GIỚI THIỆU VỀ PANDAS

Nguyễn Mạnh Hùng

ĐẠI HỌC GTVT

03-2023

Nội dung

Giới thiệu về thư viện Pandas

- Pandas package (xuất phát từ thuật ngữ "panel data" trong kinh tế lượng) là một công cụ quan trọng đối với những người làm khoa học/phân tích dữ liệu bằng Python.
- Pandas là *xương sống* của hầu hết các dự án về dữ liệu. Với Pandas, ta có thể thực hiện những công việc sau đây:
  1) Tính toán thống kê (*trung bình, tương quan, phân phối, ...*)
  2) Làm sạch dữ liệu (*xoá dữ liệu khuyết thiếu, lọc dữ liệu, ...*)
  3) Trực quan hoá dữ liệu bằng các loại biểu đồ
  4) Lưu dữ liệu đã xử lý ra file
- Pandas đã được cài đặt sẵn cùng với Anaconda. Để sử dụng thư viện Pandas, ta khai báo:

```
import pandas as pd
```

Series

Series là một mảng 1-chiều được gán nhãn để chứa dữ liệu (số, chuỗi, đối tượng Python, ...) Để tạo một Series, ta viết

pd.Series(data, index = index)

trong đó,

- **data** có thể là
  - một từ điển Python
  - một mảng ndarray
  - một số vô hướng
- **index** là một danh sách gồm các nhãn

Series

```
s = pd.Series(np.random.randn(4), index=['a','b','c','d'])
```

```
a    0.882687
b    1.665010
c    0.615606
d   -2.192034
dtype: float64
```

```
dic = {'Mai':8, 'Huy':7, 'Lan':6, 'Việt':6.5}
s = pd.Series(dic)
```

```
Mai     8.0
Huy     7.0
Lan     6.0
Việt    6.5
dtype: float64
```

```
s = pd.Series(5, index = range(4))
```

```
0    5
1    5
2    5
3    5
dtype: int64
```

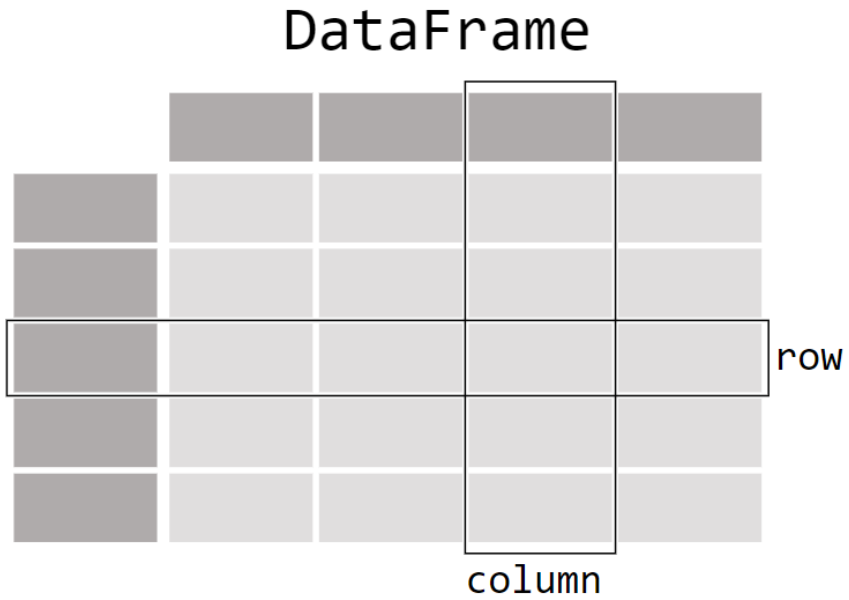
DataFrame

DataFrame là một cấu trúc dữ liệu 2-chiều được gán nhãn, các cột có thể chứa dữ liệu khác kiểu. Để tạo một DataFrame, ta viết:

pd.DataFrame(data, index = index)

DataFrame chấp nhận các kiểu input:

- từ điển gồm các mảng 1-chiều/các danh sách/các từ điển
- mảng 2-chiều ndarray
- một Series
- một DataFrame khác



DataFrame

```
d = {"one": [1.0,2.0,3.0], "two": [4.0,3.0,2.0]}

df = pd.DataFrame(d)
```

	one	two
0	1.0	4.0
1	2.0	3.0
2	3.0	2.0

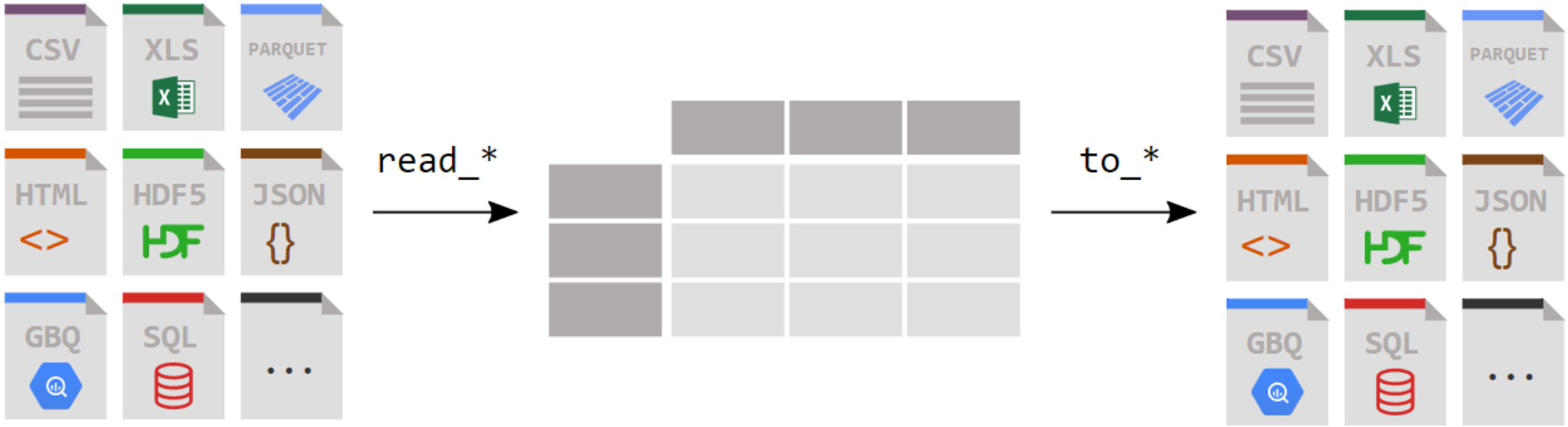
```
d = { "name": pd.Series(["Hạnh","Ngân","Hải"], index=["a","b","c"]),
      "age": pd.Series([18,23,21,15], index=["a","b","c","d"]),
      "sex": pd.Series(["nữ","nữ","nam"], index=["a","b","c"])  }

df = pd.DataFrame(d)
```

	name	age	sex
a	Hạnh	18	nữ
b	Ngân	23	nữ
c	Hải	21	nam
d	NaN	15	NaN

Đọc dữ liệu từ file

- Pandas hỗ trợ đọc/ghi dữ liệu với nhiều định dạng file khác nhau.



- Trong phần còn lại của bài giảng, dữ liệu trong file *core_HRdata.csv* sẽ được sử dụng. Để đọc file, ta dùng lệnh:

```
df = pd.read_csv('datasets\core_HRdata.csv')
```

core_HRdata.csv

	Employee Name	Employee Number	State	Zip	DOB	Age	Sex	MaritalDesc	CitizenDesc	Hispanic/Latino	...	Date of Hire
0	Brown, Mia	1103024456	MA	1450	11/24/1985	32	Female	Married	US Citizen	No	...	10/27/2008
1	LaRotonda, William	1106026572	MA	1460	4/26/1984	33	Male	Divorced	US Citizen	No	...	1/6/2014
2	Steans, Tyrone	1302053333	MA	2703	9/1/1986	31	Male	Single	US Citizen	No	...	9/29/2014
...	...	...	...	...	...	...	...	...	...	...	...	...
298	Szabo, Andrew	1201031324	MA	2140	5/6/1983	34	Male	Single	US Citizen	No	...	7/7/2014
299	True, Edward	1102024057	MA	2451	6/14/1983	34	Male	Single	Non-Citizen	No	...	2/18/2013
300	Sweetwater, Alex	1001644719	MA	2184	11/22/1966	51	Male	Single	US Citizen	No	...	8/15/2011

301 rows × 21 columns

Xem dữ liệu: .head(), .tail()

```
df.head(3)
```

	Employee Name	Employee Number	State	Zip	DOB	Age	Sex	MaritalDesc	CitizenDesc	Hispani
0	Brown, Mia	1103024456	MA	1450	11/24/1985	32	Female	Married	US Citizen	
1	LaRotonda, William	1106026572	MA	1460	4/26/1984	33	Male	Divorced	US Citizen	
2	Steans, Tyrone	1302053333	MA	2703	9/1/1986	31	Male	Single	US Citizen	

3 rows × 21 columns

Xem dữ liệu: .info()

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Employee Name         301 non-null    object 
 1   Employee Number       301 non-null    int64  
 2   State                 301 non-null    object 
 3   Zip                   301 non-null    int64  
 4   DOB                   301 non-null    object 
 5   Age                   301 non-null    int64  
 6   Sex                   301 non-null    object 
 7   MaritalDesc           301 non-null    object 
 8   CitizenDesc           301 non-null    object 
 9   Hispanic/Latino       301 non-null    object 
 10  RaceDesc              301 non-null    object 
 11  Date of Hire          301 non-null    object 
 12  Date of Termination   103 non-null    object 
 13  Reason For Term       301 non-null    object 
 14  Employment Status     301 non-null    object 
 15  Department            301 non-null    object 
 16  Position              301 non-null    object 
 17  Pay Rate              301 non-null    float64
 18  Manager Name          301 non-null    object 
 19  Employee Source       301 non-null    object 
 20  Performance Score     301 non-null    object 
dtypes: float64(1), int64(3), object(17)
memory usage: 49.5+ KB
```

Truy xuất dữ liệu: theo cột

```
df['Age'] # dtype = Series
```

```
0      32
1      33
2      31
3      32
4      29
      ..
296    38
297    31
298    34
299    34
300    51
Name: Age, Length: 301, dtype: int64
```

```
df[['Age','Sex']] # dtype = DataFrame
```

	Age	Sex
0	32	Female
1	33	Male
2	31	Male
...	...	...
298	34	Male
299	34	Male
300	51	Male

301 rows × 2 columns

Truy xuất dữ liệu: theo hàng (.loc/.iloc)

```
# truy xuất theo tên
df.loc[5:10,["Employee Name","Age"]]
```

	Employee Name	Age
5	Smith, Leigh Ann	30
6	LeBlanc, Brandon R	33
7	Quinn, Sean	33
8	Boutwell, Bonalyn	30
9	Foster-Baker, Amy	38
10	King, Janet	63

```
# truy xuất theo số thứ tự
df.iloc[5:11,[0,6,17]]
```

	Employee Name	Sex	Pay Rate
5	Smith, Leigh Ann	Female	20.50
6	LeBlanc, Brandon R	Male	55.00
7	Quinn, Sean	Male	55.00
8	Boutwell, Bonalyn	Female	34.95
9	Foster-Baker, Amy	Female	34.95
10	King, Janet	Female	80.00

Truy xuất dữ liệu: theo điều kiện

```
above50 = df.loc[df.Age>50,["Employee Name","Age","Position","Pay Rate"]]
above50.head(3)
```

	Employee Name	Age	Position	Pay Rate
10	King, Janet	63	President & CEO	80.0
27	Ruiz, Ricardo	54	IT Manager - DB	21.0
45	Favis, Donald	53	Sr. DBA	58.2

```
notmarried = df.loc[df.MaritalDesc.isin(["Single","Divorced"]),
                    ["Employee Name","Age","MaritalDesc"]]
notmarried.head(3)
```

	Employee Name	Age	MaritalDesc
1	LaRotonda, William	33	Divorced
2	Steans, Tyrone	31	Single
4	Singh, Nan	29	Single

Truy xuất dữ liệu: theo điều kiện

```
df.loc[df["Date of Termination"].notna()].shape
```

```
(103, 21)
```

```
df.loc[(df.Age>60) & (df.MaritalDesc=="Single"),
       ["Employee Name","Age","MaritalDesc"]]
```

	Employee Name	Age	MaritalDesc
85	Chace, Beatrice	67	Single
124	Harrington, Christie	65	Single

Ghi chú

Khi kết hợp nhiều mệnh đề điều kiện, mỗi điều kiện phải được đặt trong dấu ngoặc đơn **()**. Hơn nữa, chúng ta không sử dụng toán tử **or/and** mà sử dụng | thay cho **or** và dùng **&** thay cho **and**.

Xử lý trùng lặp (duplicates)

```
above65 = df.loc[df.Age>65,["Employee Name","Age"]]
temp = pd.concat([above65,above65]).reset_index(drop=True)
temp
```

	Employee Name	Age
0	Daniele, Ann	66
1	Chace, Beatrice	67
2	Demita, Carla	66
3	Daniele, Ann	66
4	Chace, Beatrice	67
5	Demita, Carla	66

```
temp.drop_duplicates(inplace=True, keep="last")
temp
```

	Employee Name	Age
3	Daniele, Ann	66
4	Chace, Beatrice	67
5	Demita, Carla	66

subset = [columns]

Đổi tên cột

```
# Đổi tên một cột
above65.rename(columns={"Employee Name":"Name"}, inplace=True)
above65
```

	Name	Age
47	Daniele, Ann	66
85	Chace, Beatrice	67
213	Demita, Carla	66

```
# Đổi tên nhiều cột thành chữ in thường
above65.columns = [col.lower() for col in above65]
above65
```

	name	age
47	Daniele, Ann	66
85	Chace, Beatrice	67
213	Demita, Carla	66

hoặc

```
above65.rename(columns = str.lower)
```

Tạo cột mới từ những cột đã có

```
# Thêm một cột tính số năm quá tuổi hưu 60
above65["exceed"] = above65.age - 60
above65
```

	name	age	exceed
47	Daniele, Ann	66	6
85	Chace, Beatrice	67	7
213	Demita, Carla	66	6

Xử lý khuyết thiếu (missing values)

```
# Xác định khuyết thiếu ở mỗi cột
df.isna().sum()
```

```
Employee Name              0
Employee Number            0
State                      0
Zip                        0
DOB                        0
Age                        0
Sex                        0
MaritalDesc                0
CitizenDesc                0
Hispanic/Latino            0
RaceDesc                   0
Date of Hire               0
Date of Termination      198
Reason For Term            0
Employment Status          0
Department                 0
Position                   0
Pay Rate                   0
Manager Name               0
Employee Source            0
Performance Score          0
dtype: int64
```

```
# Xoá các dòng chứa missing value
df.dropna()

# Xoá cột chứa missing value
df.dropna(axis = 1)

# Thay thế missing value
df["Date of Termination"].fillna("1/1/2023")
```

```
0        1/1/2023
1        1/1/2023
2        1/1/2023
3       4/15/2015
4        1/1/2023
            ...
```

Tính toán các đặc trưng thống kê

```
df[["Age","Pay Rate"]].mean()
```

```
Age         38.548173
Pay Rate    30.715249
dtype: float64
```

```
df[["Age","Pay Rate"]].std()
```

```
Age          8.942884
Pay Rate    15.216214
dtype: float64
```

```
df[["Age","Pay Rate"]].median()
```

```
Age         37.0
Pay Rate    24.0
dtype: float64
```

```
df[["Age","Pay Rate"]].describe()
```

	Age	Pay Rate
count	301.000000	301.000000
mean	38.548173	30.715249
std	8.942884	15.216214
min	25.000000	14.000000
25%	31.000000	20.000000
50%	37.000000	24.000000
75%	44.000000	43.000000
max	67.000000	80.000000

Tính toán các đặc trưng thống kê

```
df[["MaritalDesc","CitizenDesc"]].describe()
```

	MaritalDesc	CitizenDesc
count	301	301
unique	5	3
top	Single	US Citizen
freq	127	285

```
df["MaritalDesc"].value_counts()
```

```
Single        127
Married       119
Divorced       30
Separated      14
widowed        11
Name: MaritalDesc, dtype: int64
```

Tính toán các đặc trưng thống kê

```
# Ma trận hiệp phương sai
df[["Age","Pay Rate"]].cov()
```

	Age	Pay Rate
Age	79.975172	2.285846
Pay Rate	2.285846	231.533170

```
# Ma trận tương quan
df[["Age","Pay Rate"]].corr()
```

	Age	Pay Rate
Age	1.000000	0.016798
Pay Rate	0.016798	1.000000

Apply một hàm vào dữ liệu

```
# Định nghĩa hàm rating
def rating(x):
    if x > 40: return "high"
    elif 24 < x <= 40: return "medium"
    else: return "low"
```

```
# Áp dụng vào cột "Pay Rate" để đánh giá
df["Rating_category"] = df["Pay Rate"].apply(rating)
df[["Employee Name","Pay Rate","Rating_category"]].head()
```

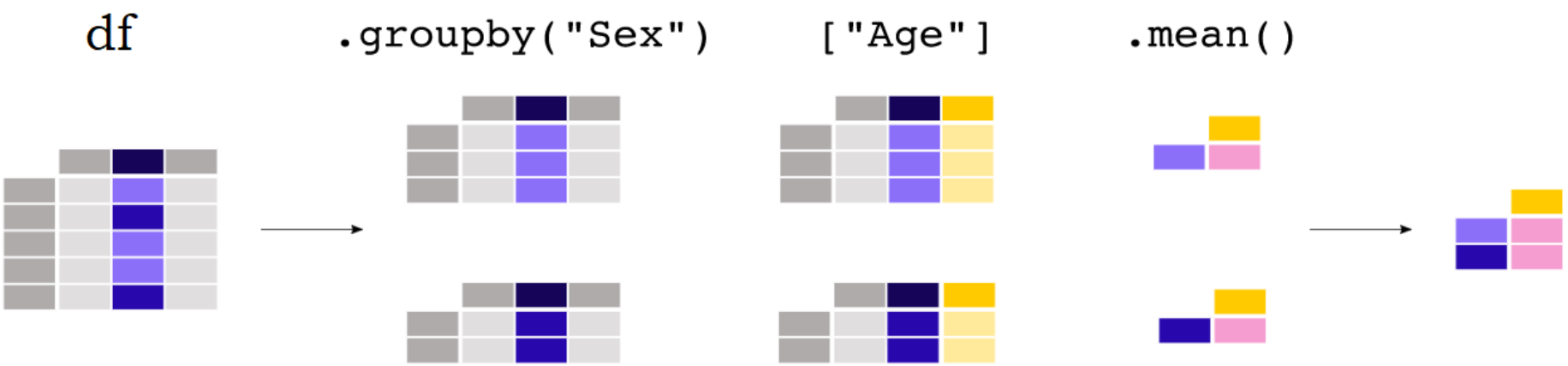
	Employee Name	Pay Rate	Rating_category
0	Brown, Mia	28.50	medium
1	LaRotonda, William	23.00	low
2	Steans, Tyrone	29.00	medium
3	Howard, Estelle	21.50	low
4	Singh, Nan	16.56	low

Thống kê tổng hợp theo nhóm

```
df.groupby("Sex")["Age"].mean()
```

```
Sex
Female    38.563218
Male      38.527559
Name: Age, dtype: float64
```

Split-Apply-Combine pattern



Ghép/nối bảng

```
eligible = df.loc[df.CitizenDesc=="Eligible NonCitizen",
                  ["Employee Name","Age","Sex","CitizenDesc"]]
eligible.head(4)
```

	Employee Name	Age	Sex	CitizenDesc
7	Quinn, Sean	33	Male	Eligible NonCitizen
28	Monroe, Peter	31	Male	Eligible NonCitizen
42	Turpin, Jumil	48	Male	Eligible NonCitizen
77	Beatrice, Courtney	47	Female	Eligible NonCitizen

```
noncitizen = df.loc[df.CitizenDesc=="Non-Citizen",
                    ["Employee Name","Age","Sex","CitizenDesc"]]
noncitizen
```

	Employee Name	Age	Sex	CitizenDesc
89	Cierpiszewski, Caroline	29	Female	Non-Citizen
189	Tavares, Desiree	42	Female	Non-Citizen
224	Homberger, Adrienne J	33	Female	Non-Citizen
299	True, Edward	34	Male	Non-Citizen

Ghép/nối bảng

```
pd.concat([eligible,noncitizen], axis=0).sort_values("Age")
```

	Employee Name	Age	Sex	CitizenDesc
89	Cierpiszewski, Caroline	29	Female	Non-Citizen
28	Monroe, Peter	31	Male	Eligible NonCitizen
7	Quinn, Sean	33	Male	Eligible NonCitizen
166	Power, Morissa	33	Female	Eligible NonCitizen
224	Homberger, Adrienne J	33	Female	Non-Citizen
299	True, Edward	34	Male	Non-Citizen
248	Roberson, May	36	Female	Eligible NonCitizen
212	Davis, Daniel	38	Male	Eligible NonCitizen

Ghép/nối bảng

```
name_age = df.loc[1:3,["Employee Name","Age"]]
name_sex = df.loc[2:4,["Employee Name","Sex"]]
```

	Employee Name	Age
1	LaRotonda, William	33
2	Steans, Tyrone	31
3	Howard, Estelle	32

	Employee Name	Sex
2	Steans, Tyrone	Male
3	Howard, Estelle	Female
4	Singh, Nan	Female

```
pd.merge(name_age, name_sex, on="Employee Name", how="left")
```

	Employee Name	Age	Sex
0	LaRotonda, William	33	NaN
1	Steans, Tyrone	31	Male
2	Howard, Estelle	32	Female

Xử lý dữ liệu thời gian

```
df["DoH"] = pd.to_datetime(df["Date of Hire"])
c = df[["Employee Name","DoH"]].copy()
```

	Employee Name	DoH
0	Brown, Mia	2008-10-27
1	LaRotonda, William	2014-01-06
2	Steans, Tyrone	2014-09-29
3	Howard, Estelle	2015-02-16
4	Singh, Nan	2015-05-01
...	...	...
296	Patronick, Luke	2011-11-07

Xử lý dữ liệu thời gian

```
c["DoH"].min(), c["DoH"].max()
```

```
(Timestamp('2006-01-09 00:00:00'), Timestamp('2016-07-21 00:00:00'))
```

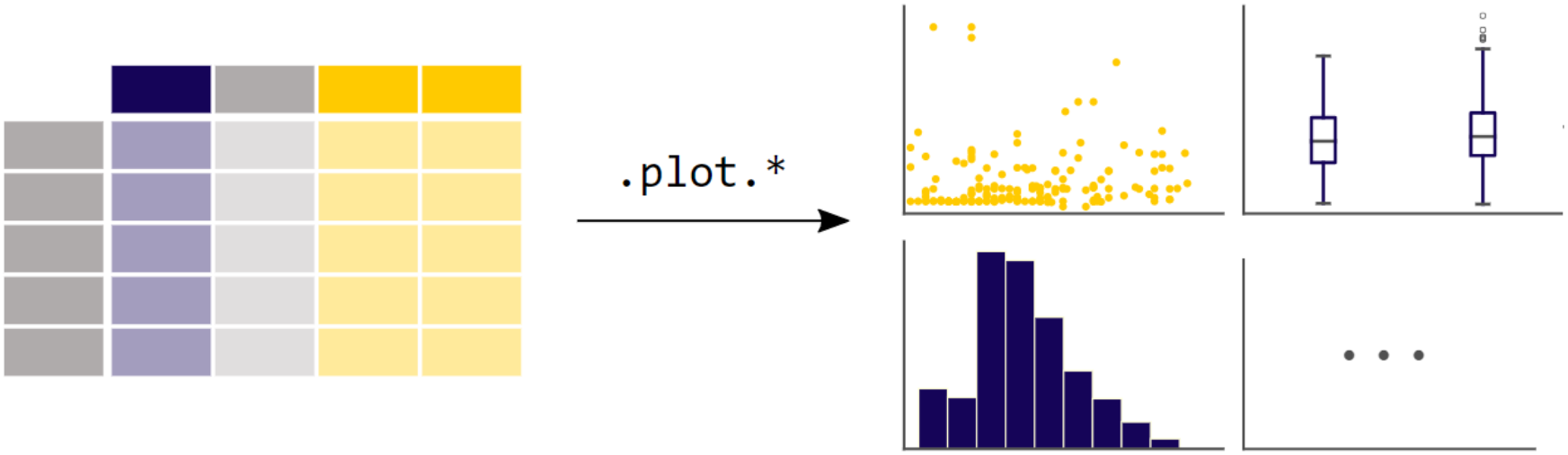
```
c["year"] = c["DoH"].dt.year
c["month"] = c["DoH"].dt.month
c["weekday"] = c["DoH"].dt.day_name()
```

	Employee Name	DoH	year	month	weekday
0	Brown, Mia	2008-10-27	2008	10	Monday
1	LaRotonda, William	2014-01-06	2014	1	Monday
2	Steans, Tyrone	2014-09-29	2014	9	Monday
3	Howard, Estelle	2015-02-16	2015	2	Monday
4	Singh, Nan	2015-05-01	2015	5	Friday
...	...	...	...	...	...

Vẽ biểu đồ

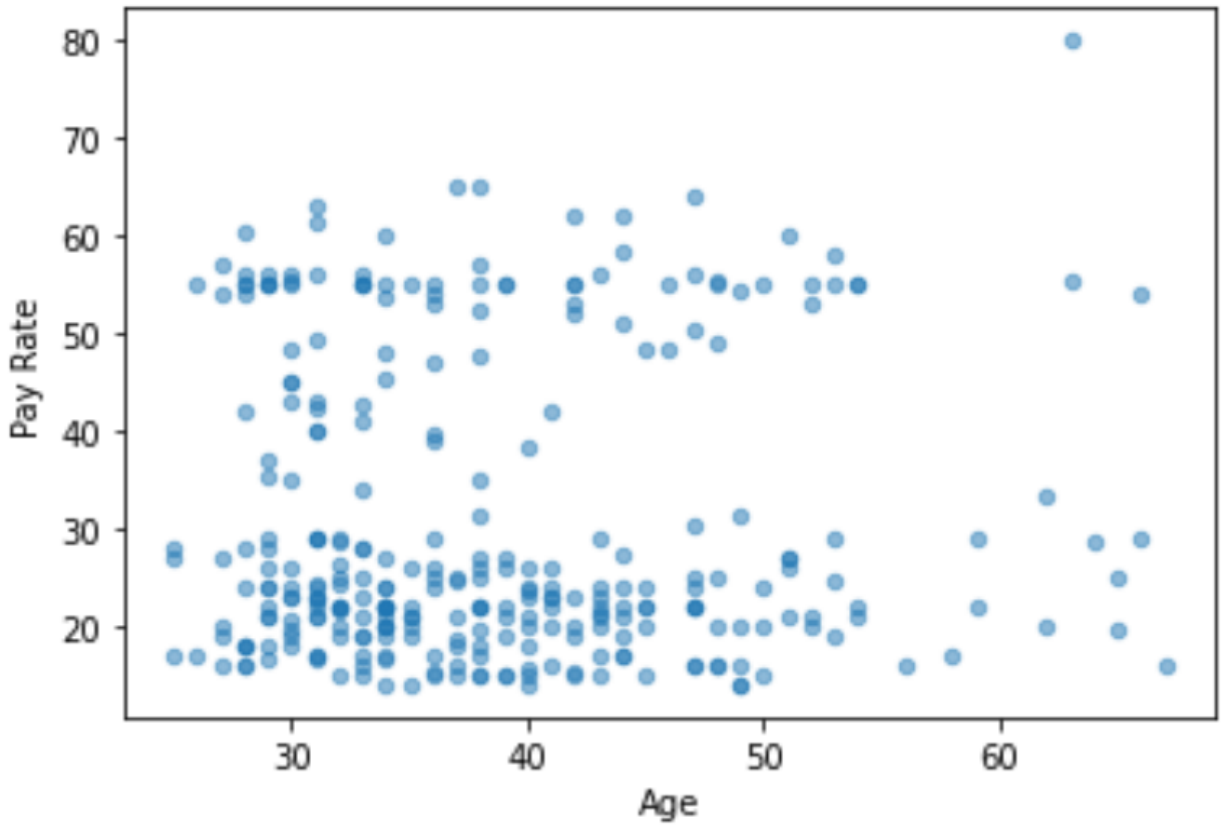
Pandas được tích hợp với Matplotlib cho phép vẽ trực tiếp từ DataFrame và Series. Để bắt đầu, ta khai báo:

```
import matplotlib.pyplot as plt
# Set font and plot size
plt.rcParams.update({'font.size':10,'figure.figsize':(6,4)})
```



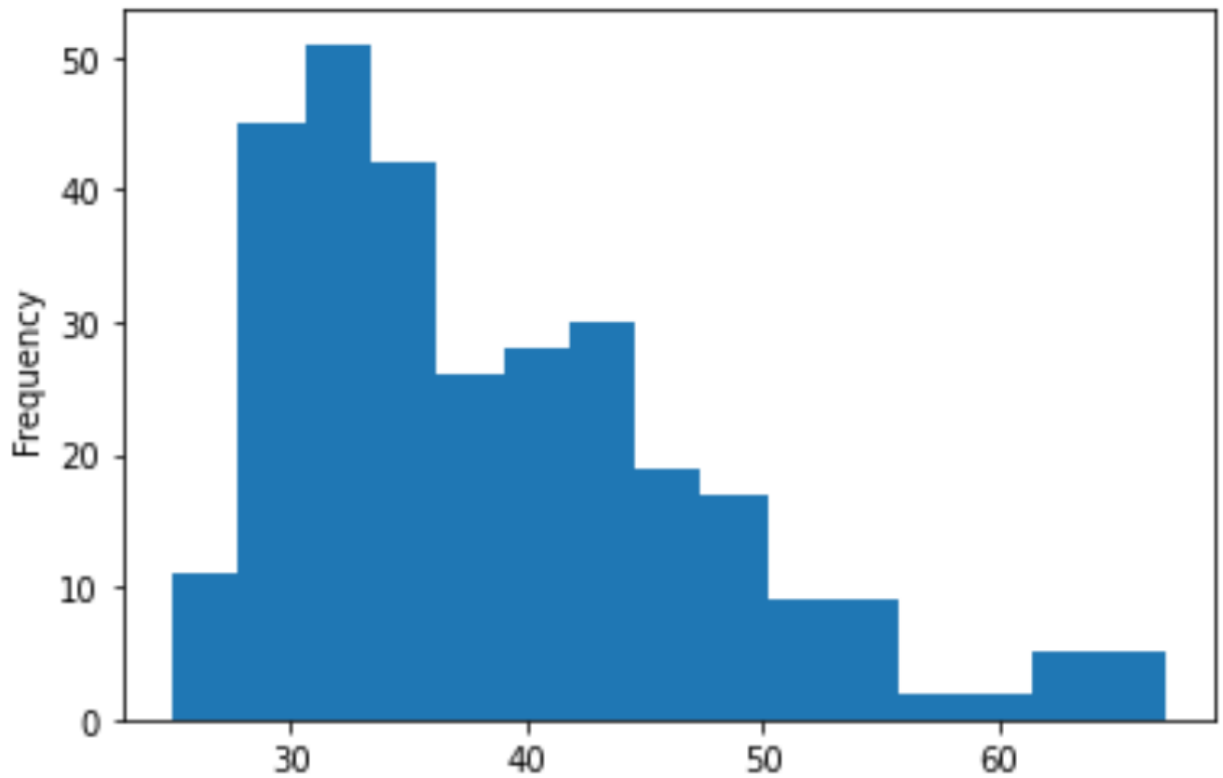
Scatter plot

```
df[["Age","Pay Rate"]].plot.scatter(x="Age", y="Pay Rate", alpha=0.5)
plt.show()
```



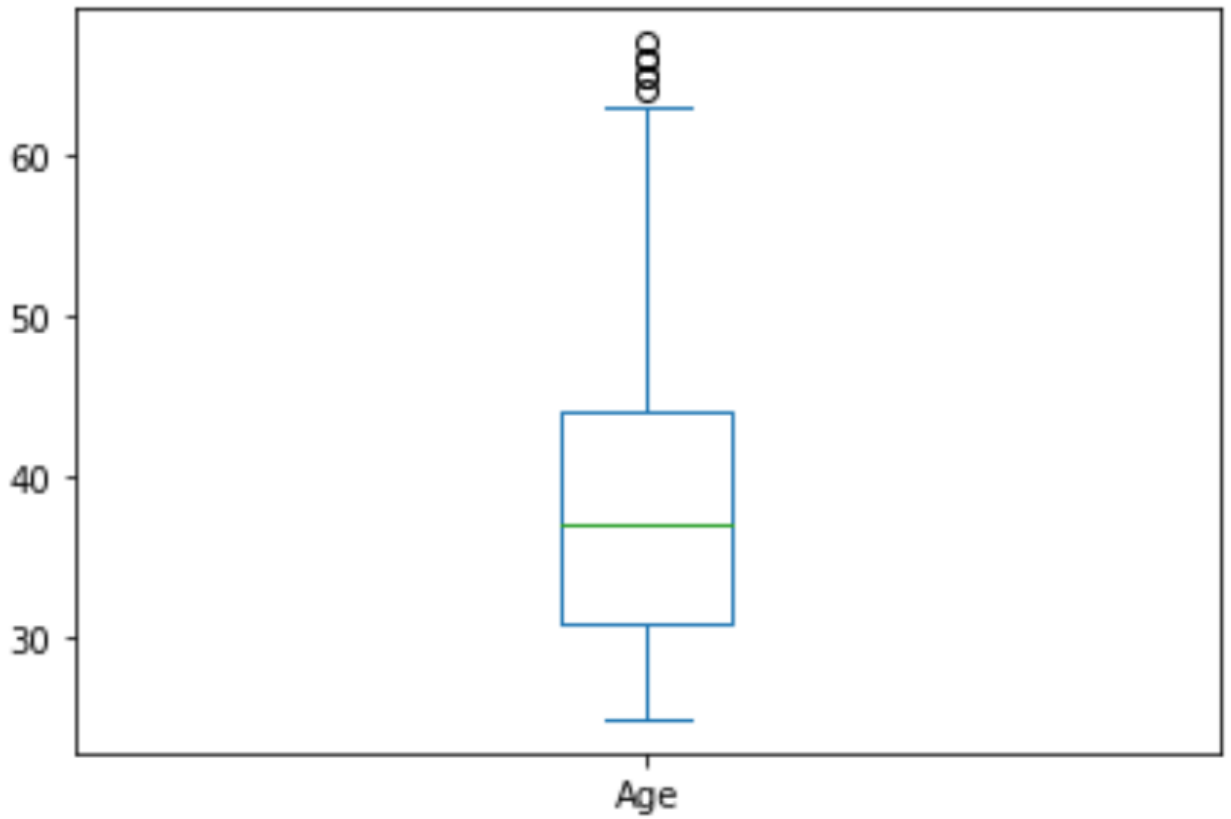
Histogram

```
df["Age"].plot.hist(bins=15)
plt.show()
```



Box plot

```
df["Age"].plot.box()
plt.show()
```



Thực hành 4

Bài tập 4.1

Viết chương trình nhập dữ liệu và tạo danh sách lớp:

- Dữ liệu được nhập từ bàn phím bao gồm: Name, StudentId, Birthday (dd/mm/yyyy). Bỏ trống Name khi muốn thoát. Ví dụ:
  > *Họ tên sinh viên (bỏ trống để thoát): Phạm Hải*
  > *Mã sinh viên: 123456*
  > *Ngày sinh: 01/01/2001*
  > *Họ tên sinh viên (bỏ trống để thoát):*
- Lưu dữ liệu nhập vào một DataFrame và ghi DataFrame vào file **students.xlsx**.
- Tính tuổi của sinh viên trong danh sách và ghi vào cột mới trong DataFrame có tên **Age**.
- Hiển thị dữ liệu trong DataFrame.

Thực hành 4

Bài tập 4.2

File *"200heightsweights.csv"* chứa dữ liệu về số đo chiều cao, cân nặng của các 200 người. Hãy thực hiện các thao tác sau đây:

- Đọc file vào một DataFrame.
- Xem thông tin của DataFrame.
- Xác định xem dữ liệu missing hay không? Nếu có, thay thế bằng giá trị trung bình ở cột tương ứng.
- Tạo ra cột mới, có tên **BMI**, tính chỉ số BMI theo công thức $BMI = weight(kg)/height(m)^2$.
- Lọc ra những người gầy, tức là có chỉ số BMI < 20.