HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Lập trình Python

Giảng viên: TS. Nguyễn Trọng Khánh

Điện thoại/E-mail: khanhnt@ptit.edu.vn

Bộ môn: CNPM- Khoa CNTT1

Học kỳ/Năm biên soạn: 2021-2022

Python – Phân tích dữ liệu

Một số thao tác phổ biến

- Đọc dữ liệu
- Chọn và lọc dữ liệu
- Tùy biến, sắp xếp, nhóm, dữ liệu
- Vẽ đồ thị
- Thống kê

Một số thư viện phổ biến

Thư viện và công cụ phổ biến xử lý dữ liệu

- NumPy
- SciPy
- Pandas
- SciKit-Learn

Thư viện hình ảnh minh họa

- matplotlib
- Seaborn

và nhiêu thư viện khác …

Một số thư viện phổ biến

NumPy:

- các đối tượng cho mảng và ma trận nhiều chiều, cũng như các hàm cho phép dễ dàng thực hiện các phép toán thống kê và toán học nâng cao
- cung cấp vectơ hóa các phép toán trên mảng và ma trận → cải thiện đáng kể hiệu suất
- nhiều thư viện python khác được xây dựng trên NumPy

Link: http://www.numpy.org/

Một số thư viện phổ biến

SciPy:

- tập các thuật toán cho đại số tuyến tính, phương trình vi phân, tích phân số, tối ưu hóa, thống kê …
- một phần của SciPy Stack
- được xây dựng trên NumPy

Link: https://www.scipy.org/scipylib/

Một số thư viện phổ biến

Pandas:

- hỗ trợ cấu trúc dữ liệu và công cụ làm việc với dữ liệu dạng bảng
- cung cấp các công cụ để thao tác dữ liệu: định hình lại, hợp nhất, sắp xếp, cắt lát, tổng hợp, v.v.
- cho phép xử lý dữ liệu bị thiếu

Link: http://pandas.pydata.org/

Một số thư viện phổ biến

SciKit-Learn:

- cung cấp các thuật toán học máy: phân loại, hồi quy, phân cụm, xác thực mô hình, v.v.

- được xây dựng trên NumPy, SciPy và matplotlib

Link: http://scikit-learn.org/

Một số thư viện phổ biến

matplotlib:

- Thư viện đồ thị python 2D cho phép tạo các hình ảnh minh họa số liệu ở nhiều định dạng
- biểu đồ đường thẳng, biểu đồ phân tán, biểu đồ thanh, biểu đồ, biểu đồ hình tròn, v.v.
- mức độ hỗ trợ tương đối thấp; cần. Thực hiện thêm một số thao tác để tạo ra hình ảnh hóa nâng cao

Link: https://matplotlib.org/

Một số thư viện phổ biến

Seaborn:

- dựa trên matplotlib
- cung cấp giao diện cao cấp hơn để vẽ đồ họa thống kê đẹp hơn

Link: https://seaborn.pydata.org/

Ví dụ

- Dữ liệu về lương
  - Đọc xử lý
  - Sắp xếp
  - Hiển thị

rank,discipline,phd,service,sex,salary
Prof,B,56,49,Male,186960
Prof,A,12,6,Male,93000
Prof,A,23,20,Male,110515
Prof,A,40,31,Male,131205
Prof,B,20,18,Male,104800
Prof,A,20,20,Male,122400
AssocProf,A,20,17,Male,81285
Prof,A,18,18,Male,126300
Prof,A,29,19,Male,94350
Prof,A,51,51,Male,57800
Prof,B,39,33,Male,128250
Prof,B,23,23,Male,134778
AsstProf,B,1,0,Male,88000
Prof,B,35,33,Male,162200
Prof,B,25,19,Male,153750
Prof,B,17,3,Male,150480
AsstProf,B,8,3,Male,75044
…..

Nhập thư viện

In [ ]:
```
#Import Python Libraries
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib as mpl
import seaborn as sns
```

Đọc dữ liệu sử dung pandas

```
In [ ]: #Read csv file
        df = pd.read_csv("Salaries.csv")
```

pandas cho phép đọc nhiều định dạng tệp khác nhau :

```
pd.read_excel('myfile.xlsx',sheet_name='Sheet1', index_col=None,
na_values=['NA'])
pd.read_stata('myfile.dta')
pd.read_sas('myfile.sas7bdat')
pd.read_hdf('myfile.h5','df')
```

Khám phá khung dữ liệu

In [3]:
```
#List first 5 records
df.head()
```

Out[3]:

	rank	discipline	phd	service	sex	salary
0	Prof	B	56	49	Male	186960
1	Prof	A	12	6	Male	93000
2	Prof	A	23	20	Male	110515
3	Prof	A	40	31	Male	131205
4	Prof	B	20	18	Male	104800

Luyện tập

- ✓ Đọc 10, 20, 50 dòng đầu tiên
- ✓ Đọc ngược từ dưới lên ?

Kiểu dữ liệu Data Frame

Pandas Type	Native Python Type	Description
object	string	Kiểu dữ liệu chung. Sẽ gán cho cột, nếu cột chứa lẫn lộn các loại dữ liệu (Số và xâu).
int64	int	Kiểu số. 64 đề cập đến bộ nhớ được cấp phát.
float64	float	Các ký tự số với số thập phân. Nếu một cột chứa số và NaN (xem bên dưới), pandas sẽ mặc định là float64
datetime64, timedelta[ns]	N/A	Lưu giữ dữ liệu thời gian. Sử dụng cho các thử nghiệm chuỗi thời gian.

Kiểu dữ liệu Data Frame

```
In [4]: #Check a particular column type
        df['salary'].dtype
```

```
Out[4]: dtype('int64')
```

```
In [5]: #Check types for all the columns
        df.dtypes
```

```
Out[4]: rank          object
        discipline    object
        phd           int64
        service       int64
        sex           object
        salary        int64
        dtype: object
```

Các thuộc tính của Data Frames

Data Frames là đối tượng → có thuộc tính và phương thức

df.attribute	description
dtypes	liệt kê kiểu dữ liệu của cột
columns	liệt kê tên các cột
axes	liệt kê các nhãn dòng và tên cột
ndim	số chiều
size	số phần tử
shape	trả về tutple thể hiện kích thước
values	biểu diễn numpy của dữ liệu

Luyện tập

- ✓ Tìm xem khung dữ liệu này có bao nhiêu bản ghi;
- ✓ Có bao nhiêu phần tử?
- ✓ Tên cột là gì?
- ✓ Có những loại cột nào trong khung dữ liệu này?

Các phương thức Data Frames

Tất cả thuộc tích và phương thức có thể được liệt kê qua lệnh : `dir(df)`

df.method()	Description
head([n]), tail([n])	n dòng đầu tiên hoặc cuối cùng
describe()	tạo thống kê mô tả (chỉ dành cho cột dạng số)
max(), min()	trả về giá trị lớn/nhỏ nhất cho tất cả các cột dạng số
mean(), median()	trả về giá trị trung bình/giá trị nằm giữa cho tất cả các cột dạng số
std()	độ lệch chuẩn
sample([n])	trả về một mẫu ngẫu nhiên của khung dữ liệu
dropna()	bỏ tất cả các bản ghi có giá trị bị thiếu

Luyện tập

- ✓ Tính tổng các cột dạng số ?
- ✓ Tính độ lệch chuẩn cho tất cả cột dạng số ;
- ✓ Giá trị trung bình cảu 50 bản ghi đầu tiên?

Chọn 1 cột trong DF

Cách 1: Tạo tập con của khung sử dụng tên cột

df['sex']

Cách 2: Sử dụng tên cột dưới dạng thuộc tính :

df.sex

Chú ý: tên cột trùng với tên thuộc tính, ví dụ *"rank", chỉ được sử dụng cách 1*.

Luyện tập

- ✓ Tính toán thông kê cơ bản của cột *salary (lớn nhất, nhỏ nhất, trung bình, giá trị giữa)*;
- ✓ Có bao nhiêu giá trị trong cột *salary* (sử dụng phương thức *count*);

Phương thức *groupby*

"group by" cho phép:

- Tách dữ liệu thành nhiều nhóm, dựa trên tiêu chí đầu vào
- Tính toán thống kế cho từng nhóm

In [ ]:
```
#Group data using rank
df_rank = df.groupby(['rank'])
```

In [ ]:
```
#Calculate mean value for each numeric column per each group
df_rank.mean()
```

rank	phd	service	salary
AssocProf	15.076923	11.307692	91786.230769
AsstProf	5.052632	2.210526	81362.789474
Prof	27.065217	21.413043	123624.804348

Phương thức *groupby*

Sau khi tạo đối tượng từ groupby, có thể thực thi các phép toán thông kê khác nhau

In [ ]:
```
#Calculate mean salary for each professor rank:
df.groupby('rank')[['salary']].mean()
```

	salary
rank	
AssocProf	91786.230769
AsstProf	81362.789474
Prof	123624.804348

Chú ý: sử dụng dấu ngoặc đơn '→ chuỗi dữ liệu; "→ Df

Phương thức *groupby*

groupby :

- dữ liệu gốc không bị xáo trộn
- mặc định các nhóm được sắp xếp trong quá trình thực thi. Thêm sort=False để bỏ mặc định, và tăng tốc quá trình tạo nhóm:

```
In [ ]: #Calculate mean salary for each professor rank:
        df.groupby(['rank'], sort=False)[['salary']].mean()
```

Data Frame: Lọc dữ liệu

Sử dụng các phép toán Boolean để tạo tập con → bộ lọc dữ liệu. Ví dụ, tạo tập con các dòng có lương lớn hơn $120K:

```
In [ ]: #Calculate mean salary for each professor rank:
        df_sub = df[ df['salary'] > 120000 ]
```

Bất kỳ toán tử Boolean nào đều có thể sử dụng để tạo tập con:

> greater; >= greater or equal;

< less; <= less or equal;

== equal; != not equal;

```
In [ ]: #Select only those rows that contain female professors:
        df_f = df[ df['sex'] == 'Female' ]
```

Data Frames: Cắt lát dữ liệu

Một số cách để tạo tập con Data Frame:

- 1 hoặc nhiều cột
- 1 hoặc nhiều dòng
- 1 tập của dòng và cột

Các dòng và cột có thể được chọn thông qua vị trí hoặc nhãn

Data Frames: Cắt lát dữ liệu

Khi chọn một cột, có thể sử dụng dấu ngoặc đơn, nhưng đối tượng kết quả sẽ là dữ liệu kiểu Chuỗi (không phải DataFrame):

```
In [ ]: #Select column salary:
        df['salary']
```

Khi cần chọn nhiều hơn một cột và/hoặc tạo đầu ra là DataFrame → sử dụng dấu ngoặc kép:

```
In [ ]: #Select column salary:
        df[['rank','salary']]
```

Data Frames: Chọn dòng

Nếu cần chọn khoảng dòng, có thể chỉ định phạm vi bằng cách sử dụng ":"

```
In [ ]: #Select rows by their position:
        df[10:20]
```

Lưu ý rằng dòng đầu tiên có vị trí 0 và giá trị cuối cùng trong phạm vi bị bỏ qua:
Vì vậy, đối với phạm vi 0:10, 10 hàng đầu tiên được trả về với các vị trí bắt đầu bằng 0 và kết thúc bằng 9

Data Frames: phương thức loc

Nếu cần chọn một loạt các dòng, kết hợp với nhãn, chúng ta có thể sử dụng phương thức loc:

In [ ]:
```
#Select rows by their labels:
df_sub.loc[10:20,['rank','sex','salary']]
```

Out[ ]:

	rank	sex	salary
10	Prof	Male	128250
11	Prof	Male	134778
13	Prof	Male	162200
14	Prof	Male	153750
15	Prof	Male	150480
19	Prof	Male	150500

Data Frames: phương thức iloc

Nếu cần chọn một dãy dòng và/hoặc cột, sử dụng vị trí, chúng ta có thể sử dụng phương pháp iloc:

In [ ]:
```
#Select rows by their labels:
df_sub.iloc[10:20,[0, 3, 4, 5]]
```

Out[ ]:

	rank	service	sex	salary
26	Prof	19	Male	148750
27	Prof	43	Male	155865
29	Prof	20	Male	123683
31	Prof	21	Male	155750
35	Prof	23	Male	126933
36	Prof	45	Male	146856
39	Prof	18	Female	129000
40	Prof	36	Female	137000
44	Prof	19	Female	151768
45	Prof	25	Female	140096

Data Frames: phương thức iloc

```
df.iloc[0]  # First row of a data frame
df.iloc[i]  #(i+1)th row
df.iloc[-1] # Last row
```

```
df.iloc[:, 0]  # First column
df.iloc[:, -1] # Last column
```

```
df.iloc[0:7]         #First 7 rows
df.iloc[:, 0:2]      #First 2 columns
df.iloc[1:3, 0:2]    #Second through third rows and first 2 columns
df.iloc[[0,5], [1,3]]  #1st and 6th rows and 2nd and 4th columns
```

Data Frames: Sắp xếp

Có thể sắp xếp dữ liệu theo một giá trị trong cột. Mặc định, việc sắp xếp sẽ diễn ra theo thứ tự tăng dần và một khung dữ liệu mới được trả về.

In [ ]:
```
# Create a new data frame from the original sorted by the column Salary
df_sorted = df.sort_values( by ='service')
df_sorted.head()
```

Out[

	rank	discipline	phd	service	sex	salary
55	AsstProf	A	2	0	Female	72500
23	AsstProf	A	2	0	Male	85000
43	AsstProf	B	5	0	Female	77000
17	AsstProf	B	4	0	Male	92000
12	AsstProf	B	1	0	Male	88000

Data Frames: Sắp xếp

Sắp xếp dữ liệu với 2 cột:

In [ ]:
```
df_sorted = df.sort_values( by =['service', 'salary'], ascending = [True, False])
df_sorted.head(10)
```

Out[

	rank	discipline	phd	service	sex	salary
52	Prof	A	12	0	Female	105000
17	AsstProf	B	4	0	Male	92000
12	AsstProf	B	1	0	Male	88000
23	AsstProf	A	2	0	Male	85000
43	AsstProf	B	5	0	Female	77000
55	AsstProf	A	2	0	Female	72500
57	AsstProf	A	3	1	Female	72500
28	AsstProf	B	7	2	Male	91300
42	AsstProf	B	4	2	Female	80225
68	AsstProf	A	4	2	Female	77500

Thiếu giá trị

Giá trị bị thiêu được đánh dấu NaN

In [ ]:
```
# Read a dataset with missing values
flights = pd.read_csv("flights.csv")
```

In [ ]:
```
# Select the rows that have at least one missing value
flights[flights.isnull().any(axis=1)].head()
```

Out[

	year	month	day	dep_time	dep_delay	arr_time	arr_delay	carrier	tailnum	flight	origin	dest	air_time	distance	hour	minute
330	2013	1	1	1807.0	29.0	2251.0	NaN	UA	N31412	1228	EWR	SAN	NaN	2425	18.0	7.0
403	2013	1	1	NaN	NaN	NaN	NaN	AA	N3EHAA	791	LGA	DFW	NaN	1389	NaN	NaN
404	2013	1	1	NaN	NaN	NaN	NaN	AA	N3EVAA	1925	LGA	MIA	NaN	1096	NaN	NaN
855	2013	1	2	2145.0	16.0	NaN	NaN	UA	N12221	1299	EWR	RSW	NaN	1068	21.0	45.0
858	2013	1	2	NaN	NaN	NaN	NaN	AA	NaN	133	JFK	LAX	NaN	2475	NaN	NaN

Thiếu giá trị

Một số phương thức xử lý dữ liệu bị thiếu:

df.method()	Thông tin
dropna()	Bỏ dữ liệu bị thiếu
dropna(how='all')	Bỏ dữ liệu trong đó tất cả các ô đều là NA
dropna(axis=1, how='all')	Bỏ cột nếu thiếu tất cả các giá trị
dropna(thresh = 5)	Bỏ các hàng chứa ít hơn 5 giá trị không bị thiếu
fillna(0)	Thay thế các giá trị bị thiếu bằng 0
isnull()	trả về True nếu thiếu giá trị
notnull()	Trả về True cho các giá trị không bị thiếu

Thiếu giá trị

- Khi tổng hợp dữ liệu, các giá trị bị thiếu sẽ được coi là 0
- Nếu thiếu tất cả các giá trị, tổng sẽ bằng NaN
- Các phương thức cumsum() và cumprod() bỏ qua các giá trị bị thiếu nhưng bảo toàn chúng trong các mảng kết quả
- Các giá trị bị thiếu trong phương thức GroupBy bị loại trừ
- Nhiều phương pháp thống kê mô tả có tùy chọn bỏ qua để kiểm soát nếu dữ liệu bị thiếu nên được loại trừ. Giá trị này được đặt thành True theo mặc định

Các chức năng tổng hợp trong Pandas

Phép tính tổng hợp – Tính toán toán thống kê trên các nhóm, i.e.

- tính tổng hoặc trung bình của nhóm
- tính kích cỡ, số lượng nhóm

Các hàm tập hợp phổ biến

min, max
count, sum, prod
mean, median, mode, mad
std, var

Các chức năng tổng hợp trong Pandas

Phương thức agg () hữu ích khi cần nhiều thống kê trên mỗi cột:

In [ ]:
```
flights[['dep_delay','arr_delay']].agg(['min','mean','max'])
```

Out[ ]:

	dep_delay	arr_delay
min	-16.000000	-62.000000
mean	9.384302	2.298675
max	351.000000	389.000000

Thống kê mô tả cơ bản

df.method()	Thông tin
describe	Thông kê cơ bản (count, mean, std, min, quantiles, max)
min, max	Giá trị lớn nhất, nhỏ nhất
mean, median, mode	Trung bình, giá trị giữa, và mode
var, std	Phương sai và độ lệch chuẩn
sem	Sai số chuẩn của giá trị trung bình
skew	Độ lệch mẫu
kurt	kurtosis

Đồ thị

Gói Seaborn được xây dựng trên matplotlib nhưng cung cấp giao diện cấp cao để vẽ đồ họa thống kê đẹp hơn → trực quan hóa dữ liệu thống kê

	Thông tin
distplot	biểu đồ histogram
barplot	ước tính xu hướng trung tâm cho một biến số
violinplot	Biểu đồ thể hiện mật độ xác suất của dữ liệu
jointplot	Biểu đồ các điểm
regplot	Biểu đồ hồi quy
pairplot	Biểu đồ ghép nối
boxplot	Biểu đố hình hộp
swarmplot	Biểu đồ phân tán phân loại
factorplot	Biểu đồ phân loại chung

Phân tích thông kê cơ bản

statsmodel và scikit-learn - đều có một số chức năng để phân tích thống kê

statsmodels:

- Hồi quy tuyến tính
- Kiểm tra ANOVA
- Kiểm tra giải thuyết ...

scikit-learn: Học máy

- kmeans
- SVM
- random forests
- ...