
 5   Pclass_0  891 non-null    uint8
 6   Pclass_1  891 non-null    uint8
 7   Pclass_2  891 non-null    uint8
 8   Sex_0     891 non-null    uint8
 9   Sex_1     891 non-null    uint8
dtypes: float64(2), int64(3), uint8(5)
memory usage: 39.3 KB
```

通常要執行模型訓練
類別屬性的資料
都會轉成one hot encoding

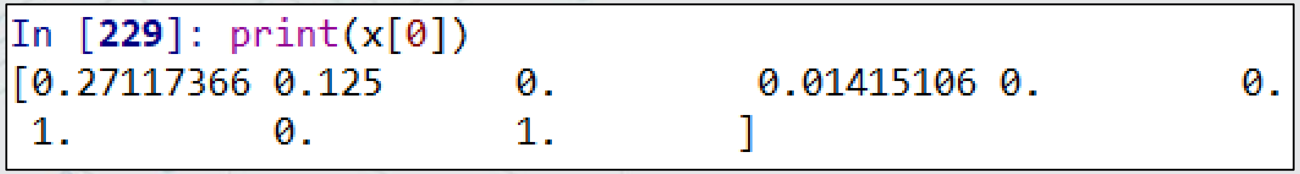
常用數據處理方法範例 - values

› 範例程式

```
# 轉成numpy.array
y=df.pop('Survived').values.astype('float32')
num_data=['Age', 'SibSp', 'Parch', 'Fare']
for col in num_data:
# min max normalization
df[col]=(df[col]-df[col].min())/(df[col].max()-df[col].min())
x=df.values.astype('float32')
print(x.shape, y.shape)
print(x[0])
```



```
In [228]: print(x.shape, y.shape)
(891, 9) (891,)
```



```
In [229]: print(x[0])
[0.27117366 0.125      0.         0.01415106 0.         0.
 1.         0.         1.        ]
```