Series

🧪 Pandas Series 建立與基本操作教學

```
# 載入 pandas 套件
import pandas as pd

# 建立 Series 並指定索引與名稱
s = pd.Series([10, 2, 3, 4, 5], index=['A', 'B', 'C', 'D', 'E'], name='月份')
print(s)
```

```
# 使用字典建立 Series
s = pd.Series({"a": 1, "b": 2, "c": 3, "d": 4, "e": 5})

# 從現有 Series 擷取部分資料
s2 = pd.Series([10, 2, 3, 4, 5], index=['A', 'B', 'C', 'D', 'E'], name='月份')
s1 = pd.Series(s2, index=["A", "C"])
print(s1)
```

Series 的屬性與存取方法

```
print(s.loc['a'])  # 顯式索引 (label-based)
print(s.iloc[0])  # 隱式索引 (position-based)
print(s.at['a'])  # 使用標籤快速訪問單一元素
print(s.iat[0])  # 使用位置快速訪問單一元素
```

訪問與篩選資料

```
s['f'] = 6
print(s.head(2))  # 取前兩個元素
print(s.tail(1))  # 取最後一個元素
```

常見方法：缺失值、統計分析

```
import numpy as np

s = pd.Series([10, 2, np.nan, None, 3, 4, 5], index=['A', 'B', 'C', 'D', 'E', 'F', 'G'],
name='data')
```

```
s.head(3)       # 前3 row
s.tail(2)       # 後2 row
s.describe()    # 敘述統計摘要
s.count()       # 非 NaN 值數量
print(s.keys())   # 返回索引物件（等同 s.index）
print(s.index)    # 索引屬性
print(s.isna())   # 是否為 NaN
s.isin([4, 5, 6])  # 是否為指定值之一
```

基本統計指標

```
print(s.mean())   # 平均值
print(s.sum())    # 總和
print(s.std())    # 標準差
print(s.var())    # 變異數
print(s.min())    # 最小值
print(s.max())    # 最大值
print(s.median())  # 中位數
```

分位數、眾數、出現次數與去重

```
print(s.quantile(0.8))     # 取得 80% 分位數的值
s['H'] = 4                 # 新增一個索引為 'H' 的元素，其值為 4
print(s.mode())            # 取得出現次數最多的值（眾數），可能回傳多個值
print(s.value_counts())    # 回傳每個值的出現次數（由大到小排序）
s.drop_duplicates()        # 回傳去除重複值後的新 Series（不會修改原 Series）
```

```
s.unique()          # 回傳所有唯一值組成的 ndarray
print(s.nunique())      # 回傳唯一值的數量（即去重後的元素個數）
```

```
6.000000000000001
0  4
Name: 月份, dtype: int64
月份
4 2
10 1
2 1
3 1
5 1
Name: count, dtype: int64
5
```

排序操作

```
s.sort_index()    # 按索引排序
s.sort_values()   # 按值排序
```

實例操作

☀ 溫度資料統計分析

```
# 問題：如何計算一周中溫度超過30度的天數？
temperatures = pd.Series([28, 31, 29, 32, 30, 27, 33],
                         index=['週一', '週二', '週三', '週四', '週五', '週六', '週日'])
print('超過30度的天數：', temperatures[temperatures > 30].count())

# 問題：如何計算一周的平均溫度？
print('平均溫度：', temperatures.mean())

# 問題：如何將一周的溫度從高到低排序？
print('從高到低排序：', temperatures.sort_values(ascending=False))
```

```
# 問題：如何找出溫度變化最大的兩天？
print('溫度變化最大的兩天', *
(temperatures.diff().abs().sort_values(ascending=False).keys()[:2]))
```

```
'''
超過30度的天數： 3
平均溫度： 30.0
從高到低排序：
週日 33
週四 32
週二 31
週五 30
週三 29
週一 28
週六 27
dtype: int64
溫度變化最大的兩天 週日 週二
'''
```

📊 股票價格分析

```
# 問題：如何計算每日股價的日收益率？
prices = pd.Series([102.3, 103.5, 105.1, 104.8, 106.2, 107.0, 106.5, 108.1, 109.3, 110.2],
                   index=pd.date_range('2023-01-01', periods=10))
returns = prices.pct_change()

# 問題：哪一天的收益率最高？
print('收益率最高日期:', returns.idxmax())

# 問題：哪一天的收益率最低？
print('收益率最低日期:', returns.idxmin())

# 問題：如何計算收益率的波動率（標準差）？
print('波動率:', returns.std())
```

```
'''
收益率最高日期: 2023-01-03 00:00:00
收益率最低日期: 2023-01-07 00:00:00
波動率: 0.007373623845361105
'''
```

🛒 銷售資料分析

```
# 問題：如何計算每季度的平均銷售額？
sales = pd.Series([120, 135, 145, 160, 155, 170, 180, 175, 190, 200, 210, 220],
                  index=pd.date_range('2022-01-01', periods=12, freq='MS'))
print(sales.resample('QS').mean()) # 每季度平均

# 問題：如何找出銷量最高的月份？
print('銷量最高的月份', sales.idxmax())

# 問題：如何計算月環比的增長率？
print('月環比的增長率')
sales.pct_change()

# 問題：如何找出連續三個月以上銷售額持續增長的月份？
mask = sales.pct_change() > 0
print(mask[mask.rolling(3).sum() == 3].keys().tolist())
```

```
'''
2022-01-01 133.333333
2022-04-01 161.666667
2022-07-01 181.666667
2022-10-01 210.000000
Freq: QS-JAN, dtype: float64
銷量最高的月份 2022-12-01 00:00:00
月環比的增長率
[Timestamp('2022-04-01 00:00:00'), Timestamp('2022-11-01 00:00:00'),
Timestamp('2022-12-01 00:00:00')]
'''
```

⏱️ 每小時銷售資料分析

```
# 問題：如何計算營業時間與非營業時間的銷售比例？
np.random.seed(42)
h = pd.Series(np.random.randint(0, 100, 24),
              index=pd.date_range('2025-01-01', periods=24, freq='h'))

# 按天統計銷售總額
day_sales = h.resample('D').sum()

# 營業時間為08:00到22:00
work_hour_mask = (h.index.hour >= 8) & (h.index.hour <= 22)
work_sales = h[work_hour_mask]
off_sales = h[~work_hour_mask]

# 計算營業時間銷售總額與非營業時間銷售總額的比例
print('營業/非營業銷售比例:', work_sales.sum() / off_sales.sum())

# 找出銷售額最高的前三個小時
print('銷售額最高的3個小時:', h.nlargest(3).keys())
```

```
'''
營業/非營業銷售比例: 1.4294354838709677
銷售額最高的3個小時: DatetimeIndex(['2025-01-01 11:00:00', '2025-01-01 01:00:00',
'2025-01-01 10:00:00'], dtype='datetime64[ns]', freq=None)
'''
```