

<div> <div>Today for Tomorrow</div> <div> <div>A</div> <div>D</div> </div> <div> <div>A</div> <div>I</div> </div> </div>	<b>PHÒNG THÍ NGHIỆM THỰC HÀNH ỨNG DỤNG</b> <b>KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO</b> Applied Data Science and Artificial Intelligence Lab
	 <div> <b>LẬP TRÌNH PYTHON CHO KHOA HỌC DỮ LIỆU</b>  <b>PYTHON FOR DATA SCIENCE</b> </div> <div>Hà Nội ★ 11.2020</div>
Nothing is faster than Innovation together	

1

<div> <div>Today for Tomorrow</div> <div> <div>A</div> <div>D</div> </div> <div> <div>A</div> <div>I</div> </div> </div>	<b>Thông tin giảng viên</b>
	<div>  <div> <b>TS. TẠ QUANG CHIẾU</b> </div> </div> <div> <b>Vị trí &amp; Đào tạo</b> <ul style="list-style-type: none"> <li>- Trưởng bộ môn Hệ thống thông tin và Tri thức, Khoa Công nghệ Thông tin</li> <li>- PhD &amp; Post-doc in Polytech Tours, France</li> </ul> <b>Lĩnh vực giảng dạy, nghiên cứu</b> <ul style="list-style-type: none"> <li>- Big data, Data science, machine learning, AI</li> <li>- Các phương pháp cho bài toán lập kế hoạch</li> </ul> <b>Contact:</b> <p>(E): <a href="mailto:quangchieu.ta@gmail.com">quangchieu.ta@gmail.com</a></p> <p>(M): 0913 522 275</p> </div>
Dr. Tạ Quang Chiếu – (E): <a href="mailto:quangchieu.ta@gmail.com">quangchieu.ta@gmail.com</a> – (M): 0913 522 275	

2

A A	D I	Objectives	
		<ul style="list-style-type: none"> <li>• Hiểu được tầm quan trọng của Khoa học dữ liệu</li> <li>• Vận dụng được các bước trong quy trình thực hiện một dự án về khoa học dữ liệu.</li> <li>• Kiến thức, kỹ năng cần thiết để trở thành một nhà khoa học dữ liệu</li> <li>• Áp dụng được Python và các thư viện phổ biến trong giải quyết một số bài toán cơ bản của Khoa học dữ liệu.</li> </ul>	
		Dr. Tạ Quang Chiếu – [E]: <a href="mailto:quangchieu.ta@gmail.com">quangchieu.ta@gmail.com</a> – [M]: 0913 522 275	3/62

3

A A	D I	References	
		<ul style="list-style-type: none"> <li>[1]. Introducing Data Science</li> <li>[2]. Python Data Science Handbook</li> <li>[3]. Data Science from Scratch</li> <li>[4]. Python crash course</li> <li>[5]. Bài giảng Khoa học dữ liệu</li> </ul>	
		Dr. Tạ Quang Chiếu – [E]: <a href="mailto:quangchieu.ta@gmail.com">quangchieu.ta@gmail.com</a> – [M]: 0913 522 275	4/62

4

A A	D I	<b>PHẦN 2: NHẬP MÔN KHOA HỌC DỮ LIỆU</b>	
		FUNDAMENTALS OF DATA SCIENCE	
<b>CONTENT</b>			
<b>1</b>		CHAPTER 1: INTRODUCTION	
<b>2</b>		<b>CHAPTER 2: PYTHON LIBRARIES FOR DATA SCIENCE</b>	
Dr. Tà Quang Chiếu - [E]: <a href="mailto:quangchieu.ta@gmail.com">quangchieu.ta@gmail.com</a> - [M]: 0913 522 275			<b>5/62</b>

5

A A	D I	<b>Content of chapter 2</b>	
2.1 Python Library for Data Science			
2.2 Numpy library			
<b>2.3 Pandas library</b>			
2.4 Matplotlib library			
2.5 Scikit-learn library			
Dr. Tà Quang Chiếu - [E]: <a href="mailto:quangchieu.ta@gmail.com">quangchieu.ta@gmail.com</a> - [M]: 0913 522 275			<b>6/62</b>

6

A	D	<div style="display: flex; justify-content: space-between; padding: 5px;"> <span>Navigation icons</span> <span>7/62</span> </div>
A	I	

# Phân tích và xử lý dữ liệu với Pandas (01)


Dr. Tà Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

7

A	D	<div style="display: flex; justify-content: space-between; padding: 5px;"> <span>Navigation icons</span> <span>8/62</span> </div>
A	I	

## Nội dung

1. Giới thiệu
2. Tạo đối tượng cơ bản trong Pandas
  - Series
  - Dataframe
3. Quan sát và truy xuất dữ liệu trong DataFrame
4. Replacing Values, Rename Columns
5. Lọc dữ liệu trong DataFrame
6. Xác định các tham số thống kê: Sum, Cumsum, Min, Max, Mean, Median, Std
7. Giá trị duy nhất (Unique)
8. Time series data



Pandas

Dr. Tà Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

8

A
D
A
I

# 1. Giới thiệu

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

9/62

9

A
D
A
I

## 1. Giới thiệu

**Pandas** là một thư viện mã nguồn mở được xây dựng dựa trên NumPy, sử dụng để thao tác và phân tích dữ liệu. Với Pandas chúng ta có thể:

- Xử lý tập dữ liệu khác nhau về định dạng: chuỗi thời gian, bảng không đồng nhất, ma trận dữ liệu
- Import dữ liệu từ nhiều nguồn khác nhau như CSV, DB/SQL...
- Xử lý vô số phép toán cho tập dữ liệu: subsetting, slicing, filtering, merging, groupBy, re-ordering, and re-shaping,...
- Xử lý dữ liệu mất mát theo mong muốn.
- Xử lý, phân tích dữ liệu tốt như mô hình hoá và thống kê.
- Tích hợp tốt với các thư viện khác của python. <https://pandas.pydata.org/>




Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

10/62

10

A
D

A
I

## 1. Giới thiệu

Pandas làm việc thông qua 3 đối tượng Series, **DataFrame**, Panel

**SERIES**

7	2	9	10
---	---	---	----

axis 0 →

**DATA FRAME**

5.2	3.0	4.5
9.1	0.1	0.3

axis 0 ↓ axis 1 →

**PANEL**

axis 0 ↓ axis 1 → axis 2 →

```

1 kiểm tra phiên bản của thư viện Pandas
2 import pandas as pd
3 print('Version Pandas: ',pd.__version__)

```

Version Pandas: 1.1.1

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

11/62

11

A
D

A
I

## 2. Series, DataFrame trong Pandas

## 2. Series, DataFrame trong Pandas

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

12/62

12

## A D A I 2.1 Series

- Series** là mảng một chiều (1D) giống như kiểu vector trong Numpy, hay như một cột của một bảng, nhưng nó bao gồm thêm một bảng đánh index.

Index	Data
0	2.80
1	3.00
2	4.44
3	5.00

Series

dtype: float64

Row-1 →

Row-2 →

...

Row-n →

idx

↓

Index

Pandas Data structure

Column-1	Column-2	...	Column-n
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...

DataFrame

Series

7

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

13/62

13

## A D A I 2.1 Series

- Tạo Series sử dụng phương thức;**
  - pd.Series(data, index, dtype, name)**

```

1 #Tạo một đối tượng series
2 #index mặc định đánh số từ 0
3 data = pd.Series([2.8, 3, 4.44, 5])
4 data

```

```

0    2.80
1    3.00
2    4.44
3    5.00
dtype: float64

```

```

1 #Mỗi một đối tượng series bao gồm 2 thành phần
2 #1. values
3 #2. index
4
5 print('Values:', data.values)
6 print('Indices:', data.index)

```

```

Values: [2.8 3.  4.44 5. ]
Indices: RangeIndex(start=0, stop=4, step=1)

```

```

1 #Tạo một đối tượng series với index thiết lập
2 data = pd.Series([1.25, 2, 3.5, 4.75, 8.0],
3                  index=['a', 'b', 'c', 'd', 'k'])
4 data

```

```

a    1.25
b    2.00
c    3.50
d    4.75
k    8.00
dtype: float64

```

```

1 print('Values:', data.values)
2 print('Indices:', data.index)

```

```

Values: [0.25 0.5  0.75 1. ]
Indices: Index(['a', 'b', 'c', 'd'], dtype='object')

```

<https://pandas.pydata.org/docs/reference/api/pandas.Series.html>

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

14/62

14

## A D 2.2 DataFrame

**DataFrame:** Cấu trúc dạng bảng 2D, kích thước có thể thay đổi được. Dữ liệu một cột là đồng nhất nhưng có thể không đồng nhất giữa các cột

Pandas Data structure

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

15/62

15

## A D 2.2 DataFrame

- Tạo DataFrame sử dụng phương thức;**
  - `pd.DataFrame(data, index, columns, dtype)`

```

1 #Tạo một DataFrame từ một biến Dict
2 #Chỉ số được tạo mặc định từ 0
3 data_dict = {
4     'apples': [3, 2, 0, 1],
5     'oranges': [0, 3, 7, 2]}
6
7 purchases = pd.DataFrame(data_dict)
8 purchases

```

	apples	oranges
0	3	0
1	2	3
2	0	7
3	1	2

```

1 #Tạo DataFrame với index thiết lập
2 purchases = pd.DataFrame(data_dict,
3                           index=['June', 'Robert', 'Lily', 'David'])
4 purchases

```

	apples	oranges
June	3	0
Robert	2	3
Lily	0	7
David	1	2

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

16/62

16



A D
A I

## 3. Quan sát và truy cập dữ liệu trong DataFrame

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

17/62

17

A D
A I

## 3.1 Quan sát dữ liệu

- Đọc file dữ liệu mẫu: `csv_Data_loan` (Bài 06)
- Đây là file dữ liệu cho biết thông tin về các khoản vay cho các mục đích khác nhau của người dùng Mỹ.

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

18/62

18

AD

AI

## 3.1 Quan sát dữ liệu

- df.info()** : Hiển thị thông tin chi tiết biến DataFrame
- df.head(n)**: Hiển thị n dòng đầu tiên của biến df (default = 5)
- df.tail(n)** : Hiển thị n dòng cuối cùng biến df (default = 5)
- df.shape** : Hiển thị kích thước (rows x columns) của biến df
- df.columns**: Tên các cột trong biến df
- df.isnull()** : Kiểm tra dữ liệu rỗng trong biến df
- df.isnull().sum()** : Tính tổng các dòng dữ liệu null trong df
- df.count()** : Tổng số dòng dữ liệu không null trong df
- df.size** : Số phần tử của biến df (=rows x columns)
- df.dtypes** : Kiểu dữ liệu của từng columns trong df

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

19/62

19

AD

AI

## 3.1 Quan sát dữ liệu

- df.describe()** : Một số đặc trưng thống kê của biến df
  - Tham số include='O': thống kê các cột có kiểu dữ liệu Object
  - Tham số include='all': Thống kê tất cả các cột trong df

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

20/62

20

### 3.2 Truy cập dữ liệu trong DataFrame

- `df[['Col1', 'Col2', 'Col3']]`: Chỉ truy cập dữ liệu của các cột có tên **Col1, Col2, Col3** trong dataframe `df`

```

1 #Truy xuất dữ liệu theo cột
2 #Lấy dữ liệu của một cột
3 df_state = df_loan[['addr_state']]
4 df_state.head()

```

	addr_state
0	AZ
1	GA
2	IL
3	CA
4	AZ

```

1 #Truy xuất dữ liệu theo cột
2 #Chỉ lấy dữ liệu của 3 cột: loan_amnt, int_rate, purpose
3 df_loan1 = df_loan[['loan_amnt','int_rate','purpose']]
4 df_loan1.head()

```

	loan_amnt	int_rate	purpose
0	5000	10.65	credit_card
1	2500	15.27	car
2	2400	15.96	small_business
3	10000	13.49	other
4	5000	7.90	wedding

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

21/62

21

### 3.2 Truy cập dữ liệu trong DataFrame

- `df.iloc[[index_row],[index_col]]`: Truy cập tới dữ liệu của hàng và cột qua **chỉ số index\_row, index\_col (tương tự như với Numpy)**

```

1 #Sử dụng .iloc truy xuất dữ liệu như với Numpy
2 #Truy xuất 10 dòng dữ liệu từ [10 --> 20) tất cả các cột
3 df_loan.iloc[10:20,:]

```

	loan_amnt	term	int_rate	emp_length	home_ownership	annual_inc	purpose	addr_state	dti	delinq_2yrs	revol_util	total_acc	bad_loan
10	9000	36 months	13.49	0.0	RENT	30000.00	debt_consolidation	VA	10.08	0.0	91.70	9.0	1
11	3000	36 months	9.91	3.0	RENT	15000.00	credit_card	IL	12.56	0.0	43.10	11.0	0
12	10000	36 months	10.65	3.0	RENT	100000.00	other	CA	7.06	0.0	55.50	29.0	1
13	1000	36 months	16.29	0.0	RENT	28000.00	debt_consolidation	MO	20.31	0.0	81.50	23.0	0
14	10000	36 months	15.27	4.0	RENT	42000.00	home_improvement	CA	18.60	0.0	70.20	28.0	0
15	3600	36 months	6.03	10.0	MORTGAGE	110000.00	major_purchase	CT	10.52	0.0	16.00	42.0	0

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

22/62

22

A D  
A I

## 3.2 Truy cập dữ liệu trong DataFrame

- **df.loc[[name\_index],[name\_col]]**: Truy cập tới dữ liệu của hàng và cột qua **name\_index**, **tên cột name\_col**

```

1 #Truy cập từ dòng 20 đến dòng 25 của df
2 #chỉ lấy dữ liệu 4 cột: loan_amnt, home_ownership, purpose, addr_state
3 df_loan.loc[20:25,['loan_amnt','home_ownership','purpose','addr_state']]

```

	loan_amnt	home_ownership	purpose	addr_state
20	10000	RENT	debt_consolidation	CA
21	6000	RENT	major_purchase	CA
22	15000	MORTGAGE	credit_card	IL
23	15000	RENT	debt_consolidation	NY
24	5000	RENT	other	PA
25	4000	MORTGAGE	debt_consolidation	FL

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275
23/62

23

A D  
A I

## 3.2 Truy cập dữ liệu trong DataFrame

Type	Notes
df[val]	Select single column or sequence of columns from the DataFrame; special case conveniences: boolean array (filter rows), slice (slice rows), or boolean DataFrame (set values based on some criterion)
df.loc[val]	Selects single row or subset of rows from the DataFrame by label
df.loc[:, val]	Selects single column or subset of columns by label
df.loc[val1, val2]	Select both rows and columns by label
df.iloc[where]	Selects single row or subset of rows from the DataFrame by integer position
df.iloc[:, where]	Selects single column or subset of columns by integer position
df.iloc[where_i, where_j]	Select both rows and columns by integer position
df.at[label_i, label_j]	Select a single scalar value by row and column label
df.iat[i, j]	Select a single scalar value by row and column position (integers)
reindex method	Select either rows or columns by labels
get_value, set_value methods	Select single value by row and column label

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275
24/62

24

A D A I

## 4. Replacing Values, Rename columns

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

25/62

25

A D A I

### 4.1 Replacing Values

- Thay thế 1 giá trị trong Dataframe, thực hiện tương tự như với Numpy. Sử dụng `.loc`; `.iloc` để xác định phần tử cần cập nhật, thay đổi giá trị

	loan_amnt	home_ownership	purpose	addr_state
0	5000	RENT	credit_card	AZ
1	2500	RENT	car	GA
2	2400	RENT	small_business	IL
3	10000	RENT	other	CA
4	5000	RENT	wedding	AZ
5	3000	RENT	car	CA
6	5600	OWN	small_business	CA
7	5375	RENT	other	TX
8	6500	OWN	debt_consolidation	AZ
9	12000	OWN	debt_consolidation	CA
10	9000	RENT	debt_consolidation	VA

```

1 #Thay thế giá trị purpose: credit_card--> wedding
2 #của index đầu tiên
3 df_new.loc[0,'purpose'] = 'wedding'
4 df_new

```

	loan_amnt	home_ownership	purpose	addr_state
0	5000	RENT	wedding	AZ
1	2500	RENT	car	GA
2	2400	RENT	small_business	IL
3	10000	RENT	other	CA

```

1 #Thay thế giá trị thuộc tính loan_amnt: 2400 --> 8800
2 #của index = 2
3 df_new.iloc[2,0] = 8800
4 df_new

```

	loan_amnt	home_ownership	purpose	addr_state
0	5000	RENT	wedding	AZ
1	2500	RENT	car	GA
2	8800	RENT	small_business	IL
3	10000	RENT	other	CA

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

26/62

26

## A D A I 4.1 Replacing Values

- df.replace():** Thay thế các giá trị trong toàn bộ DataFrame. (tham số inplace=True|False áp dụng thay đổi cho dataframe hiện tại hay không?).

	loan_amnt	home_ownership	purpose	addr_state
0	5000	RENT	credit_card	AZ
1	2500	RENT	car	GA
2	2400	RENT	small_business	IL
3	10000	RENT	other	CA
4	5000	RENT	wedding	AZ
5	3000	RENT	car	CA
6	5600	OWN	small_business	CA
7	5375	RENT	other	TX
8	6500	OWN	debt_consolidation	AZ
9	12000	OWN	debt_consolidation	CA
10	9000	RENT	debt_consolidation	VA

```

1 #Thay thế nhiều giá trị trong DataFrame
2 #RENT --> MORTGAGE
3 #car --> small_business
4 df_new.replace({'RENT': 'MORTGAGE',
5                'car': 'small_business'})

```

```

1 #khi muốn thay đổi áp dụng lên DataFrame hiện tại
2 #thiết lập tham số inplace=True
3 df_new.replace({'RENT': 'MORTGAGE',
4                'car': 'small_business'}, inplace=True)
5 df_new

```

	loan_amnt	home_ownership	purpose	addr_state
0	5000	MORTGAGE	wedding	AZ
1	2500	MORTGAGE	small_business	GA
2	8800	MORTGAGE	small_business	IL
3	10000	MORTGAGE	other	CA
4	5000	MORTGAGE	wedding	AZ
5	3000	MORTGAGE	small_business	CA
6	5600	OWN	small_business	CA
7	5375	MORTGAGE	other	TX
8	6500	OWN	debt_consolidation	AZ
9	12000	OWN	debt_consolidation	CA
10	9000	MORTGAGE	debt_consolidation	VA

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

27/62

27

## A D A I 4.1 Replacing Values

- df.replace():** Thay thế các giá trị theo từng cột (tham số inplace=True|False áp dụng thay đổi cho dataframe hiện tại hay không?).

	loan_amnt	home_ownership	purpose	addr_state
0	5000	RENT	credit_card	AZ
1	2500	RENT	car	GA
2	2400	RENT	small_business	IL
3	10000	RENT	other	CA
4	5000	RENT	wedding	AZ
5	3000	RENT	car	CA
6	5600	OWN	small_business	CA
7	5375	RENT	other	TX
8	6500	OWN	debt_consolidation	AZ
9	12000	OWN	debt_consolidation	CA
10	9000	RENT	debt_consolidation	VA

```

1 #Thay thế tên viết tắt bằng tên đầy đủ.
2 state_name={'AZ': 'Arizona',
3            'GA': 'Georgia',
4            'IL': 'Illinois',
5            'CA': 'California',
6            'TX': 'Texas',
7            'VA': 'Virginia'}
8 #Trong cột addr_state
9 df_new['addr_state'].replace(state_name, inplace=True)
10 df_new

```

	loan_amnt	home_ownership	purpose	addr_state
0	5000	RENT	credit_card	Arizona
1	2500	RENT	car	Georgia
2	2400	RENT	small_business	Illinois
3	10000	RENT	other	California
4	5000	RENT	wedding	Arizona
5	3000	RENT	car	California

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

28/62

28

A D
A I

## 4.2 Rename Columns

- **df.rename():** thay đổi tên cột trong DataFrame

```

1 #Muốn áp dụng thay đổi vào trực tiếp biến df, sử dụng inplace=True
2 df_new.rename(columns={'loan_amnt':'Số tiền vay',
3                        'home_ownership':'Tình trạng nhà ở',
4                        'purpose':'Mục đích vay tiền',
5                        'addr_state':'Địa chỉ'}, inplace=True)
6 df_new.head()

```

01

	Số tiền vay	Tình trạng nhà ở	Mục đích vay tiền	Địa chỉ
0	5000	RENT	credit_card	AZ
1	2500	RENT	car	GA
2	2400	RENT	small_business	IL
3	10000	RENT	other	CA
4	5000	RENT	wedding	AZ

```

1 #Đổi tên cột sang viết hoa
2 df_new.rename(str.upper, axis='columns')

```

02

	SỐ TIỀN VAY	TÌNH TRẠNG NHÀ Ở	MỤC ĐÍCH VAY TIỀN	ĐỊA CHỈ
0	5000	RENT	credit_card	AZ
1	2500	RENT	car	GA
2	2400	RENT	small_business	IL
3	10000	RENT	other	CA
4	5000	RENT	wedding	AZ

Dr. Tạ Quang Chiếu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
29/62

29

A D
A I

# Thực hành 1

Dr. Tạ Quang Chiếu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
30/62

30

## Thực hành 1

**Mô tả file dữ liệu: Data\_Patient.csv**

- File dữ liệu chứa thông tin của 300 bệnh nhân bị bệnh tim mạch
- Mỗi dòng ứng với thông tin của một bệnh nhân, bao gồm 9 thuộc tính

	A	B	C	D	E	F	G	H	I
	id	feature_1	feature_2	feature_3	feature_4	feature_5	feature_6	feature_7	feature_8
1	Patient_01	63	Male	Typical angina	145	233	150	6	0
2	Patient_02	67	Male	Asymptomatic	160	286	108	3	1
3	Patient_03	67	Male	Asymptomatic	120	229	129	7	1
4	Patient_04	37	Male	Non-anginal pain	130	250	187	3	0
5	Patient_05	41	Female	Atypical angina	130	204	172		0
6	Patient_06	56	Male	Atypical angina	120	236	178	3	0
7	Patient_07	62	Female	Asymptomatic	140	268	160	3	1
8	Patient_08	57	Female	Asymptomatic	120	354	163	3	0
9	Patient_09	63	Male	Asymptomatic	130	254	147	7	1
10	Patient_10	53	Male	Asymptomatic	140	203	155	7	1
11	Patient_11	57	Male	Asymptomatic	140	192	148	6	0
12	Patient_12	56	Female	Atypical angina	140	294	153	3	0
13	Patient_13	56	Male	Non-anginal pain	130	256	142	6	1
14	Patient_14	44	Male	Atypical angina	120	263	173	7	0
15	Patient_15	52	Male	Non-anginal pain	172	199	162	7	0
16	Patient_16	57	Male	Non-anginal pain	150	168	174	3	0
17	Patient_17	48	Male	Atypical angina	110	229	168	7	1
18	Patient_18	54	Male	Asymptomatic	140	239	160	3	0
19	Patient_19	48	Female	Non-anginal pain	130	275	139	3	0
20	Patient_20	49	Male	Atypical angina	130	266	171	3	0

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

31/62

31

## Thực hành 1

**Chi tiết thông tin của một bệnh nhân như sau:**

- id:** Mã của bệnh nhân (số)
- Feature\_1:** Tuổi của bệnh nhân (số)
- Feature\_2:** Giới tính của bệnh nhân (chuỗi: Male – Female)
- Feature\_3:** Cho biết loại triệu chứng đau ngực mà bệnh nhân này mắc phải, với 4 giá trị: (Typical angina, Atypical angina, Non-anginal pain, Asymptomatic)
- Feature\_4:** Huyết áp của bệnh nhân – đơn vị: mmhg (số)
- Feature\_5:** Chỉ số cholesterol của bệnh nhân – đơn vị: mg/dl (số)
- Feature\_6:** Thông số nhịp tim của bệnh nhân – đơn vị: lần/phút (số)
- Feature\_7:** Chỉ số Thalassemia của bệnh nhân chỉ gồm 3 giá trị (3: Bình thường | 4: Kiểm khuyết cố định | 7: Kiểm khuyết có thể đảo ngược)
- Feature\_8:** Cho biết bệnh nhân có bị bệnh tim hay không? (0: Không bị bệnh tim mạch | 1: Bị bệnh tim mạch)

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

32/62

32



A

D


A

I

## Thực hành 1

**Yêu cầu 1.1:**

- Đọc dữ liệu từ file `Data_Patient.csv` vào biến kiểu dataframe: `df_patient` với cột `feature_1` là cột chỉ số (`index_col`)
- Hiển thị thông tin tổng quan của tập dữ liệu
- Hiển thị thông tin của 10 bệnh nhân đầu tiên và 5 bệnh nhân cuối cùng của tập dữ liệu.
- Đặt lại tên các cột dữ liệu trong Dataframe như sau:
  - Feature\_1 → Age
  - Feature\_2 → Gender
  - Feature\_3 → Type
  - Feature\_4 → Blood\_pressure
  - Feature\_5 → Cholesterol
  - Feature\_6 → Heartbeat
  - Feature\_7 → Thalassemia
  - Feature\_8 → Result



Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

33/62

33

A

D

A

I

## Thực hành 1

**Yêu cầu 1.2:**

- Sử dụng phương thức `.describe()` cho biết:
  - Thuộc tính Age:
    - Tuổi của bệnh nhân trẻ nhất
    - Tuổi của bệnh nhân già nhất
  - Thuộc tính Cholesterol:
    - Cholesterol trung bình của các bệnh nhân
    - Độ lệch chuẩn của giá trị này trong toàn bộ tập dữ liệu
  - Bao nhiêu bệnh nhân giới tính nam (Male)
  - Có bao nhiêu giá trị khác nhau của thuộc tính Type. Giá trị xuất hiện nhiều nhất là giá trị nào, bao nhiêu lần.

	Gender	Type
count	300	295
unique	2	4
top	Male	Asymptomatic
freq	205	139

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

34/62

34

A

D

I

A

Thực hành 1

**Yêu cầu 1.3:**

- Cho biết những cột nào trong dữ liệu có chứa missing data và số lượng missing là bao nhiêu?

**Yêu cầu 1.4:**

- Hiển thị thông tin của các bệnh nhân:
  - Bệnh nhân có index: **Patient\_100; Patient\_150; Patient\_200**
  - Bệnh nhân ở vị trí 255 đến 260, với 3 thuộc tính: Age, Gender và Result

id	Age	Gender	Type	Blood_pressure	Cholesterol	Heartbeat	Thalassemia	Result
Patient_100	45	Male	Asymptomatic	115	260	185	3.0	0
Patient_150	52	Male	Typical angina	152	298	178	7.0	0
Patient_200	50	Female	Asymptomatic	110	254	159	3.0	0

id	Age	Gender	Result
Patient_255	42	Female	0
Patient_256	67	Female	0
Patient_257	76	Female	0
Patient_258	70	Male	0
Patient_259	57	Male	1
Patient_260	44	Female	0

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

35/62

35

A

D

I

A

Thực hành 1

**Yêu cầu 1.5:**

- Thay đổi giá trị cho thuộc tính Gender: **Male** → **0**, **Female** → **1**
- Thay đổi giá trị cho thuộc tính Result: **0** → **No**, **1** → **Yes**
- Cập nhật giá trị thuộc tính Thalassemia của bệnh nhân có index: **Patient\_05 bằng giá trị 4.0**

id	Age	Gender	Type	Blood_pressure	Cholesterol	Heartbeat	Thalassemia	Result
Patient_01	63	0	Typical angina	145	233	150	6.0	No
Patient_02	67	0	Asymptomatic	160	286	108	3.0	Yes
Patient_03	67	0	Asymptomatic	120	229	129	7.0	Yes
Patient_04	37	0	Non-anginal pain	130	250	187	3.0	No
Patient_05	41	1	Atypical angina	130	204	172	4.0	No

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

36/62

36

A

D

A

I

## 5. Filter Data

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

37/62

37

A

D

A

I

## 5. Filter Data

- Để lọc dữ liệu trong DataFrame có thể sử dụng nhiều cách khác nhau

```

1 #Lọc danh sách người giới tính nam
2 #Cách 1:
3 df_male1 = df_bmi[df_bmi.Gender=="Male"]
4 df_male1.head(2)

```

	Personal	Gender	Height_cm	Weight_kg
0	P1	Male	174	96
1	P2	Male	189	87

01

```

1 #Cách 2: sử dụng phương thức query
2 df_male2 = df_bmi.query('Gender=="Male"')
3 df_male2.head(2)

```

	Personal	Gender	Height_cm	Weight_kg
0	P1	Male	174	96
1	P2	Male	189	87

02

```

1 #Cách 3: sử dụng iloc
2 df_male3 = df_bmi.loc[(df_bmi.Gender=="Male")]
3 df_male3.head(2)

```

	Personal	Gender	Height_cm	Weight_kg
0	P1	Male	174	96
1	P2	Male	189	87

03

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

38/62

38

## A D A I 5. Filter Data

- Sử dụng toán tử **& (and)** - **| (or)** - **~ (not)** để kết hợp nhiều điều kiện trong khi lọc dữ liệu

```
1 #Kết hợp nhiều tiêu chí lọc dữ liệu
2 #Lọc người có giới tính Female và cân nặng dưới 70kg
3 df_p1 = df_bmi[(df_bmi.Gender == 'Female') & (df_bmi.Weight_kg < 70)]
4 df_p1
```

	Personal	Gender	Height_cm	Weight_kg	
	24	P25	Female	172	67
	25	P26	Female	151	64
	32	P33	Female	195	65
	51	P52	Female	176	54

```
1 #Kết hợp nhiều tiêu chí tìm kiếm
2 #Lọc người có chiều cao > 195 cm hoặc cân nặng > 150kg
3 df_p2 = df_bmi[(df_bmi.Height_cm > 195) | (df_bmi.Weight_kg > 150)]
4 df_p2
```

	Personal	Gender	Height_cm	Weight_kg	
	28	P29	Female	163	159
	29	P30	Male	179	152
	34	P35	Female	157	153
	36	P37	Female	197	114

```
1 # toán tử ~ - Not
2 df_p3 = df_bmi[~(df_bmi.Weight_kg < 155)]
3 df_p3
```

	Personal	Gender	Height_cm	Weight_kg	
	28	P29	Female	163	159
	65	P66	Female	179	158

Dr. Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

39/62

39

## A D A I 5. Filter Data

- Sử dụng phương thức **.isin()** để kết lọc dữ liệu theo một tập hợp

```
1 #Lọc ra những người có cân nặng bằng 150, 155 và 160kg
2 # phương thức isin (tương tự như in)
3 df_p4 = df_bmi[df_bmi.Weight_kg.isin([150,155,160])]
4 df_p4
```

	Personal	Gender	Height_cm	Weight_kg	
	102	P103	Male	161	155
	106	P107	Male	166	160
	123	P124	Female	184	160
	134	P135	Female	171	155
	135	P136	Female	183	150

Dr. Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

40/62

40

A D
A I

## 6. Tính toán min, max, mean, median, std, sum, cumsum

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

41/62

41

A D
A I

## 6. Đặc trưng thống kê trong DataFrame

- Sử dụng phương thức `.max()`, `.min()`, `.sum()`, `.mean()`, `.median()`, `.cumsum()`, `.std()` để tính các đặc trưng thống kê cho DataFrame hoặc theo từng cột.

```

1 #tìm Max, Min của thuộc tính cân nặng
2 w_max = df_bmi['Weight_kg'].max()
3 w_min = df_bmi['Weight_kg'].min()
4 print('Cân nặng lớn nhất:', w_max, '(kg)')
5 print('Cân nặng nhỏ nhất:', w_min, '(kg)')
```

Cân nặng lớn nhất: 160 (kg)  
Cân nặng nhỏ nhất: 50 (kg)

```

1 #tìm độ lệch chuẩn của chiều cao, cân nặng
2 h_std = df_bmi['Height_cm'].std()
3 w_std = df_bmi['Weight_kg'].std()
4 print('sdt của chiều cao:', h_std)
5 print('sdt của cân nặng:', w_std)
```

sdt của chiều cao: 16.37526067959376  
sdt của cân nặng: 32.38260746964435

```

1 #tìm Mean, Median của chiều cao
2 h_mean = df_bmi['Height_cm'].mean()
3 h_median = df_bmi['Height_cm'].median()
4 print('Chiều cao trung bình:', h_mean, '(cm)')
5 print('Trung vị:', h_median, '(cm)')
```

Chiều cao trung bình: 169.944 (cm)  
Trung vị: 170.5 (cm)

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

42/62

42

A
D
A
I

## 7. Xác định giá trị duy nhất (Unique)

Dr. Tà Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

43/62

43

A
D
A
I

## 7. Unique

- df.unique():** liệt kê danh sách các giá trị khác nhau trong một cột dữ liệu của DataFrame.
- df.value\_counts():** Tính tổng số theo từng giá trị khác nhau trong một cột dữ liệu của DataFrame. Kết quả là một đối tượng series.

```

1 #Xác định giá trị duy nhất trong một cột
2 df_bmi['Gender'].unique()

array(['Male', 'Female'], dtype=object)

1 #Thống kê số Lượng theo giá trị duy nhất
2 unique_gender = df_bmi['Gender'].value_counts()
3 unique_gender

Female    255
Male      245
Name: Gender, dtype: int64

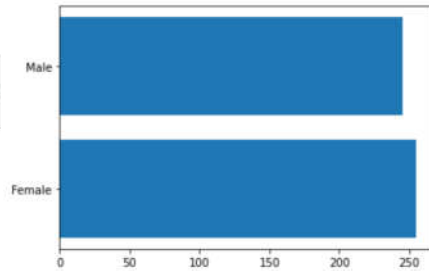
```

```

1 #Vẽ đồ thị thể hiện kết quả
2 plt.barh(unique_gender.index, unique_gender.values)

```

<BarContainer object of 2 artists>



Gender	Count
Male	245
Female	255

Dr. Tà Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

44/62

44

A D
A I

## Thực hành 2

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275
45/62

45

A D
A I

## Thực hành 2

**Yêu cầu 2.1:**

- **Đọc dữ liệu từ file Data\_Patient.csv vào biến kiểu dataframe: df\_patient với cột feature\_1 là cột chỉ số (index\_col)**
- **Đặt lại tên các cột dữ liệu trong Dataframe như sau:**

- Feature\_1 → Age
- Feature\_2 → Gender
- Feature\_3 → Type
- Feature\_4 → Blood\_pressure
- Feature\_5 → Cholesterol
- Feature\_6 → Heartbeat
- Feature\_7 → Thalassemia
- Feature\_8 → Result

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275
46/62

46

A

D

A

I

Thực hành 2

Yêu cầu 2.2:

?

- Lọc dữ liệu trong df\_patient thành các DataFrame:**
  - df\_male: chứa danh sách bệnh nhân Nam
  - df\_female: chứa danh sách bệnh nhân nữ
  - df\_no: danh sách những người không bị bệnh đau tim
  - df\_yes: danh sách những người bị bệnh đau tim

Yêu cầu 2.3:

- Lọc trong df\_patient đưa ra danh sách bệnh nhân thỏa mãn yêu cầu sau:**
  1. Những người bị mắc bệnh **đau tim** và trên **70 tuổi**
  2. Người có giới tính **Female**, có huyết áp trên **170 mmhg** nhưng **không bị bệnh đau tim**.
  3. Những người có triệu chứng đau ngực là **Typical angina**, giới tính **Male** và **bị bệnh đau tim**.

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

47/62

47

A

D

A

I

Thực hành 2

Yêu cầu 2.4: Xác định:

?

1. Chỉ số huyết áp (**Blood\_pressure**) thấp nhất, cao nhất, trung bình, trung vị và độ lệch chuẩn của tập dữ liệu
2. Chỉ số nhịp tim (**Heartbeat**) thấp nhất, cao nhất, trung bình, trung vị và độ lệch chuẩn của tập dữ liệu

1. Chỉ số huyết áp:

Min: 94

Max: 200

Mean: 131.68666666666667

Median: 130.0

Std: 17.682497692285477

2. Chỉ số nhịp tim:


Min: 71

Max: 202

Mean: 149.56333333333333

Median: 152.5

Std: 22.818595118151098



Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

48/62

48



A

D

A

I

## Thực hành 2

?

**Yêu cầu 2.5: Xác định:**

- Số giá trị khác nhau của thuộc tính **Type**
- Vẽ đồ thị dạng cột thể hiện kết quả thống kê số lượng theo từng giá trị khác nhau của thuộc tính Type

Asymptomatic	139
Non-anginal pain	84
Atypical angina	49
Typical angina	23

BIỂU ĐỒ THỐNG KÊ SỐ LƯỢNG THEO THUỘC TÍNH TYPE

Type	Count
Asymptomatic	139
Non-anginal pain	84
Atypical angina	49
Typical angina	23

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

49/62

49

A

D

A

I

## 7. Time series data

(Tiếp cận từ bài toán với dữ liệu thực tế)

<https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/03.11-Working-with-Time-Series.ipynb#scrollTo=eZUEgXlv7Nje>

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

50/62

50

A

D

A

I

7.1 Mô tả bài toán

**Fremont Bridge**

Trang web | Chỉ đường | Lưu

4.4 ★★★★★ 127 đánh giá trên Google

Cầu ở Portland, Oregon

Được dịch từ tiếng Anh - Cầu Fremont là cây cầu vòm bằng thép bắc qua sông Willamette nằm ở Portland, Oregon, Hoa Kỳ. Nó mang giao thông Interstate 405 và US 30 giữa trung tâm thành phố và Bắc Portland, nơi giao nhau với Xa lộ liên tiểu bang 5.

[Wikipedia \(tiếng Anh\)](#)

Xem mô tả gốc


Địa chỉ: Stadium Fwy, Portland, OR 97232, Hoa Kỳ

Chiều cao: 116 m

Bắt đầu xây dựng: 1968

Tổng chiều dài: 656 m


Khoảng hở bên dưới: 53 m



+ Người ta lắp đặt thiết bị để đếm số lượng xe đạp đi qua chiều phía đông và phía tây của cây cầu Fremont Bridge theo từng giờ.

+ Chi tiết:

<https://data.seattle.gov/Transportation/Fremont-Bridge-Bicycle-Counter/65db-xm6k>



Dr. Tạ Quang Chiếu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

51/62

51

A

D

A

I

7.2 Tập dữ liệu

- Tập dữ liệu là số lượng xe đạp đi qua cây cầu Fremont Bridge. Dữ liệu này được thu thập tự động thông qua các cảm biến ở 2 lối đi bộ ở phía đông và phía tây của cây cầu. Số lượng xe đạp được tổng hợp theo từng giờ.
- Tập dữ liệu bao gồm 4 cột:
  - Date: Thời gian (ngày - giờ): 10/03/2012 12:00:00 AM (Kiểu thời gian)
  - Fremont Bridge Total: Tổng số xe đi theo cả 2 lối đông và tây (Kiểu số nguyên)
  - Fremont Bridge East Sidewalk: Số xe đạp đi qua lối phía đông của cầu tương ứng với thời gian (Kiểu số nguyên)
  - Fremont Bridge West Sidewalk: Số xe đạp đi qua lối phía tây của cầu tương ứng với thời gian (Kiểu số nguyên)



Date	Fremont Bridge Total	Fremont Bridge East Sidewalk	Fremont Bridge West Sidewalk
10/03/2012 12:00:00 AM	13	4	9
10/03/2012 01:00:00 AM	10	4	6
10/03/2012 02:00:00 AM	2	1	1
10/03/2012 03:00:00 AM	5	2	3
10/03/2012 04:00:00 AM	7	6	1
10/03/2012 05:00:00 AM	31	21	10
10/03/2012 06:00:00 AM	155	105	50
10/03/2012 07:00:00 AM	352	257	95
10/03/2012 08:00:00 AM	437	291	146
10/03/2012 09:00:00 AM	276	172	104
10/03/2012 10:00:00 AM	118	72	46
10/03/2012 11:00:00 AM	42	10	32
10/03/2012 12:00:00 PM	76	35	41
10/03/2012 01:00:00 PM	90	42	48
10/03/2012 02:00:00 PM	128	77	51
10/03/2012 03:00:00 PM	164	72	92
10/03/2012 04:00:00 PM	315	133	182

Dr. Tạ Quang Chiếu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

52/62

52

A


D

A

I

## 7.3 Mục tiêu

- Phân tích dữ liệu chuỗi thời gian (Time Series Data) sử dụng Pandas.
- Kết hợp với các biểu đồ để tìm ra được những Insight ẩn chứa trong tập dữ liệu.



Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

53/62

53

A

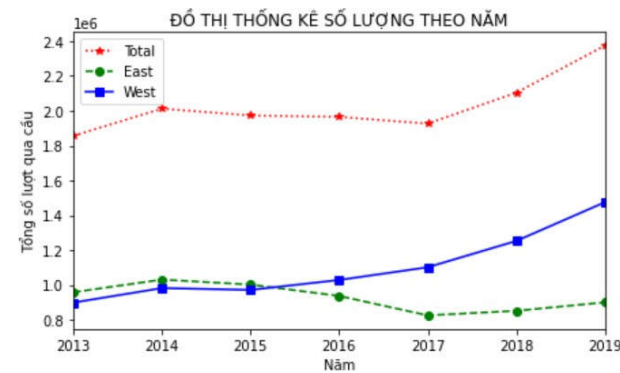
D

A

I

## 7.4 Kết quả

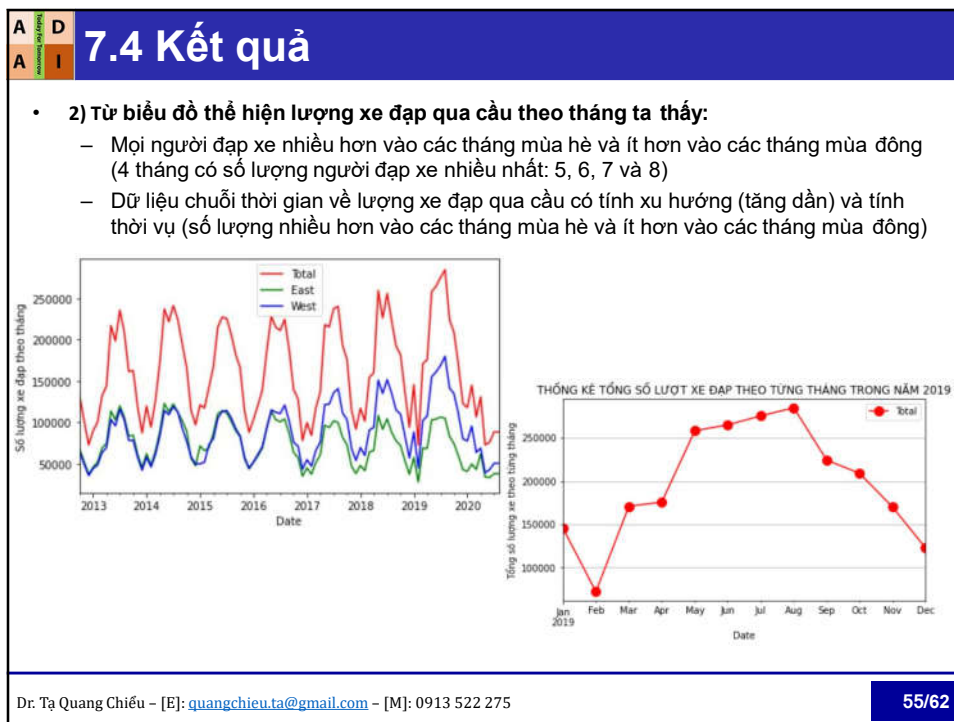
- 1) Từ biểu đồ thống kê tổng số xe đạp qua cầu theo năm ta thấy:**
  - Số lượng người đi xe đạp qua cầu Fremont có xu hướng tăng lên theo từng năm, những năm gần đây tăng nhanh.
  - Lượt xe đạp qua lối đi phía tây nhiều hơn lối đi phía đông, và cũng có xu hướng tăng nhanh trong những năm gần đây.



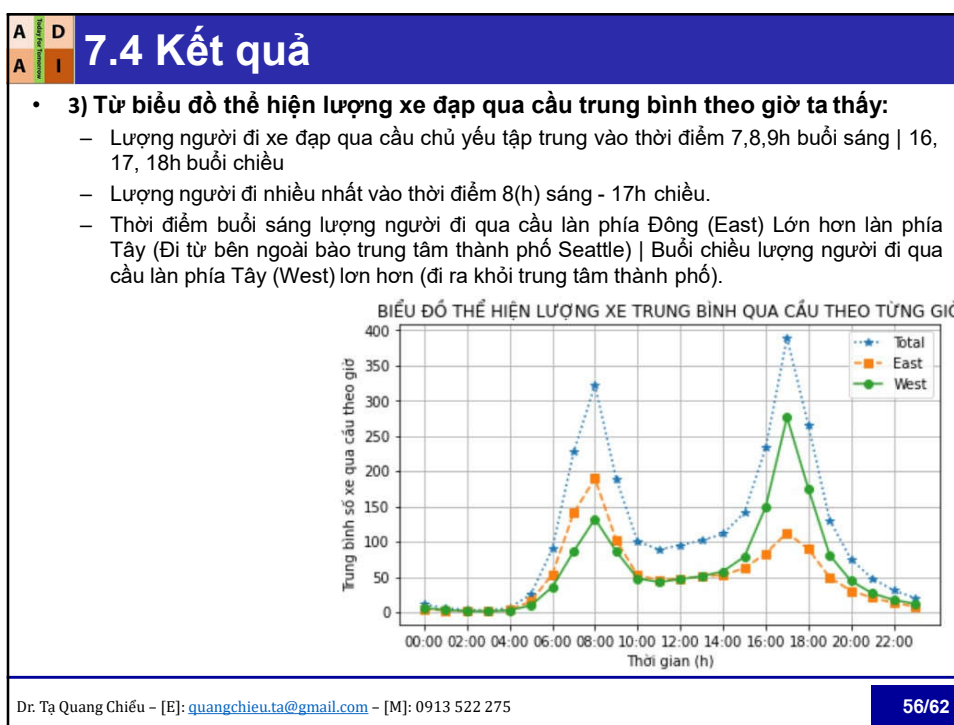
Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

54/62

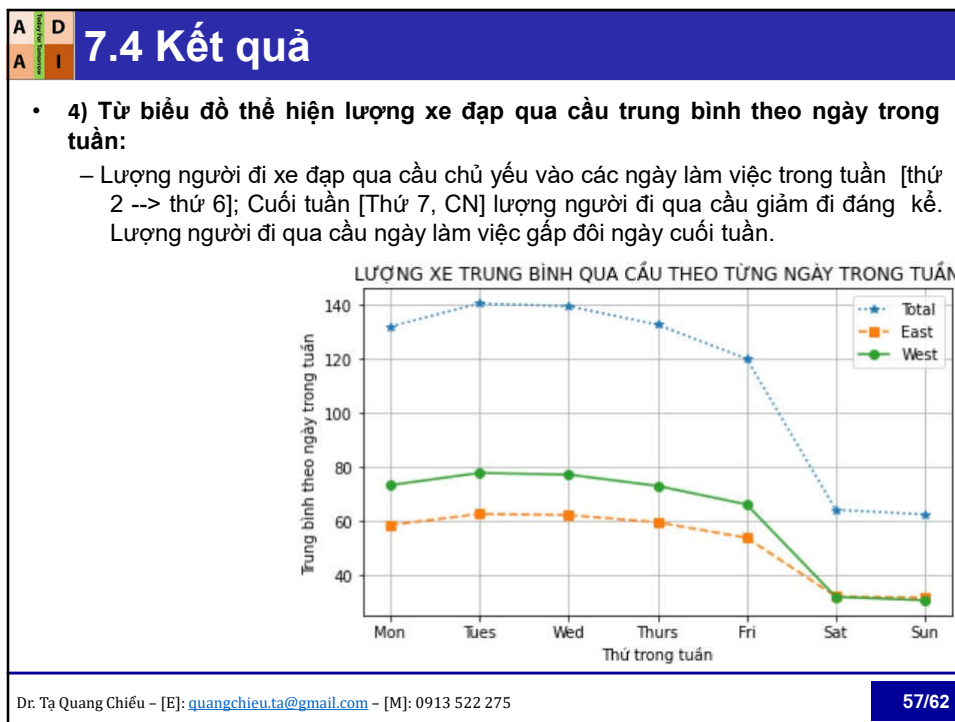
54



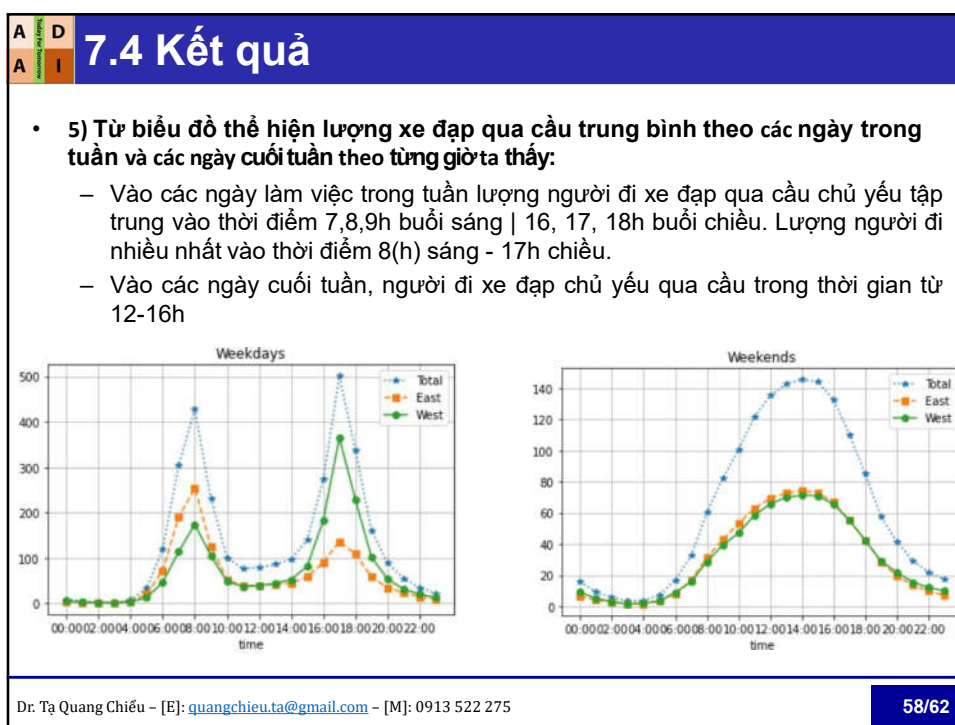
55



56



57



58

A large blue rectangle with the text "Thực hành 3" (Practical 3) in the center. The text is in a bold, black, sans-serif font. The rectangle is bordered by a thin white line on the left and bottom. In the top-left corner, there is a small logo consisting of a blue square with the letter 'A' and a red square with the letter 'D', and below them, a blue square with the letter 'A' and a red square with the letter 'I'.

59

A
D

A
I

# Thực hành 3

Dựa vào dữ liệu chuỗi thời gian quan trắc thông số nhiệt độ của Hà Giang và Cà Mau từ năm 2012 đến 2019 (Data\_Temperature.csv), sử dụng các kỹ thuật để tìm ra những thông tin từ 2 bộ dữ liệu đó.

TimeVN	HaGiang	CaMau
2012-08-01 7:00	26.2	25.4
2012-08-01 10:00	31.4	30.8
2012-08-01 13:00	35.1	30
2012-08-01 16:00	35.8	27.7
2012-08-01 19:00	30.4	28.1
2012-08-01 22:00	24	27.6
2012-08-02 1:00	23.8	24.4
2012-08-02 4:00	23.9	24.5
2012-08-02 7:00	24	25.4
2012-08-02 10:00	28.8	23.3
2012-08-02 13:00	34.7	24
2012-08-02 16:00	26.7	27.5
2012-08-02 19:00	25.4	27.2
2012-08-02 22:00	24.8	26.6
2012-08-03 1:00	24.5	26.5
2012-08-03 4:00	24.4	26.4
2012-08-03 7:00	24.6	26.4
2012-08-03 10:00	30.1	26.5
2012-08-03 13:00	34.4	30.6
2012-08-03 16:00		28.9
2012-08-03 19:00		27.6

Data\_Temperature
➡

Dr. Tạ Quang Chiếu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

**60/62**

60

