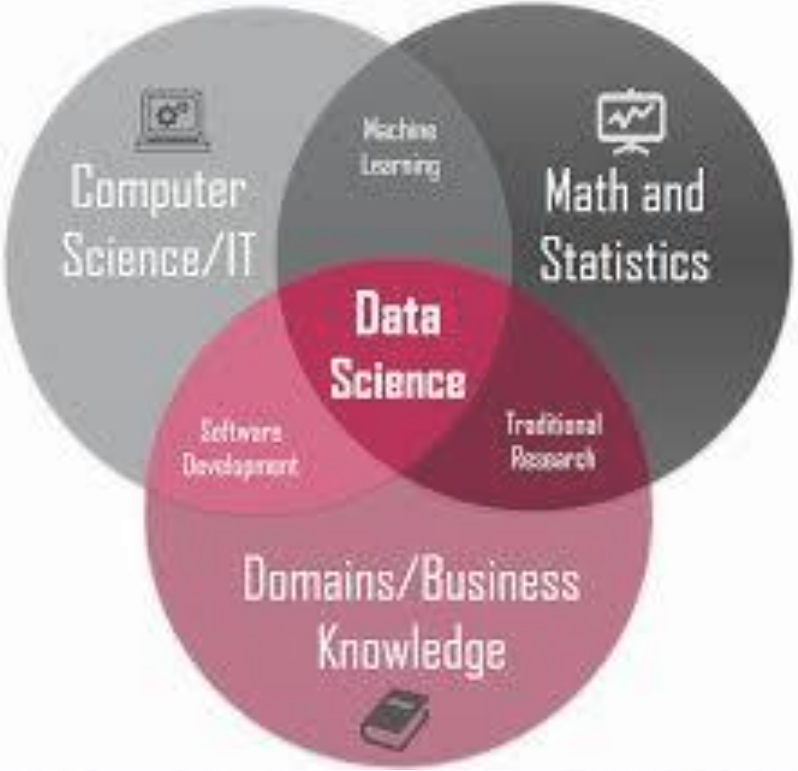
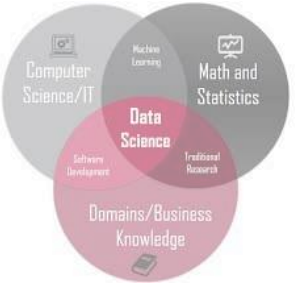
AI & DS



U05-資料預先處理與清洗
Pandas

2023.04_V1.0

單元大綱



※使用Pandas 套件

- 資料清洗處理
- 資料檢查
- 資料合併
- 樞紐分析表
- (*)圖片增量

※使用Keras套件

- ImageDataGenerator模組
- augmentor模組

[資料引用]

Part 1 資料清洗處理

資料預先處理

- 資料預先處理注意事項:

1.資料清洗

- 前置處理工作
- 缺失值處理
- 重複值處理
- 異常值處理

2.資料檢查

- 資料範圍檢查
- 資料格式檢查

學號	姓名	性別	電郵	國文	英文	數學	歷史	地理
s1110001	李彥文	男	1110001@happy.edu.tw	90	85	98	88	96
s1110002	吳佳樺	女	1110002@happy.edu.tw	75		86	82	89
s1110003	劉豪春	男	1110003happy.edu.tw	83	88	74	75	83
s1110004	趙人豪	男	1110004@happy.edu.tw		85	80	88	82
s1110005	杜清芷	女	1110005@happy.edu.tw	51	66	51	77	
s1110006	吳雅婷	女	1110006@happy	94	62	90	62	85
s1110007	周常明	男	1110008@happy.edu.tw	74	84	94	82	99
s1110007	周常明	男	1110008@happy.edu.tw	74	84	94	82	99
s1110009	張呈康	男	1110007@happy.edu.tw	-1	-1	-1	-1	-1
s1110010	林芳仲	男	1110010@happy.edu.tw	85		88	80	82
s1110011	林喬羽	男	1110011@happy.edu.tw	45	86	89	97	68
s1110012	張家賢	男	1110012@happy.edu.tw	89	71	73		90
s1110013	許淑芬	女	1110013@happy.edu.tw	47	83	87	71	56
s1110014	林美萱	女	1110014@happy.edu.tw	-1	96	88	89	99
s1110015	楊惠玟	女	1110015@happy.edu.tw	77	85	93	98	80
s1110016	林瓊文	女	1110016@happy.edu.tw	91	88	60	54	46
s1110017	連旺美	女	1110017@happy.edu.tw	92	86	83	93	98
s1110018	李岳玟	女	1110018@happy.edu.tw	65	65	59	46	93
s1110019	李怡璇	女	1110019@happy.edu.tw	68	-1	45	58	91

在《學生月考成績檔.csv》檔案中,有沒有看到 dirty data?

資料預先處理

- 資料預先處理注意事項:

1.資料檢查

2.資料清洗

函式	功能說明
info()	顯示 DataFrame 所包含的內容資訊
describe()	顯示 DataFrame 資料相關統計數據
isnull()	篩選資料欄位為空值的記錄
dropna()	刪除資料欄位為空值的記錄
fillna()	填值到空值欄位
duplicated()	檢測 DataFrame 中重複的記錄
drop_duplicates()	刪除重複的記錄

資料讀取預覽

- 使用 Pandas 套件的 read_csv() 讀取*.csv檔的資料
- 將資料讀入DataFrame物件(df)
- 使用head ()顯示內容

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/資料科學自學聖經/ch05/學生月考成績檔.csv')
df.head(10)#顯示前10筆
```

	學號	姓名	性別	電郵	國文	英文	數學	歷史	地理
0	s1110001	李彥文	男	1110001@happy.edu.tw	90.0	85.0	98.0	88.0	96.0
1	s1110002	吳佳樺	女	1110002@happy.edu.tw	75.0	NaN	86.0	82.0	89.0
2	s1110003	劉豪春	男	1110003happy.edu.tw	83.0	88.0	74.0	75.0	83.0
3	s1110004	趙人豪	男	1110004@happy.edu.tw	NaN	85.0	80.0	88.0	82.0
4	s1110005	杜清芷	女	1110005@happy.edu.tw	51.0	66.0	51.0	77.0	NaN
5	s1110006	吳雅婷	女	1110006@happy	94.0	62.0	90.0	62.0	85.0
6	s1110007	周常明	男	1110008@happy.edu.tw	74.0	84.0	94.0	82.0	99.0
7	s1110007	周常明	男	1110008@happy.edu.tw	74.0	84.0	94.0	82.0	99.0
8	s1110009	張呈康	男	1110007@happy.edu.tw	-1.0	-1.0	-1.0	-1.0	-1.0
9	s1110010	林芳仲	男	1110010@happy.edu.tw	85.0	NaN	88.0	80.0	82.0

取得資料集的資料摘要: df.info()

- 使用 df.info() 取得DataFrame簡要摘要:索引,資料型態,非空值和記憶體使用

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   學號      30 non-null     object
 1   姓名      30 non-null     object
 2   性別      30 non-null     object
 3   電郵      30 non-null     object
 4   國文      28 non-null     float64
 5   英文      28 non-null     float64
 6   數學      29 non-null     float64
 7   歷史      29 non-null     float64
 8   地理      29 non-null     float64
dtypes: float64(5), object(4)
memory usage: 2.2+ KB
```

資料筆數與索引值範圍

資料欄位數

資料欄位資訊

資料欄位內容(資料綱要)

記憶體使用

取得資料集的資料摘要:df.describe()

- 使用 df. describe()用於產生資料的敘述性統計數據,包括:數據分佈的集中趨勢,分散狀態和形狀,不包含空值。(※若欄位中包含數值及字串,使用df.describe()預設會傳回數值型態的資料欄位摘要資訊!!)

```
df.describe()
```

count:計次,資料筆數

mean:平均值

std:標準差

min:最小值

25%:第1四分位,從小到第25%的數

50%:第2四分位,從小到第50%的數,中位數

25%:第3四分位,從小到第75%的數

max:最大值

	國文	英文	數學	歷史	地理
count	28.000000	28.000000	29.000000	29.000000	29.000000
mean	69.964286	73.678571	76.482759	76.517241	76.896552
std	24.345480	24.331132	21.903393	22.146623	25.866891
min	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
25%	64.750000	66.000000	63.000000	62.000000	70.000000
50%	76.000000	84.000000	86.000000	82.000000	84.000000
75%	85.500000	87.250000	92.000000	93.000000	93.000000
max	94.000000	97.000000	99.000000	99.000000	99.000000

缺失值處理

使用 df.info() 觀察資料集中每個欄位的資料。若每個欄位中非空值的資料筆數,與資料總數不同，其差值就是缺失值的數量。

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   學號      30 non-null     object
 1   姓名      30 non-null     object
 2   性別      30 non-null     object
 3   電郵      30 non-null     object
 4   國文      28 non-null     float64
 5   英文      28 non-null     float64
 6   數學      29 non-null     float64
 7   歷史      29 non-null     float64
 8   地理      29 non-null     float64
dtypes: float64(5), object(4)
memory usage: 2.2+ KB
```

資料總筆數

資料有缺失值

顯示有空值的資料: df.isnull()

使用 df.isnull()顯示欄位有空值的紀錄。

[語法]

條件變數 = DataFrame 變數.isnull().空值型態(axis= 'column')

DataFrame 變數[條件變數]

- **空值型態**: all : 表示所有欄位都是空值才符合條件　　any : 表示任何欄位有空值的資料

```
condition = df.isnull().any(axis='columns')
df[condition]
```

	學號	姓名	性別	電郵	國文	英文	數學	歷史	地理
1	s1110002	吳佳樺	女	1110002@happy.edu.tw	75.0	NaN	86.0	82.0	89.0
3	s1110004	趙人豪	男	1110004@happy.edu.tw	NaN	85.0	80.0	88.0	82.0
4	s1110005	杜清芷	女	1110005@happy.edu.tw	51.0	66.0	51.0	77.0	NaN
9	s1110010	林芳仲	男	1110010@happy.edu.tw	85.0	NaN	88.0	80.0	82.0
11	s1110012	張家賢	男	1110012@happy.edu.tw	89.0	71.0	73.0	NaN	90.0
20	s1110021	林昆輝	男	1110021@happy.edu.tw	69.0	66.0	NaN	95.0	84.0
23	s1110024	吳堯嬌	女	1110024@happy.edu.tw	NaN	55.0	72.0	52.0	52.0

移除空值的資料: df.dropna()

如何處理空值: 1.移除空值資料(僅適用於少量空值資料) 2.填補空值資料

[語法] DataFrame 變數.dropna(how=空值型態, thresh=數值, subset=欄位串列)

- **how** : 非必填。若值=all: 表示所有欄位都是空值才刪除；若值=any(預設): 表示任何欄位有空值就刪除資料。
- **thresh** : 非必填。表示非空值欄位小於參數值就刪除資料。
- **subset** : 非必填。參數值是欄位名稱組成的串列，表示在串列中的欄位若有空值就刪除資料。

```
df = pd.read_csv('/content/drive/MyDrive/資料科學自學聖經/ch05/學生月考成績檔.csv')
df = df.dropna()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23 entries, 0 to 29
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   學號      23 non-null     object
 1   姓名      23 non-null     object
 2   性別      23 non-null     object
 3   電郵      23 non-null     object
 4   國文      23 non-null     float64
 5   英文      23 non-null     float64
 6   數學      23 non-null     float64
 7   歷史      23 non-null     float64
 8   地理      23 non-null     float64
dtypes: float64(5), object(4)
memory usage: 1.8+ KB
```

已刪除空值資料

已沒有缺失資料

空值填補: df.fillna()

[語法] DataFrame 變數.fillna(value=數值, method=填充位置, axis=列或行)

- **value** : 此參數為固定值,亦可使用DataFrame的統計函式來取得計算結果。Ex:mean(), max(), min()...等
- **method** : 此參數是以鄰近資料來填補。參數值為:
  backfill or bfill: 表示以下一個資料填補　　ffill or pad: 表示以上一個資料填補
- **axis** : 非必填。若設定**method**參數可搭配此參數設定填補列或行。
  參數值: index 或 0 表示以列資料填補,此為預設值; 參數值: column 或 1 表示以行資料填補

※ **value**或 **method** 參數只能擇一, 否則會出現錯誤!!

- 將所有空值資料以該欄位的平均值填補:

```
df = pd.read_csv('/content/drive/MyDrive/資料科學自學聖經/ch05/學生月考成績檔.csv')
df['國文'] = df['國文'].fillna(df['國文'].mean())
df['英文'] = df['英文'].fillna(df['英文'].mean())
df['數學'] = df['數學'].fillna(df['數學'].mean())
df['歷史'] = df['歷史'].fillna(df['歷史'].mean())
df['地理'] = df['地理'].fillna(df['地理'].mean())
df
```

※這裏是以mean()進行資料填補,
欄位的資料型態必須是「數值」!!

- 若將所有空值資料變更為下一列的資料:

```
df = pd.read_csv('/content/drive/MyDrive/資料科學自學聖經/ch05/學生月考成績檔.csv')
df.fillna(method='backfill')
```

重複資料處理: df.duplicated()

- 顯示重複資料

[語法] df. duplicates()

```
df = pd.read_csv('/content/drive/MyDrive/資料科學自學聖經/ch05/學生月考成績檔.csv')
df[df.duplicated()]
```

	學號	姓名	性別	電郵	國文	英文	數學	歷史	地理
7	s1110007	周常明	男	1110008@happy.edu.tw	74.0	84.0	94.0	82.0	99.0

- 刪除重複資料

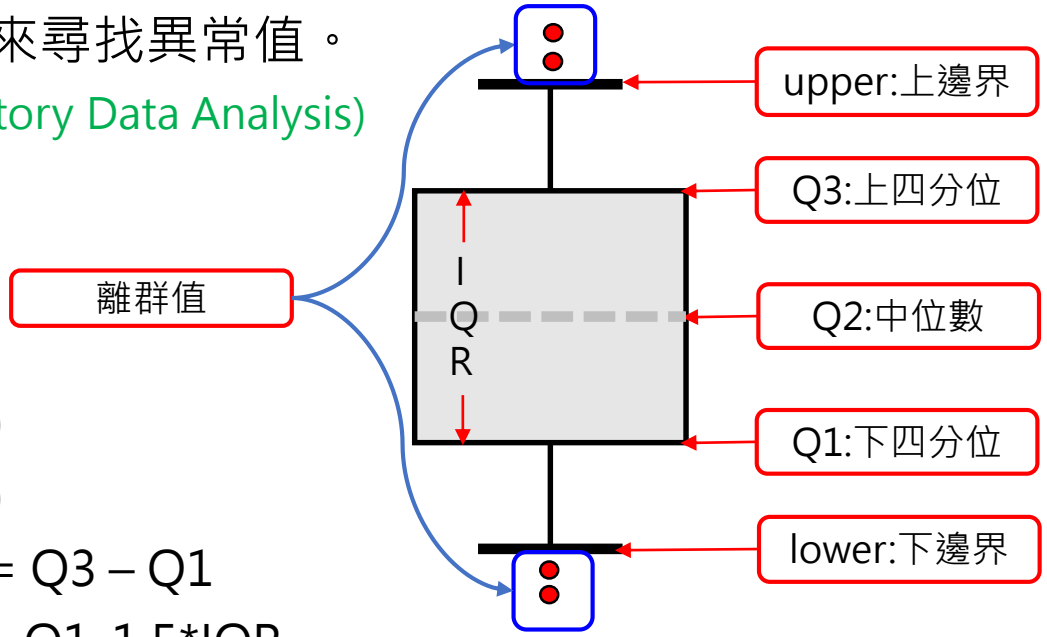
[語法] DataFrame 變數.drop_duplicates([keep=刪除型態, ignore_index=布林值])

- keep : 非必填,設定要保留哪一筆資料。有三種值:
  first (預設):保留第一個重複資料,其餘刪除; last: 保留最一個重複資料,其餘刪除; False: 刪除全部重複資料
- ignore_index : 非必填,設定刪除後是否重新建立索引值。
  True: 會重新建立索引值; False(預設): 不會重新建立索引值。

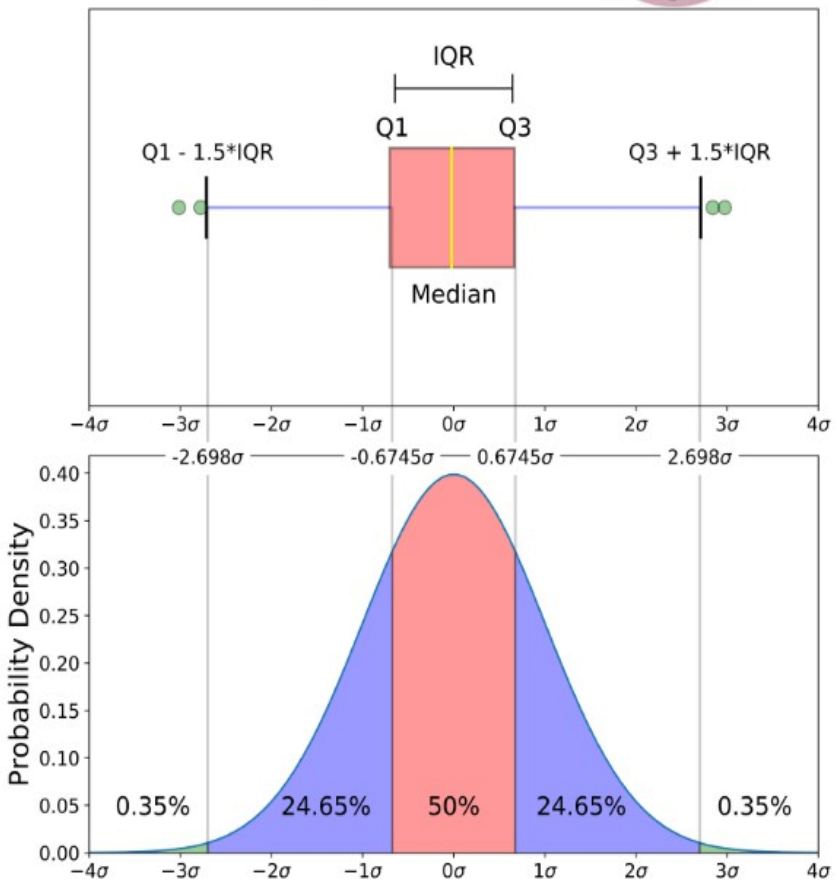
```
df = df.drop_duplicates()
df.info()
```

異常值(離群值)處理

- 異常值(離群值)是指在數值資料中出現某些過大或過小的數值。
- 常用「箱型(盒鬚)圖」來尋找異常值。
  (1977, J. W. TUKEY, Exploratory Data Analysis)



- 演算步驟:
  S-1.將資料由小到大排序
  S-2.計算下四分位數(Q1)
  S-3.計算上四分位數(Q3)
  S-4.計算四分位距(IQR) = Q3 – Q1
  S-5.計算下邊界(lower) = Q1-1.5*IQR
  S-6.計算上邊界(upper) = Q3+1.5*IQR
  S-7.大於上邊界或小於下邊界者皆為異常值



Part 2

資料檢查

範圍檢查

- 範圍檢查是指檢查資料數值是否在合理的範圍內,例如:檢查成績是否在0-100分內。

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/資料科學自學聖經/ch05/學生月考成績檔.csv')
df = df.dropna()
df
```

8	s1110009	張呈康	男	1110007@happy.edu.tw	-1.0	-1.0	-1.0	-1.0	-1.0

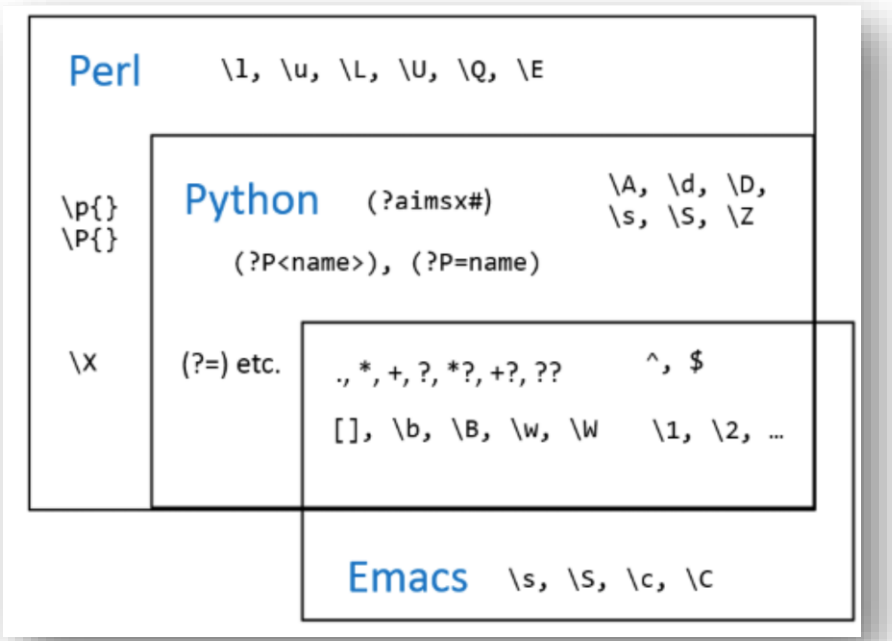
- 使用以下程式檢查所有成績數值, 若成績為負值就更正為0分

```
df.loc[df['國文'] < 0, '國文'] = 0
df.loc[df['英文'] < 0, '英文'] = 0
df.loc[df['數學'] < 0, '數學'] = 0
df.loc[df['歷史'] < 0, '歷史'] = 0
df.loc[df['地理'] < 0, '地理'] = 0
df
```

8	s1110009	張呈康	男	1110007@happy.edu.tw	0.0	0.0	0.0	0.0	0.0

資料格式檢查:正規表達式(regex) (1/2)

- **正規表示法 (又譯正則表示法, 正規表達式)** 是字串處理的瑞士刀, 它使用詮釋字元 (metacharacter) 組成的範本字串來描述符合特定規則的文字, 被廣泛地應用於網路爬蟲, 系統維護, 自動化測試, 以及自然語言處理等大量文本操作場域, 主要用來檢索 (search), 擷取 (extract), 驗證 (validate) 或替換 (replace) 符合特定樣式 (pattern) 的字串。
- 正規表達式起源於數學中的形式語言 (formal language) 理論, 最早可追溯到神經生理學家 Warren McCulloch 與 Walter Pitts 於 1943 年所寫的神經網路論文; 不過 "Regular expression" 一詞卻要到 1956 年數學家 Stephen Kleene 提出有限自動機 (Finite Automata) 的論文才第一次出現, 這個有限自動機就是正規表式法的數學理論模型.
- Perl 語言的設計者 Larry Wall 在 1986 年發布的 Perl 語言函式庫中實作了正規式引擎, 其正規式句法被稱為 PCRE (Perl Compatible Regular Expression) 流派, 後來許多程式語言 (例如 Java, R, Python, PHP, Julia, Ruby 等) 也採用了 PCRE 字元集, 使得 PCRE 漸漸成為主流, 而正規式也成為現代主流程式語言必備的基礎模組.
- Python 最早期的 re 模組基本上屬於 PCRE 流派, 後來加入的 regex 模組則同時兼具 PCRE 與 POSIX 流派之功能. POSIX 風格的正規表達式版本基本上與 PCRE 相容, 但增加了一些額外的功能而且完全支援 Unicode. 有些語言例如 PHP, R, 與 Python 等同時支援 PCRE 與 POSIX 風格的正規表達式引擎。



[Ref]
引用文獻: http://yhhuang1966.blogspot.com/2020/04/python_11.html
引用圖片: https://www.johndcook.com/blog/regex-perl-python-emacs/

資料格式檢查:正規表達式(regex) (2/2)

- **在字串中尋找指定格式子字串的語法為:**

[語法] 字串變數.contains(pat[, case=布林值, na=填補值, regex=布林值])

- pat :要尋找的子字串或正規表達式。regex=False, 表示尋找子字串; regex=True, 表示正規表達式
- case: 非必填, 預設值為:True,表示尋找的子字串要區分大小寫; 若為False則不區分。
- na: 非必填, 表示若為缺失值時,則替換為此參數。
- regex:非必填,預設值為:True,表示 pat參數為正規表達式; False 則表示pat 參數為字串。

- **檢查格式常用的正規表達式(右圖):**

格式	正規表達式
電子郵件	^([a-zA-Z0-9._%-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4})*$
URL 網址	^(?:(https?|ftp):\/\/)?((?:[a-zA-Z0-9.\-]+\.)+(?:[a-zA-Z0-9]{2,4}))((?:/[\w+=%&.~\-]*)*)\??([\w+=%&.~\-]*)$
IP 位址	^((?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?))*$
身份證字號	^[A-Za-z][1-2]\d{8}$
手機號碼	^09\d{2}-?\d{3}-?\d{3}$
Visa 信用卡	^(4[0-9]{12}(?:[0-9]{3})?)*$
MasterCard 信用卡	^(5[1-5][0-9]{14})*$

Part 3

資料合併

資料附加、串接與融合

- Pandas套件提供: append(), concat(), merge() 函式進行資料的附加
- 資料附加:

  [語法] DataFrame 變數1.append(DataFrame 變數2, ignore_index=布林值)

- ignore_index: 非必填,設定資料附加後是否重新建立索引值: True 表示重新建立, False(預設)表示不會重建。
- 資料串接:

  [語法] pd.concat([DataFrame 變數1, DataFrame 變數2], ignore_index=布林值, join =合併方式)

- ignore_index: 非必填,設定資料附加後是否重新建立索引值: True 表示重新建立, False(預設)表示不會重建。
- join: 非必填, 設定值有二:
  outer(預設): 聯集, 會保留所有欄位,與append()效果相同
  inner: 交集, 只會保留共同欄位資料
- 資料融合:

  [語法] pd.merge(DataFrame 變數1, DataFrame 變數2, left_on=欄位, right_on=欄位, on=欄位
  , suffixes=[後綴1, 後綴2,...], how=合併方式)

- left_on: 非必填, 設定第一個資料集合併欄位基準。
  on: 非必填, 設定值二個資料集共同合併欄位基準。
  how: 非必填, 有以下四個值:
  outer: 聯集, 會保留所有欄位。
  left: 會保留第一個資料集所有欄位。

  right_on:非必填, 設定第二個資料集合併欄位基準。
  suffixes:非必填, 設定值是一個串列。

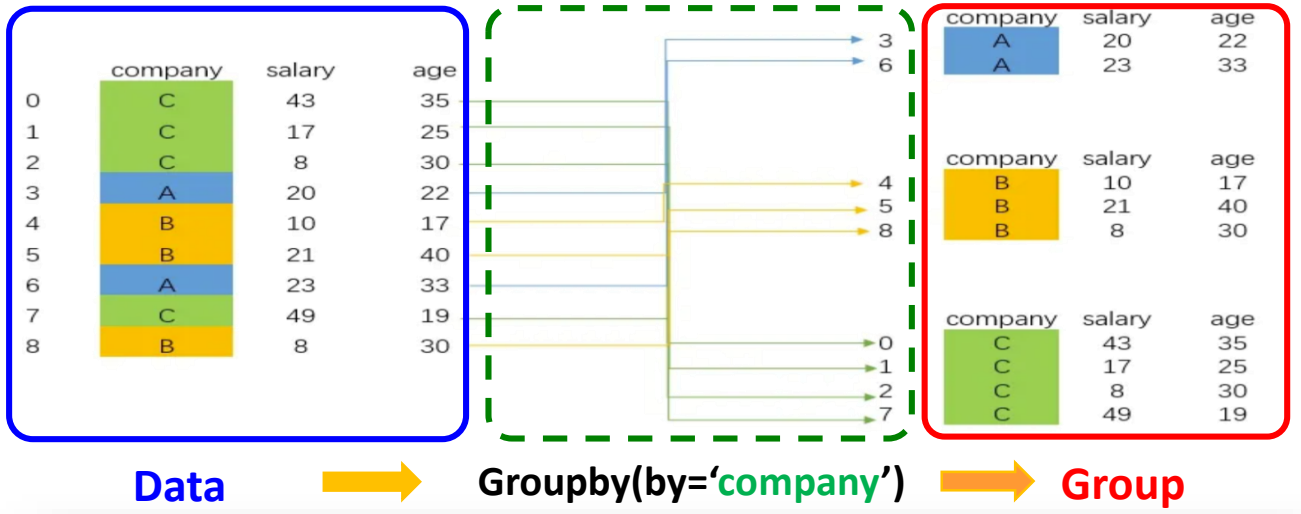
  Inner(預設): 交集, 只會保留共同欄位。
  right: 會保留第二個資料集所有欄位。

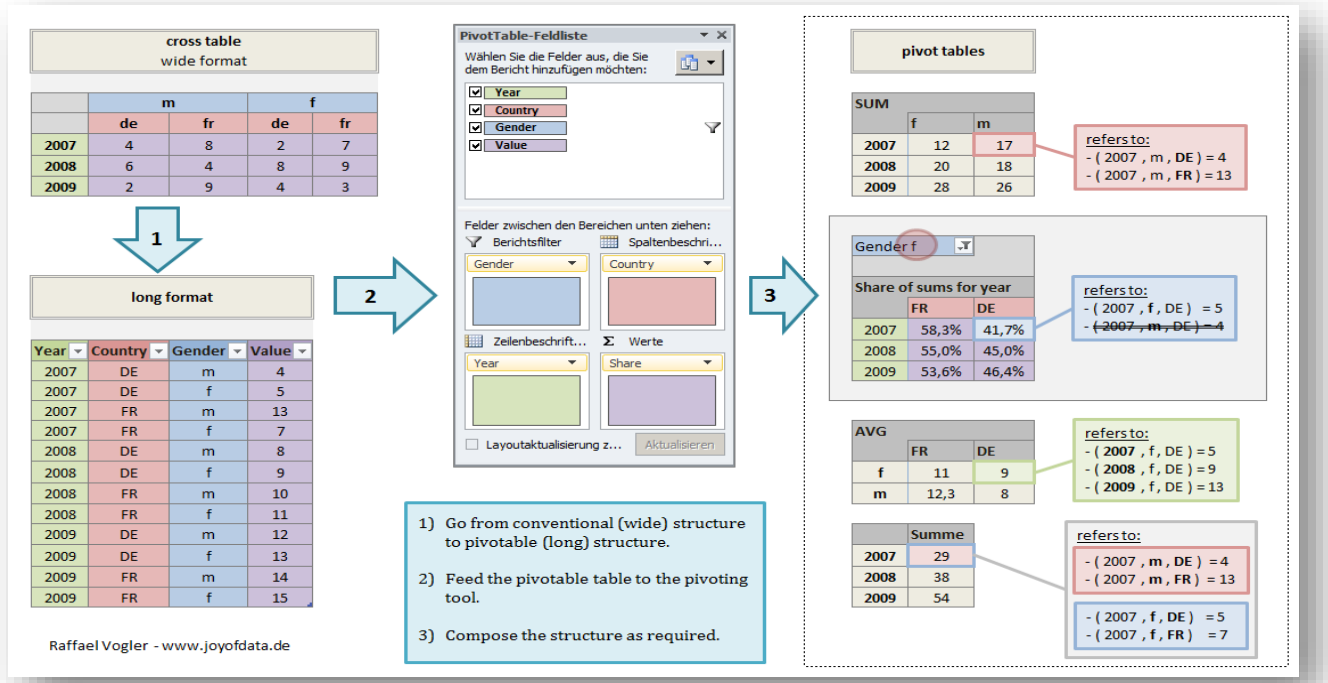
Part 4

樞紐分析表

Pivot Analysis 樞紐分析

- 在結構化的資料(例如:資料表)中,針對存在於欄位或記錄列的中資料,進行資料的排序(sorting)、篩選(filtering)或是將特定的資料建立資料子集合(subsetting),並不一定容易。
- 透過「**樞紐分析**」(Pivot Analysis),可以以一種簡單且省時的方式組織、篩选和重新篩選以及操作數據。
- 在數據分析中,可利用GroupBy將一維化的分群,但若要做到二維化的分群就不容易,必須使用「樞紐分析」完成。





樞紐分析表

- **樞紐分析表語法為:**

[語法] pd.pivot_table(DataFrame 變數, index=列欄位, column=行欄位,
values=分析欄位, margins=布林值, margins_name=字串,
aggfunc=統計項目, fill_value=值, dropna=布林值)

- **index**: **必填參數**, 是一個串列,功能是設定要分析的「列」欄位。
- **column**: 非必填, 是一個串列,功能是設定要分析的「行」欄位,若有多個列欄位,結果會以巢狀方式呈現。
- **values**: 非必填,是一個串列,功能是設定要進行統計的欄位。
- **margins** : 非必填,True表示要計算總和, False(預設)表示不要計算總和。
- **margins_name** : 非必填,當**margins**=True才有效, 功能是設定總和欄位的名稱。預設值為: All。
- **aggfunc(聚合函數)**: 非必填,是一個串列,功能是設定要統計的項目。常用的統計項目有:
  mean(平均)(預設), sum(總和), max(極大值), min(極小值), count(次數)。
- **fill_value**: 非必填,功能是設定資料為空值時以此設定值為優先, 預設值為; None。
- **dropna** : 非必填,True(預設),表示要刪除空值資料, False表示不刪除空值資料。

樞紐分析(Pivot Analysis)

Lab05:鐵達尼號旅客存活分析

資料解析

- **目標**：如何利用Seaborn 內建的鐵達尼號資料(titanic.csv),從當中分析其存活機率。
- **資料來源**:
  https://github.com/mwaskom/seaborn-data
- **歷史紀錄**:
  https://pse.is/4v93rb