Nội dung

1. I/O – Đọc/ghi dữ liệu
2. Làm sạch dữ liệu
3. Trực quan hóa dữ liệu
4. Phương thức thao tác dữ liệu
5. Gộp dữ liệu
6. **Phương thức thao tác trên chuỗi**
7. Phương thức thao tác trên Timestamps
8. Tổng kết

Phương thức thao tác trên chuỗi

❑ Tách dữ liệu chuỗi thành list: dùng

`df["Tên_cột].str.split("ký_tự_tách")`

- Ví dụ

```
df_merge = pd.merge(df_left, df_right, how='inner')
print(df_merge)
```

```
  _key1 _key2   city user_name hide_date profession
0    K0    z0 city_0    user_0       h_0        p_0
1    K1    z1 city_1    user_1       h_1        p_1
2    K2    z2 city_2    user_2       h_2        p_2
3    K3    z3 city_3    user_3       h_3        p_3
```

```
city = df_merge['city'].str.split('_')
print(city)
```

```
0    [city, 0]
1    [city, 1]
2    [city, 2]
3    [city, 3]
Name: city, dtype: object
```

```
city[2][1]
```

```
'2'
```

Phương thức thao tác trên chuỗi

- Tìm chuỗi có nằm trong chuỗi hay không: dùng

  `df["Tên_cột"].str.contains("chuỗi")`

  - Ví dụ

```
city_find = df_merge['city'].str.contains("2")
print(city_find)
```

```
0    False
1    False
2     True
3    False
Name: city, dtype: bool
```

Phương thức thao tác trên chuỗi

- **Thay chuỗi bằng chuỗi: dùng**

  `df["Tên_cột"].str.replace("chuỗi_cũ", "chuỗi_mới")`

  - Ví dụ

```
city_new = df_merge['city'].str.replace('_', ' No ')
print(city_new)
```

```
0    city No 0
1    city No 1
2    city No 2
3    city No 3
Name: city, dtype: object
```

Phương thức thao tác trên chuỗi

- **Tìm chuỗi đầu tiên thỏa regular expression (RE): dùng**

  `df["tên_cột"].str.extract('RE')`

  - Ví dụ:

```
city_extract = df_left['city'].str.extract('([a-z]{0,})', expand=True)
print(type(city_extract))
print(city_extract)
```

```
<class 'pandas.core.frame.DataFrame'>
      0
0  city
1  city
2  city
3  city
```

```
city_num = df_left['city'].str.extract('(\d)', expand=False)
print(type(city_num))
print(city_num)
```

```
<class 'pandas.core.series.Series'>
0    0
1    1
2    2
3    3
Name: city, dtype: object
```

Regular Expression

- RegEx là chuỗi ký tự đặc biệt để so khớp hoặc so sánh chuỗi thỏa điều kiện nào đó.
- Ví dụ:
  - ^a...s$
  - [0-9]{2,4}
- Để sử dụng thư viện RegEx : **import re**

Regular Expression

- Một số ký hiệu:
  - Hoặc : |
  - Nhóm : ()
  - Số lượng ký tự : ?*+{m,n}
  - Ký tự đánh dấu: ^ $
  - Ký tự meta : . [] [-][^]
  - Ký tự : \d\D\w\W...
- Ví dụ:
  - “cat|mat” ~ “cat” or “mat”
  - “gr(e|a)y” ~ “grey” or “gray”

Regular Expression

- Số lượng ký tự: ?*+{m,n}
- Ví dụ:
  - “colou?r” ~ “colour” or “color”
  - “94*9” ~ “99” or “9449” or “944449”
  - “36+40” ~ “3640” or “366640”
  - “go{2,3}gle” ~ “google” or “gooogle”
  - “9{3}” ~ “999”
  - “s{2,} ~”ss” or “sss” or “sssss” ....

Regular Expression

- Ký tự đánh dấu : ^ $
- Ví dụ:
  - “^object” ~ “object” or “object-oriented” ...
  - “^2020” ~ “2020” or “2020/01/05” ...
  - “er$” ~ “driver” or “programer” ...
  - “2019$” ~ “2019” or “05/01/2019” ...

Regular Expression

- Ký tự meta : . [] [-][^]
- Ví dụ:
  - “87.1” ~ “8721” or “8731” or “8751”
  - “[xyz]” ~ “x” or “y” or “z”
  - “[a-zA-Z]” -> tất cả ký tự (chữ hoa, chữ thường)
  - “[^0-9]” -> Không lấy các ký số từ 0-9

Regular Expression

- Ký tự : \d\D\w\W...
- Ví dụ:
  - \d : ký số [0-9]
  - \D : không phải ký số
  - \s : ký tự đơn là tab(\t), newline (\n), khoảng trắng (\v)
  - \w : ký tự [a-zA-Z0-9_]
  - \w+ : 1 hoặc nhiều ký tự [a-zA-Z0-9_]

https://www.tutorialspoint.com/python/python_reg_expressions.htm

Regular Expression

```
# Giới tính - 
df['female'] = df['data'].str.extract('(\d)', expand=True)
df
```

	data	female
0	Arizona 1 2014-12-23 3242.0	1
1	Iowa 1 2010-02-23 3453.7	1
2	Oregon 0 2014-06-20 2123.0	0
3	Maryland 0 2014-03-14 1123.6	0
4	Florida 1 2013-01-15 2134.0	1
5	Georgia 0 2012-07-14 2345.6	0

```
# Nơi đăng ký
df['state'] = df['data'].str.extract('([A-Z]\w{0,})', expand=True)
df
```

	data	female	date	score	state
0	Arizona 1 2014-12-23 3242.0	1	2014-12-23	3242.0	Arizona
1	Iowa 1 2010-02-23 3453.7	1	2010-02-23	3453.7	Iowa
2	Oregon 0 2014-06-20 2123.0	0	2014-06-20	2123.0	Oregon
3	Maryland 0 2014-03-14 1123.6	0	2014-03-14	1123.6	Maryland
4	Florida 1 2013-01-15 2134.0	1	2013-01-15	2134.0	Florida
5	Georgia 0 2012-07-14 2345.6	0	2012-07-14	2345.6	Georgia

	data
0	Arizona 1 2014-12-23 3242.0
1	Iowa 1 2010-02-23 3453.7
2	Oregon 0 2014-06-20 2123.0
3	Maryland 0 2014-03-14 1123.6
4	Florida 1 2013-01-15 2134.0
5	Georgia 0 2012-07-14 2345.6