Pandas

Pandas

ย่อมาจาก Panel Data เป็นไลบลารีแบบ Open source ของ Python ซึ่งเป็นไลบลารีที่ช่วยในการเก็บข้อมูลมาวิเคราะห์ให้ง่ายขึ้น โดยลักษณะการเก็บจะมี Series (1 มิติ) , DataFrame (2 มิติ) และ Panel(3 มิติ)

การนำเข้า Pandas เพื่อใช้งานและตั้งชื่อว่า pd

```
In [22]: import pandas as pd
```

Series

เป็นข้อมูลอะเรย์ 1 มิติประกอบไปด้วย index และ data

```
In [24]: s = pd.Series([5,20,30,35,60],index=['a','b','c','d','e']);

In [25]: s

Out[25]: a     5
         b    20
         c    30
         d    35
         e    60
         dtype: int64
```

Series

การเข้าถึง

```
In [25]: s

Out[25]: a     5
         b    20
         c    30
         d    35
         e    60
         dtype: int64

In [26]: s[1]

Out[26]: 20
```

```
In [27]: s['c']

Out[27]: 30

In [29]: s[1:4]

Out[29]: b    20
         c    30
         d    35
         dtype: int64
```

DataFrame

เป็นข้อมูลอะเรย์ 2 มิติประกอบไปด้วย index , column และ data

```
In [31]: df1 = pd.DataFrame(np.random.randint(10,50,[5,4]))
         df1
```

Out[31]:

	0	1	2	3
0	17	46	20	12
1	29	40	33	45
2	43	35	17	30
3	30	12	38	45
4	37	24	46	31

DataFrame

In [33]:
```
df2=pd.DataFrame(np.random.randint(10,50,[5,4]) ,
                 index = 'a b c d e'.split() ,
                 columns = list('WXYZ'))
df2
```

Out[33]:

	W	X	Y	Z
a	39	45	31	25
b	22	19	23	18
c	27	30	48	33
d	17	17	34	33
e	18	49	37	24

DataFrame

การเข้าถึง

```
In [35]: df2['W']

Out[35]: a    39
         b    22
         c    27
         d    17
         e    18
         Name: W, dtype: int32
```

```
In [38]: df2.W

Out[38]: a    39
         b    22
         c    27
         d    17
         e    18
         Name: W, dtype: int32
```

```
In [40]: df2.loc['b']

Out[40]: W    22
         X    19
         Y    23
         Z    18
         Name: b, dtype: int32
```

```
In [41]: df2.iloc[1]

Out[41]: W    22
         X    19
         Y    23
         Z    18
         Name: b, dtype: int32
```

DataFrame

การเข้าถึง

```
In [45]: df2[['W','Z']]
```

Out[45]:

	W	Z
a	39	25
b	22	18
c	27	33
d	17	33
e	18	24

```
In [49]: df2.loc[['a','b']]
```

Out[49]:

	W	X	Y	Z
a	39	45	31	25
b	22	19	23	18

DataFrame

.loc() เป็นการเข้าถึงแบบอิงจากชื่อ

.iloc() เป็นการเข้าถึงแบบอิงจากตัวแหน่ง

โดยทั้งสองจะสามารถเลือกแถวและคอลัมน์ได้

[แถว , คอลัมน์] ถ้าไม่ใส่ , จะเป็นแค่เลือกแถว

*** List หรือ : ก็ได้

DataFrame


```
In [50]: df2
```

Out[50]:

	W	X	Y	Z
a	39	45	31	25
b	22	19	23	18
c	27	30	48	33
d	17	17	34	33
e	18	49	37	24

```
In [51]: df2.loc[['c','e']]
```

Out[51]:

	W	X	Y	Z
c	27	30	48	33
e	18	49	37	24

```
In [53]: df2.loc['c':'e']
```

Out[53]:

	W	X	Y	Z
c	27	30	48	33
d	17	17	34	33
e	18	49	37	24

```
In [54]: df2.loc['c':'e',['X','Y']]
```

Out[54]:

	X	Y
c	30	48
d	17	34
e	49	37

```
In [55]: df2.loc[:,'W':'Y']
```

Out[55]:

	W	X	Y
a	39	45	31
b	22	19	23
c	27	30	48
d	17	17	34
e	18	49	37

```
In [58]: df2.loc['b','X']
```

Out[58]: 19

DataFrame


```
In [61]: df = df2
```

```
In [62]: df
```

Out[62]:

	W	X	Y	Z
a	39	45	31	25
b	22	19	23	18
c	27	30	48	33
d	17	17	34	33
e	18	49	37	24

```
In [63]: df>30
```

Out[63]:

	W	X	Y	Z
a	True	True	True	False
b	False	False	False	False
c	False	False	True	True
d	False	False	True	True
e	False	True	True	False

```
In [64]: df[df>30]
```

Out[64]:

	W	X	Y	Z
a	39.0	45.0	31.0	NaN
b	NaN	NaN	NaN	NaN
c	NaN	NaN	48.0	33.0
d	NaN	NaN	34.0	33.0
e	NaN	49.0	37.0	NaN

```
In [66]: df['Z']>30
```

Out[66]:

```
a    False
b    False
c     True
d     True
e    False
Name: Z, dtype: bool
```

```
In [67]: df[df['Z']>30]
```

Out[67]:

	W	X	Y	Z
c	27	30	48	33
d	17	17	34	33

DataFrame


```
In [69]: df[df['Z']>30]['X']

Out[69]: c    30
         d    17
         Name: X, dtype: int32

In [70]: df[df['Z']>30].loc['c']

Out[70]: W    27
         X    30
         Y    48
         Z    33
         Name: c, dtype: int32
```

DataFrame

In [71]:
```
data = {
    'Company':['SCG','SCG','PTT','PTT','KTB','KTB' ] ,
    'Person' :list("ABCDEF") ,
    'Sales'  : [200,300,450,100,140,350]
    }
```

In [73]:
```
df = pd.DataFrame(data)
df
```

Out[73]:

	Company	Person	Sales
0	SCG	A	200
1	SCG	B	300
2	PTT	C	450
3	PTT	D	100
4	KTB	E	140
5	KTB	F	350

DataFrame

groupby() คือการรวมค่าที่เหมือนกันในคอลัมนั้นๆ

```
In [74]: df.groupby('Company')
Out[74]: <pandas.core.groupby.generic.DataFrameGroupBy object at 0x0000022521BEDC10>

In [75]: df.groupby('Company').sum()
Out[75]:
```

Company	Sales
KTB	490
PTT	550
SCG	500

```
In [76]: df.groupby('Company').mean()
Out[76]:
```

Company	Sales
KTB	245
PTT	275
SCG	250

DataFrame


```
In [78]: df.groupby('Company').describe()
```

Out[78]:

	Sales							
	count	mean	std	min	25%	50%	75%	max
Company								
KTB	2.0	245.0	148.492424	140.0	192.5	245.0	297.5	350.0
PTT	2.0	275.0	247.487373	100.0	187.5	275.0	362.5	450.0
SCG	2.0	250.0	70.710678	200.0	225.0	250.0	275.0	300.0

```
In [79]: df.groupby('Company').describe().transpose()
```

Out[79]:

	Company	KTB	PTT	SCG
Sales	count	2.000000	2.000000	2.000000
	mean	245.000000	275.000000	250.000000
	std	148.492424	247.487373	70.710678
	min	140.000000	100.000000	200.000000
	25%	192.500000	187.500000	225.000000
	50%	245.000000	275.000000	250.000000
	75%	297.500000	362.500000	275.000000
	max	350.000000	450.000000	300.000000

DataFrame

unique() คือการหาค่าที่ไม่ซ้ำ

value_counts() คือการนับจำนวนของแต่ละตัว (ที่ไม่ซ้ำกัน)

```
In [81]: df['Company'].unique()

Out[81]: array(['SCG', 'PTT', 'KTB'], dtype=object)

In [82]: df['Company'].nunique()

Out[82]: 3

In [83]: df['Company'].value_counts()

Out[83]: SCG    2
         PTT    2
         KTB    2
         Name: Company, dtype: int64
```

DataFrame


```
In [84]: df[ df['Sales'] == 200   ]
```

Out[84]:

	Company	Person	Sales
0	SCG	A	200

```
In [85]: df['Sales'].max()
```

Out[85]: 450

```
In [86]: df[ df['Sales'] == df['Sales'].max() ]
```

Out[86]:

	Company	Person	Sales
2	PTT	C	450

DataFrame

apply() คือการนำฟังก์ชันเข้าไปคำนวนกับคอลัน์นั้นๆ

```
In [87]: def fn(x):
             return x*2

In [88]: df['Sales'].apply(fn)

Out[88]: 0    400
         1    600
         2    900
         3    200
         4    280
         5    700
         Name: Sales, dtype: int64

In [89]: df['Sales']

Out[89]: 0    200
         1    300
         2    450
         3    100
         4    140
         5    350
         Name: Sales, dtype: int64
```

```
In [91]: df['Sales'].sum()

Out[91]: 1540
```

DataFrame

การอ่านข้อมูลจากไฟล์ CSV หรือ Excel

```
In [93]: df = pd.read_csv('ExsampleCSV')
         df
```

Out[93]:

	a	b	c	d
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

```
In [99]: df = pd.read_excel('Sample - Superstore.xls')
         df
```

Out[99]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer

DataFrame

.head() นำแค่ 5 แถวบนมาโชว์

.tail() นำ 5 แถวล่างมาโชว์

.info() คือการโชว์รายละเอียด

```
In [100]: df.head()
```

```
In [134]: df.tail()
```

```
In [101]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   datetime64[ns]
 3   Ship Date      9994 non-null   datetime64[ns]
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
```

DataFrame

.sort_values() เป็นการเรียงลำดับจากน้อยไปมาก
(ถ้า ascending=False จะมากไปน้อย)

```
In [103]: df['Profit'].sort_values()

Out[103]: 7772   -6599.9780
          683    -3839.9904
          9774   -3701.8928
          3011   -3399.9800
          4991   -2929.4845
                    ...
          4098    4630.4755
          9039    4946.3700
          4190    5039.9856
          8153    6719.9808
          6826    8399.9760
          Name: Profit, Length: 9994, dtype: float64
```