1. Fundamentals of Data Analysis

Outline

- Pandas Basics
  - Input and Output
  - Examining DataFrame
  - Indexing and Slicing
  - Conditional
  - Changing Data Type
  - Sort
  - Apply and Applymap
  - Plot
  - Exercises and Solutions

Pandas Basics

這一節會教大家如何使用 Pandas 做基本的資料處理，包括存取檔案、檢查數據表格、選取特定資料、資料排序、資料轉換、以及繪製圖表。

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

print(pd.__version__)
```

In [ ]:

```
from pathlib import Path
data_folder = Path("../data/")
```

Input and Output

Pandas 提供許多常見類型資料的輸入和輸出，如 csv、json、excel、html、sql、pickle等。

更多關於 Pandas IO Tools: https://pandas.pydata.org/pandas-docs/stable/io.html (https://pandas.pydata.org/pandas-docs/stable/io.html)

In [ ]:

```
news = pd.read_csv(data_folder / "news.csv")
```

In [ ]:

```
polls = pd.read_json(data_folder / "polls.json")
```

pandas.DataFrame.to_csv

```
In [ ]: polls.to_csv(data_folder / "polls.csv", index=False)
```

Pickle (Serialization)

```
In [ ]: polls.to_pickle(data_folder / "polls.p")
        polls_pickle = pd.read_pickle(data_folder / "polls.p")
```

Examining DataFrame

讀進來資料後，可以使用 `.info()` 看數據資料型別、 `.head()` 選出前幾筆資料、 `.tail()` 選出最後幾筆資料、 `.columns` 來看欄位名稱、 `.index` 來看索引名稱。

```
In [ ]: news.info()
```

```
In [ ]: news.head()
```

```
In [ ]: news.tail(2)
```

```
In [ ]: polls.info()
```

```
In [ ]: polls.columns
```

```
In [ ]: polls.index
```

Indexing and Slicing

可以選出特定部分的數據表格和欄位。

```
In [ ]: # df[column]
        news['title']
```

```
In [ ]: # df.column
        news.title
```

```
In [ ]: # df.loc[row_name, column_name]
        polls.loc[10, '支持率']
```

```
In [ ]: # df.iloc[row_number, column_number]
        polls.iloc[10, 0]
```

In [ ]:
```
# use : to select a range
polls.iloc[0:3, 1:4]
```

In [ ]:
```
polls.iloc[1:4, :]
```

In [ ]:
```
# use list to select a multiple items
polls.loc[[1,5,2], ['機構','時間']]
```

Conditional

可以選出滿足特定條件的數據表格。

In [ ]:
```
polls['機構'] == 'TVBS'
```

In [ ]:
```
polls[polls['機構'] == 'TVBS']
```

Changing Data Type

可以將資料轉成適當的資料型別。

In [ ]:
```
# polls['有效樣本'] > 1000
```

In [ ]:
```
polls.info()
```

In [ ]:
```
polls['有效樣本'] = pd.to_numeric(polls['有效樣本'], errors='coerce')
polls.info()
```

In [ ]:
```
polls[polls['有效樣本'] > 1000]
```

In [ ]:
```
polls['時間'] = pd.to_datetime(polls['時間'])
polls.info()
```

In [ ]:
```
polls[polls['時間'] > '2018-07-01']
```

In [ ]:
```
polls['機構'] = polls['機構'].astype('category')
polls['訪問主題'] = polls['訪問主題'].astype('category')
polls.info()
```

Sort

可以將資料排序。

In [ ]:
```
polls.sort_values('有效樣本')
```

In [ ]:

```
polls.sort_values('有效樣本', ascending=False)
```

In [ ]:

```
# df.set_index([column1, column2])
polls.set_index('時間')
```

In [ ]:

```
polls_arranged = polls.set_index(['訪問主題','時間'])
polls_arranged.head()
```

In [ ]:

```
polls_arranged.sort_index()
```

Apply and Applymap

可以對數據做一致的轉換。

In [ ]:

```
taipei = polls[polls['訪問主題'] == '台北市長當選人']
print(taipei.head())
```

In [ ]:

```
taipei.iloc[0,0]
```

In [ ]:

```
pd.Series(taipei.iloc[0,0])
```

In [ ]:

```
percentage = taipei['支持率'].apply(pd.Series)
```

In [ ]:

```
percentage.head()
```

In [ ]:

```
percentage = percentage.applymap(lambda x: float(x.strip('%')))
```

In [ ]:

```
percentage.head()
```

In [ ]:

```
taipei = pd.concat([percentage, taipei.iloc[:,1:]], axis=1)
```

In [ ]:

```
taipei.head()
```

In [ ]:

```
taipei = taipei.set_index('時間')
```

In [ ]:

```
taipei.head()
```

Plot

可以直接繪製成圖表。

In [ ]:

```
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:

```
# find your font path here
fpath = "/Library/Fonts/Microsoft/Microsoft Jhenghei.ttf"
zhfont = mpl.font_manager.FontProperties(fname=fpath, size=14)
sns.set(font=zhfont.get_family())
sns.set_style("whitegrid", {"font.sans-serif": ['Microsoft JhengHei']})
```

In [ ]:

```
taipei.iloc[:,[0,2,3]].plot()
```

In [ ]:

```
ax = taipei.iloc[:,[0,2,3]].plot()
ax.set_title('台北市長當選人')
ax.set_ylabel('支持率')
```

Exercises and Solutions

▶ 1. 從 `polls` 選出 `訪問主題` 為 `高雄市長當選人` 的表格

▶ 2. 從 1. 得到的表格中，將 `支持率` 轉變為一個新的表格 (提示 `apply`)

▶ 3. 將 2. 得到的表格中，把每格的百分比轉為 `float` (提示 `applymap`)

▶ 4. 將 1. 和 3. 得到的表格合併

▶ 5. 設定 4. 的表格的 `index` 為 `時間`

▶ 6. 選取 `2018-08-01` 之後的表格

▶ 7. 畫出兩位候選人隨著時間的支持度變化

More about:

1. 10 Minutes to pandas (https://pandas.pydata.org/pandas-docs/stable/10min.html)
2. Pandas tutorial (http://pandas.pydata.org/pandas-docs/stable/tutorials.html)

Matplotlib https://matplotlib.org (https://matplotlib.org)
Seaborn https://seaborn.pydata.org (https://seaborn.pydata.org)