BÀI THỰC HÀNH TUẦN 4

1. ĐỌC FILE
2. import pandas as pd
3. import matplotlib.pyplot as plt
4. import random
5. df =pd.read\_csv('./orginal\_sales\_data\_edit.csv')

#Đọc 5 dòng đầu

df.head(5)

#Doc 5 dong cuoi

df.tail(5)

#Chuyen kieu

df["YEAR\_ID"]=df["YEAR\_ID"].astype("int32")

#loai bo cac dong trung nhau

df.drop\_duplicates(inplace=True)

#loai bo nhung dong trong

df['ADDRESSLINE2'].isna().sum()

#loai bo cac dong khong biet

df["ADDRESSLINE2"].fillna('Unknown',inplace=True)

df['ADDRESSLINE2'].isna().sum()

#lay du lieu 1 cot theo dang chuoi

df["QUANTITYORDERED"]

#Lay du lieu ve mot mang

df["QUANTITYORDERED"].values

#Lay du lieu tu dong so 4 den dong 9

df[4:10]

#doc du lieu tu 4 den 9

df.loc[4:10]

df.iloc[4:10]

# Lay thong tin tai dong co chi so la 2

df.loc[2]

#Lay thong tin tu dong 4 den 10 cua mot so cot

df.loc[4:10,['QUANTITYORDERED','SALES']]

#lay tu dong 2 den 9, tu coy 4 den 7

df.iloc[2:9,4:7]

Su dung loc nhieu hon vi ap dung cho ca hai

#Lay tai chi so 2

df.iloc[2]

#Lay tu dong dau tien den 9

df.iloc[:10]

#lay tu dong 2 den 9, tu cot 4 den 7

df.iloc[2:9,4:7]

#sap xep du lieu theo sales tang dan

df.sort\_values(by='SALES',ascending=True)

# sort theo nhieu tieu chi

df.sort\_values(by=['QUANTITYORDERED','PRICEEACH'],ascending=[True,False])

#Loc du lieu co  1 dieu kien

df[df['SALES']>5000]

df.loc[df['SALES']>5000]

#Lọc dữ liệu theo nhiều điều kiện

df[(df['SALES']>5000) & (df['QUANTITYORDERED']>40)]

Thường dùng kiểu loc hơn

#Loc gia1 tri va gan dieu kien

df.loc[df['PRICEEACH']<65,'FLAG']='CHEAP'

#Loc gia1 tri va gan dieu kien

df.loc[df['PRICEEACH']>=65,'FLAG']='EXPENSIVE'

df[['FLAG']]

#Viết hàm trả về giá trị có nhiề điều kiện và áp dụng hàm gán trị trả về cho 1 cột

def foo(x):

    if x<10:

        return 'BAD'

    elif x>=10 and x<50:

        return 'GOOD'

    else:

        return 'EXCELLENT'

df['WORTH']=df[['QUANTITYORDERED']].applymap(foo)

df[['QUANTITYORDERED','WORTH']]

#Ánh xạ giá trị tới 1 cột

dict\_map ={1:'Qui\_1',2:'Qui\_2',3:'Qui\_3',4:'Qui\_4'}

df['QTR\_ID']=df['QTR\_ID'].map(dict\_map)

df['QTR\_ID']

#lay nhung dong du lieu bang 1 dieu kien nao do

df[['QUANTITYORDERED','PRICEEACH']].loc[df['YEAR\_ID']==2003]

#Cac ham tong hop

#Lay theo so luong, don gia nho nhat

df.aggregate({'QUANTITYORDERED':['sum','min'],'PRICEEACH':['min','max','mean'],'SALES':['sum','mean']})

#Tong hop co phan nhom

df.groupby(['DEALSIZE'])['SALES'].agg(['sum','mean'])

df.groupby(['DEALSIZE','QTR\_ID'])['SALES','QUANTITYORDERED'].agg(['sum','mean','std'])

std: hàm tính độ lệch chuẩn

Tạo bảng Pivot Table

pd.pivot\_table(df,values='SALES',columns='YEAR\_ID',aggfunc=['sum','mean'])

pd.pivot\_table(df,values=['SALES','QUANTITYORDERED'],columns=['YEAR\_ID','QTR\_ID'],aggfunc=['sum','mean'])

df.groupby(['DEALSIZE','QTR\_ID'])['SALES','QUANTITYORDERED'].agg(['sum','mean','std'])

---Them

#Doc file du lieu voi cac tham so

df=pd.read\_csv('orginal\_sales\_data\_edit.csv',encoding='utf-8')

#Xem chieu dai cua df, tuong duong shape [0]

print('Len:', len(df))

#Xem thong tin dataframe vừa đọc được

df.info()

#Xem kich thuoc cua dataframe

print('Shape:',df.shape)

#Lay theo nhieu cot

df[['SALES','QUANTITYORDERED']]

df[['SALES','QUANTITYORDERED']].head(5)

#Lay theo hang va cot mong muon

df[['SALES','QUANTITYORDERED']][:10]

#Lay ban ghi co dieu kien

pp=df[df['QUANTITYORDERED']>30]

pp[:5]

#Loc 5 dong co Sales>3000

pri =df[df.SALES>3000]

pri.head(5)

#So sanh chuoi

nam\_compare=df['CONTACTLASTNAME'].str.contains('Brown')

nam\_compare.head(10)

#Lay gia tri tra ve numpy arrays

df['SALES'].values

#Them 1 cot dua vao du lieu da co

df['TIENBAN']=df['SALES']<5000

df.head(5)

#Khoi tao cot moi co gia tri rong

df['TINHTRANGDONHANG']=None

df['TINHTRANGDONHANG']

#Sua gia tri cot

df['TINHTRANGDONHANG']='DA THANH TOAN'

#XOA COT TRONG DATAFRAME

df.drop('TINHTRANGDONHANG',axis=1)

#Xoa nhieu cot dung lenh df.drop(['cot 1','cot 2'],axis=1)

df.drop(columns=['B','C'])#xoa cac cot co ten la B va C

df.drop([0,1])#Xoa ban ghi co chi so la 1 va 2

#Sử dụng hàm describe() cho bạn các thống kê cơ bản về dữ liệu:

df.describe()

#Xem cụ thể hơn trên từng cột như sau:

df['SALES'].count()

#Dem moi gia tri xuat hien bao nhieu lan

df['SALES'].value\_counts()

#vẽ đồ thị xem phân bố giá trị của một trường trong dataframe

import matplotlib.pyplot as plt

df['PRODUCTLINE'].value\_counts().plot(kind ='bar')

#Ve do thi dang pie

df['PRODUCTLINE'].value\_counts().plot(kind ='pie')

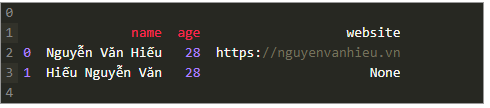
**Tạo mới dataframe từ python dictionary**

peoples = {'name': ['Nguyễn Văn Hiếu', 'Hiếu Nguyễn Văn'], 'age': [28, 28], 'website': ['https://nguyenvanhieu.vn', None]}

df = pd.DataFrame(peoples)

print(df)

Bạn sẽ có 1 dataframe như sau:



Lưu ý: các list của từng key trong từ điển phải có cùng kích thước.

**Tạo mới dataframe từ các python list**

txts = ['chỗ này ăn cũng khá ngon', 'ngon, nhất định sẽ quay lại', 'thái độ phục vụ quá tệ']

labels = [1, 1, 0]

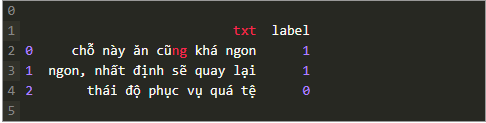
df = pd.DataFrame()

df['txt'] = txts

df['label'] = labels

print(df)

Bạn sẽ thu được dataframe như sau:

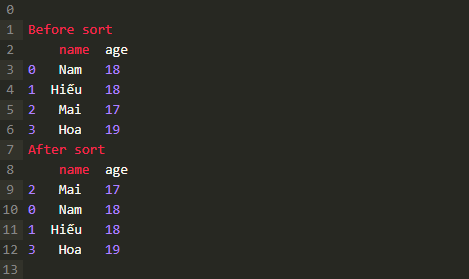


MỘT SỐ THAO TÁC TRÊN DATAFRAME

### Sắp xếp dataframe

Với thư viện pandas python, bạn có thể sắp xếp dataframe tăng dần, hay giảm dần theo 1 hoặc nhiều cột chỉ định.

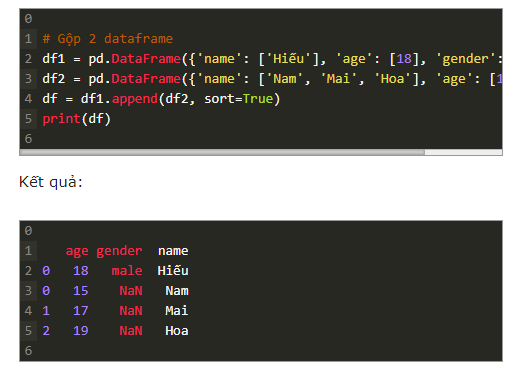
|  |  |
| --- | --- |
| 0  1  2  3  4  5  6 | # Sắp xếp df tăng dần theo cột nào đó  df = pd.DataFrame({'name': ['Nam', 'Hiếu', 'Mai', 'Hoa'], 'age': [18,18,17,19]})  print('Before sort\n', df)  df = df.sort\_values('age', ascending=True)  print('After sort\n', df)  Kết quả in ra được như sau: |



Bạn có thể sắp xếp theo nhiều cột có độ ưu tiên giảm dần, bằng cách truyền vào list tên cột. Ví dụ df.sort\_values(['age', 'name'], ascending=True)

### Nối 2 dataframe

Bạn có thể nối 2 dataframe thành 1 dataframe mới bằng cách sử dụng hàm append của thư viện pandas.



### Xáo trộn các bản ghi trong dataframe

Trong xử lý dữ liệu, bạn chắc chắn sẽ cần tới việc xáo trộn dữ liệu. Rất may, thư viện pandas có thể giúp chúng ta làm việc đó.

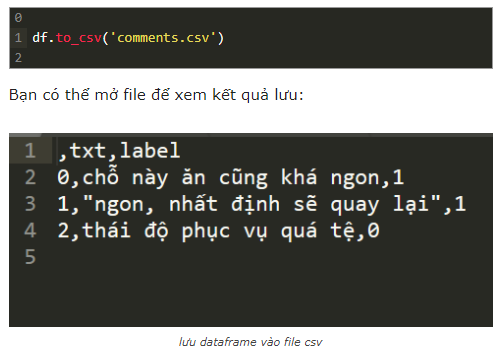


Giải thích thêm:

* *frac*: Chỉ định số bản ghi sẽ trả về ở mỗi lần random. Nếu bằng 1, tức là random ngẫu nhiên tất cả các bản ghi.
* *.reset\_index():*Sắp xếp lại cột chỉ số của dataframe.
* *drop*:với giá trị *True,*nó sẽ ngăn không cho hàm reset\_index tạo cột mới từ cột chỉ số của dataframe ban đầu.

## Lưu dataframe về file csv

Thư viện pandas python cho phép bạn lưu lại dataframe chỉ với một dòng code. Quá đơn giản phải không nào?

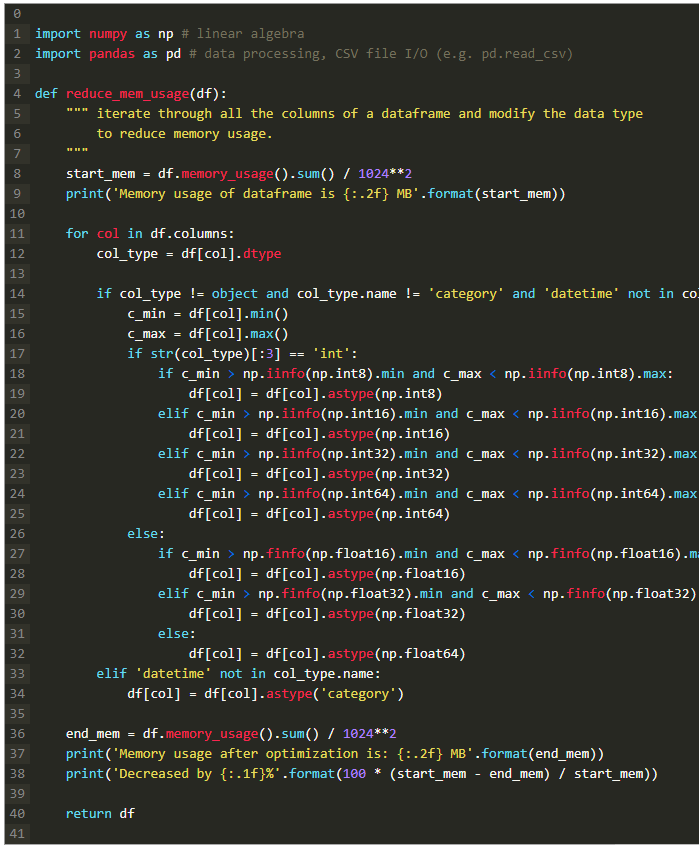


Các tham số của hàm to\_csv khá tương tự với hàm read\_csv. Bạn đọc có thể xem thêm thông tin đầy đủ của hàm này [tại đây](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.to_csv.html).

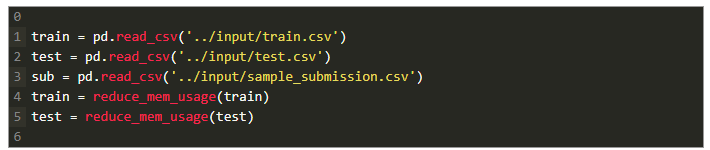
Tới đây mình xin kết thúc bài hướng dẫn về thư viện pandas python. Qua bài viết này, tôi tin chắc bạn đã có được những kiến thức cần thiết và có thể làm chủ thư viện pandas trong python. Bạn cũng có thể xem một ví dụ thực tế sử dụng thư viện này tại bài viết [code thuật toán linear regression này](https://nguyenvanhieu.vn/thuat-toan-linear-regression/).

## Tôi ưu bộ nhớ khi dùng pandas

Mục này mình đọc được từ bài của Anh Lê Huỳnh Đức trên Forum machinelearningcoban, xin được trích dẫn lại:



Chạy thử:



BÀI THAM KHẢO

* [Thư viện pandas python là gì?](https://blog.luyencode.net/thu-vien-pandas-python/#thu-vien-pandas-python-la-gi)
  + [Tại sao lại dùng thư viện pandas?](https://blog.luyencode.net/thu-vien-pandas-python/#tai-sao-lai-dung-thu-vien-pandas)
* [Đọc file csv sử dụng thư viện pandas](https://blog.luyencode.net/thu-vien-pandas-python/#doc-file-csv-su-dung-thu-vien-pandas)
  + - [Đọc csv file vào dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#doc-csv-file-vao-dataframe)
* [Thao tác với dataframe trong pandas](https://blog.luyencode.net/thu-vien-pandas-python/#thao-tac-voi-dataframe-trong-pandas)
  + [Xem thông tin của dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#xem-thong-tin-cua-dataframe)
  + [Truy xuất dữ liệu trên dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#truy-xuat-du-lieu-tren-dataframe)
    - [Lấy 1 cột theo tên cột](https://blog.luyencode.net/thu-vien-pandas-python/#lay-1-cot-theo-ten-cot)
    - [Lấy theo nhiều cột](https://blog.luyencode.net/thu-vien-pandas-python/#lay-theo-nhieu-cot)
    - [Lấy bản ghi theo chỉ số](https://blog.luyencode.net/thu-vien-pandas-python/#lay-ban-ghi-theo-chi-so)
    - [Lấy các bản ghi theo điều kiện](https://blog.luyencode.net/thu-vien-pandas-python/#lay-cac-ban-ghi-theo-dieu-kien)
    - [Lấy giá trị trả về numpy arrays](https://blog.luyencode.net/thu-vien-pandas-python/#lay-gia-tri-tra-ve-numpy-arrays)
  + [Thêm, sửa, xóa trong dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#them-sua-xoa-trong-dataframe)
    - [Thêm cột từ dữ liệu mới](https://blog.luyencode.net/thu-vien-pandas-python/#them-cot-tu-du-lieu-moi)
    - [Thêm cột dựa vào dữ liệu đã có](https://blog.luyencode.net/thu-vien-pandas-python/#them-cot-dua-vao-du-lieu-da-co)
    - [Khởi tạo cột mới có giá trị rỗng](https://blog.luyencode.net/thu-vien-pandas-python/#khoi-tao-cot-moi-co-gia-tri-rong)
    - [Thêm bản ghi trong dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#them-ban-ghi-trong-dataframe)
    - [Sửa giá trị của cột](https://blog.luyencode.net/thu-vien-pandas-python/#sua-gia-tri-cua-cot)
    - [Xóa cột trong dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#xoa-cot-trong-dataframe)
    - [Xóa bản ghi theo chỉ số](https://blog.luyencode.net/thu-vien-pandas-python/#xoa-ban-ghi-theo-chi-so)
* [Hiểu dữ liệu trong dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#hieu-du-lieu-trong-dataframe)
* [Tạo mới dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#tao-moi-dataframe)
* [Một số thao tác khác trên dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#mot-so-thao-tac-khac-tren-dataframe)
  + [Sắp xếp dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#sap-xep-dataframe)
  + [Nối 2 dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#noi-2-dataframe)
  + [Xáo trộn các bản ghi trong dataframe](https://blog.luyencode.net/thu-vien-pandas-python/#xao-tron-cac-ban-ghi-trong-dataframe)
* [Lưu dataframe về file csv](https://blog.luyencode.net/thu-vien-pandas-python/#luu-dataframe-ve-file-csv)
* [Tôi ưu bộ nhớ khi dùng pandas](https://blog.luyencode.net/thu-vien-pandas-python/#toi-uu-bo-nho-khi-dung-pandas)

Thư viện pandas python là gì?

Thư viện pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu. Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này. Chính sự linh hoạt và hiệu quả đã khiến cho pandas được sử dụng rộng rãi.

**Tại sao lại dùng thư viện pandas?**

* DataFrame đem lại sự linh hoạt và hiệu quả trong thao tác dữ liệu và lập chỉ mục;
* Là một công cụ cho phép đọc/ ghi dữ liệu giữa bộ nhớ và nhiều định dạng file: csv, text, excel, sql database, hdf5;
* Liên kết dữ liệu thông minh, xử lý được trường hợp dữ liệu bị thiếu. Tự động đưa dữ liệu lộn xộn về dạng có cấu trúc;
* Dễ dàng thay đổi bố cục của dữ liệu;
* Tích hợp cơ chế trượt, lập chỉ mục, lấy ra tập con từ tập dữ liệu lớn.
* Có thể thêm, xóa các cột dữ liệu;
* Tập hợp hoặc thay đổi dữ liệu với *group by*cho phép bạn thực hiện các toán tử trên tập dữ liệu;
* Hiệu quả cao trong trộn và kết hợp các tập dữ liệu;
* Lập chỉ mục theo các chiều của dữ liệu giúp thao tác giữa dữ liệu cao chiều và dữ liệu thấp chiều;
* Tối ưu về hiệu năng;
* Pandas được sử dụng rộng rãi trong cả học thuật và thương mại. Bao gồm thống kê, thương mại, phân tích, quảng cáo,…

Để cài đặt thư viện Pandas, bạn có thể làm theo một số cách khác nhau theo [tài liệu hướng dẫn](https://pandas.pydata.org/pandas-docs/stable/install.html):

1. Sử dụng pip

|  |  |
| --- | --- |
| 0  1  2 | pip install pandas |

1. Sử dụng conda

|  |  |
| --- | --- |
| 0  1  2 | conda install pandas |

Bây giờ chúng ta sẽ bắt đầu học cách sử dụng thư viện pandas python. Nhưng trước khi bắt đầu, hãy import thư viện pandas nhé. Chúng ta sẽ dùng cả thư viện matplotlib nữa.

Nếu bạn chưa biết về thư viện matplotlib, hãy đọc[bài viết này](https://nguyenvanhieu.vn/khoa-hoc-lap-trinh-python/#Matplotlib) trước nhé.

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6  7 | # Render our plots inline  %matplotlib inline    import pandas as pd  import matplotlib.pyplot as plt  import random |

Đọc file csv sử dụng thư viện pandas

Đọc csv file vào dataframe

Bạn có thể dễ dàng đọc vào một file .csv bằng cách sử dụng hàm read\_csv và được trả về 1 dataframe. Mặc định, hàm này sẽ phân biệt các trường của file csv theo dấu phẩy. Cách đọc hết sức đơn giản như sau:

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df = pd.read\_csv('./people.csv') |

Bạn có thể in ra n bản ghi đầu tiên của dataframe sử dụng hàm head. Ngược lại của hàm head là hàm tail

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df.head(5) |

Và kết quả in ra có dạng như sau:

đọc file csv sử dụng thư viện pandas

Tuy nhiên, bạn cũng sẽ phải lưu ý một vài tham số của hàm read\_csv như:

* *encoding*: chỉ định encoding của file đọc vào. Mặc định là *utf-8.*
* *sep*: thay đổi dấu ngăn cách giữa các cột. Mặc định là dấu phẩy (*‘,’)*
* *header*: chỉ định file đọc vào có header(tiêu đề của các cột) hay không. Mặc định là *infer*.
* *index\_col*: chỉ định chỉ số cột nào là cột chỉ số(số thứ tự). Mặc định là *None*.
* *nrows*: chỉ định số bản ghi sẽ đọc vào. Mặc định là *None –*đọc toàn bộ.

|  |  |
| --- | --- |
| 0  1  2  3 | peoples\_df = pd.read\_csv('./people.csv', encoding='utf-8', header=None, sep=',')  peoples\_df.head(5) |

Khi tôi chỉ định không có header, dòng header của chúng ta đã biến thành 1 bản ghi dữ liệu:

Bạn đọc có thể xem mô tả đầy đủ từng tham số của hàm read\_csv của thư viện pandas python [tại đây](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html).

Thao tác với dataframe trong pandas

Khi bạn đọc file csv như phía trên, peoples\_df sẽ là một đối tượng có hàng, cột gọi là DataFrame. Vậy nếu bạn muốn lấy giá trị theo hàng hoặc cột cụ thể nào đó thì làm như nào?

Thư viện pandas python cung cấp rất nhiều hàm hữu ích cho phép bạn thao tác với đối tượng dữ liệu dataframe này. Hãy cùng tôi đi khám phá sự hay ho này nhé.

Xem thông tin của dataframe

Bạn có thể xem thông tin của dataframe vừa đọc vào bằng cách sử dụng hàm .info() hoặc xem kích thước của dataframe này với thuộc tính shape. Cụ thể như sau:

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6  7 | # Xem chiều dài của df, tương đương shape[0]  print('Len:', len(peoples\_df))  # Xem thông tin dataframe vừa đọc được  peoples\_df.info()  # Xem kích thước của dataframe  print('Shape:', peoples\_df.shape) |

Và đây là những gì chúng ta sẽ thấy:

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | Len: 40  <class 'pandas.core.frame.DataFrame'>  RangeIndex: 40 entries, 0 to 39  Data columns (total 11 columns):  person\_ID    40 non-null int64  name         40 non-null object  first        40 non-null object  last         40 non-null object  middle       15 non-null object  email        40 non-null object  phone        40 non-null object  fax          40 non-null object  title        40 non-null object  age          40 non-null int64  is\_young     40 non-null bool  dtypes: bool(1), int64(2), object(8)  memory usage: 3.2+ KB  Shape: (40, 11) |

Truy xuất dữ liệu trên dataframe

Lấy 1 cột theo tên cột

Để chỉ định cột muốn lấy, bạn chỉ cần truyền vào tên cột như sau:

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df['name'] |

lấy 1 cột của dataframe dùng thư viện pandas python

Lấy theo nhiều cột

Thay vì truyền vào 1 string thì hãy truyền vào 1 list các tên cột. Mình thêm .head(5) để chỉ in ra 5 bản ghi đầu tiên cho ngắn, mặc định sẽ lấy hết.

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df[['name', 'age']].head(5) |

lấy nhiều cột trong dataframe

Lấy bản ghi theo chỉ số

Để lấy một hoặc nhiều bản ghi liên tiếp trong dataframe, sử dụng cơ chế trượt theo chỉ số giống như trên list trong python. Lấy 5 bản ghi đầu tiên:

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df[0:5] |

Trong trường hợp này kết quả giống như hàm head phía trên. Đều là lấy 5 bản ghi đầu tiên.

Bạn cũng có thể kết hợp lấy theo hàng và cột mong muốn:

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df[['name', 'age']][:5] |

Lấy các bản ghi theo điều kiện

|  |  |
| --- | --- |
| 0  1  2  3 | young\_pp = peoples\_df[peoples\_df['age'] < 25]  young\_pp[:5] |

lấy bản ghi dataframe theo điều kiện

Một ví dụ khác: Lấy tất cả các bản ghi chứa thông tin của người có chức danh là giáo sư:

|  |  |
| --- | --- |
| 0  1  2  3 | professor = peoples\_df[peoples\_df.title == 'Professor']  professor.head(5) |

Hoặc 1 ví dụ so sánh chuỗi như sau:

|  |  |
| --- | --- |
| 0  1  2  3 | name\_compare = peoples\_df['name'].str.contains('Rosella')  name\_compare.head(5) |

Kết quả thu được là một dataframe có 1 cột chưa 2 giá trị True hoặc False

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6  7 | 0     True  1    False  2    False  3    False  4    False  Name: name, dtype: bool |

Lấy giá trị trả về numpy arrays

Để lấy giá trị của một cột trả về dưới dạng numpy array trong thư viện pandas python, bạn chỉ cần thêm .values vào sau, ví dụ:

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df['name'].values |

Output bạn thu được như sau:

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | array(['Burks, Rosella ', 'Avila, Damien ', 'Olsen, Robin ',         'Moises, Edgar Estes', 'Brian, Heath Pruitt',         'Claude, Elvin Haney', 'Mosley, Edmund ', 'Derek, Antoine Mccoy',         'Hawkins, Callie ', 'Pate, Andrea ', 'Austin, Liz ',         'Kendrick, Reba Alford', 'Sims, Angelina ', 'Mullins, Kimberly ',         'Chuck, Lloyd Haney', 'Payne, Ladonna ',         'Baxter, Johnathan Browning', 'Weiss, Gilbert ',         'Deirdre, Florence Barrera', 'Fernando, Toby Calderon',         'Garrison, Patrica ', 'Effie, Leila Vinson', 'Buckley, Rose ',         'Stanton, Kathie ', 'Banks, Shannon ', 'Barnes, Cleo ',         'Brady, Nellie ', 'Katheryn, Ruben Holt', 'Michael, Dianne ',         'Grant, Adam ', 'Head, Kurtis ', 'Berger, Jami ',         'Earline, Jaime Fitzgerald', 'Evelyn, Summer Frost',         'Quentin, Sam Hyde', 'Dunlap, Ann ', 'Shields, Rich Pena',         'Page, Winnie ', 'Sparks, Ezra ', 'Kaufman, Elba '], dtype=object) |

Nếu bạn quan tâm tới numpy array, hãy đọc bài hướng dẫn về numpy của tôi [tại đây nhé](https://nguyenvanhieu.vn/khoa-hoc-lap-trinh-python/#Numpy).

Thêm, sửa, xóa trong dataframe

Thêm cột từ dữ liệu mới

Để thêm cột vào một dataframe có sẵn. Trước tiên, bạn cần có 1 list dữ liệu tương ứng với cột mà bạn muốn thêm. Tức là chiều dài của list phải tương ứng với số bản ghi của dataframe bạn muốn thêm.

Ở đây, tôi sẽ sử dụng thư viện random để sinh ngẫu nhiên một list năm sinh và thêm vào dataframe như sau:

|  |  |
| --- | --- |
| 0  1  2  3  4  5 | df\_len = len(peoples\_df)  birthday = [random.randrange(1980, 2000, 1) for i in range(df\_len)]  peoples\_df['birthday'] = birthday  peoples\_df.tail(5) |

thêm cột vào dataframe pandas python

Thêm cột dựa vào dữ liệu đã có

Giả sử ở đây mình muốn thêm cột is\_young có giá trị True nếu tuổi < 25 và False trong trường hợp còn lại.

|  |  |
| --- | --- |
| 0  1  2  3 | peoples\_df['is\_young'] = peoples\_df['age'] < 25  peoples\_df.head(5) |

thêm cột vào dataframe pandas

Khởi tạo cột mới có giá trị rỗng

Sử dụng cú pháp đơn giản như dưới đây, bạn sẽ có một trường mới và tất cả các giá trị là None

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df['new\_column'] = None |

Thêm bản ghi trong dataframe

Về vấn đề thêm bản ghi, chúng ta thường ít khi sử dụng nên tôi sẽ không trình bày. Bạn đọc quan tâm có thể đọc thêm tại [tài liệu này](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.append.html).

Sửa giá trị của cột

Để sửa giá trị của 1 cột, bạn làm tương tự như thêm mới cột. Nhưng khác với thêm ở chỗ là tên cột bạn truyền vào đã có trong dataframe. Còn thêm là 1 tên trường mới hoàn toàn chưa có. Chẳng hạn, bạn muốn thay đổi trường **name,**bạn chỉ cần làm như sau:

|  |  |
| --- | --- |
| 0  1  2  3  4 | peoples\_df['name'] = xxx #list các tên mới có chiều dài bằng chiều dài của dataframe  // Hoặc reset trường name về None  peoples\_df['name'] = None |

Xóa cột trong dataframe

Bạn có thể sử dụng một trong các cách sau:

|  |  |
| --- | --- |
| 0  1  2  3  4 | peoples\_df.drop('tên cột cần xóa', axis=1) # Xóa 1 cột  peoples\_df.drop(['cột 1', 'cột 2'], axis=1) # Xóa nhiều cột  df.drop(columns=['B', 'C']) # Xóa các cột có tên là B và C |

Xóa bản ghi theo chỉ số

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df.drop([0, 1]) # Xóa bản ghi ở chỉ số 1 và 2 |

Hiểu dữ liệu trong dataframe

Thư viện pandas python cung cấp cho bạn một số hàm giúp bạn hiểu về cấu trúc, phân bố của dữ liệu. Dưới đây là cách để bạn khám phá và hiểu dữ liệu của mình.

Tôi đã bổ sung trường age vào file people.csv và tiến hành đọc lại.

|  |  |  |  |
| --- | --- | --- | --- |
| 0  1  2 | peoples\_df['age'] | | |
| 0  1  2  3  4  5  6  7  8  9  10  11  12  13 | | 0     25  1     23  2     21  3     18  4     30  5     35  .  .  .  38    25  39    25  Name: age, dtype: int64 |

Sử dụng hàm describe() cho bạn các thống kê cơ bản về dữ liệu:

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df.describe() |

xem thống kê dữ liệu trong dataframe

Xem cụ thể hơn trên từng cột như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| 0  1  2 | peoples\_df['age'].value\_counts() | | |
| 0  1  2  3  4  5  6  7  8  9  10  11  12 | | 25    21  28     6  35     3  29     2  23     2  22     2  30     1  24     1  21     1  18     1  Name: age, dtype: int64 |

Bạn cũng có thể vẽ đồ thị xem phân bố giá trị của một trường trong dataframe như sau.

|  |  |
| --- | --- |
| 0  1  2 | peoples\_df['age'].value\_counts().plot(kind='bar') |

Tạo mới dataframe

Có một vài cách để tạo ra dataframe trong thư viện pandas python. Bạn có thể dùng cách mà bạn cho là dễ sử dụng, đôi khi cũng phải tùy vào từng trường hợp mà nên chọn cách nào nữa.

**Tạo mới dataframe từ python dictionary**

|  |  |
| --- | --- |
| 0  1  2  3  4 | peoples = {'name': ['Nguyễn Văn Hiếu', 'Hiếu Nguyễn Văn'], 'age': [28, 28], 'website': ['https://nguyenvanhieu.vn', None]}  df = pd.DataFrame(peoples)  print(df) |

Bạn sẽ có 1 dataframe như sau:

|  |  |
| --- | --- |
| 0  1  2  3  4 | name  age                   website  0  Nguyễn Văn Hiếu   28  https://nguyenvanhieu.vn  1  Hiếu Nguyễn Văn   28                      None |

Lưu ý: các list của từng key trong từ điển phải có cùng kích thước.

**Tạo mới dataframe từ các python list**

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6  7 | txts = ['chỗ này ăn cũng khá ngon', 'ngon, nhất định sẽ quay lại', 'thái độ phục vụ quá tệ']  labels = [1, 1, 0]  df = pd.DataFrame()  df['txt'] = txts  df['label'] = labels  print(df) |

Bạn sẽ thu được dataframe như sau:

|  |  |
| --- | --- |
| 0  1  2  3  4  5 | txt  label  0     chỗ này ăn cũng khá ngon      1  1  ngon, nhất định sẽ quay lại      1  2       thái độ phục vụ quá tệ      0 |

Lưu ý: các list này phải có cùng kích thước.

Một số thao tác khác trên dataframe

Sắp xếp dataframe

Với thư viện pandas python, bạn có thể sắp xếp dataframe tăng dần, hay giảm dần theo 1 hoặc nhiều cột chỉ định.

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6 | # Sắp xếp df tăng dần theo cột nào đó  df = pd.DataFrame({'name': ['Nam', 'Hiếu', 'Mai', 'Hoa'], 'age': [18,18,17,19]})  print('Before sort\n', df)  df = df.sort\_values('age', ascending=True)  print('After sort\n', df) |

Kết quả in ra được như sau:

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6  7  8  9  10  11  12  13 | Before sort      name  age  0   Nam   18  1  Hiếu   18  2   Mai   17  3   Hoa   19  After sort      name  age  2   Mai   17  0   Nam   18  1  Hiếu   18  3   Hoa   19 |

Bạn có thể sắp xếp theo nhiều cột có độ ưu tiên giảm dần, bằng cách truyền vào list tên cột. Ví dụ df.sort\_values(['age', 'name'], ascending=True)

Nối 2 dataframe

Bạn có thể nối 2 dataframe thành 1 dataframe mới bằng cách sử dụng hàm append của thư viện pandas.

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6 | # Gộp 2 dataframe  df1 = pd.DataFrame({'name': ['Hiếu'], 'age': [18], 'gender': ['male']})  df2 = pd.DataFrame({'name': ['Nam', 'Mai', 'Hoa'], 'age': [15,17,19]})  df = df1.append(df2, sort=True)  print(df) |

Kết quả:

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6 | age gender  name  0   18   male  Hiếu  0   15    NaN   Nam  1   17    NaN   Mai  2   19    NaN   Hoa |

Xáo trộn các bản ghi trong dataframe

Trong xử lý dữ liệu, bạn chắc chắn sẽ cần tới việc xáo trộn dữ liệu. Rất may, thư viện pandas có thể giúp chúng ta làm việc đó.

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6 | # Xáo trộn các hàng trong df - shuffle dataframe rows  df = pd.DataFrame({'name': ['Hiếu', 'Nam', 'Mai', 'Hoa'], 'age': [18,15,17,19]})  print('Before shuffle\n', df)  df = df.sample(frac=1).reset\_index(drop=True)  print('After shuffle\n', df) |

Kết quả(Có thể khác nhau nhé – xáo trộn mà)

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6  7  8  9  10  11  12  13 | Before shuffle      name  age  0  Hiếu   18  1   Nam   15  2   Mai   17  3   Hoa   19  After shuffle      name  age  0   Mai   17  1   Nam   15  2  Hiếu   18  3   Hoa   19 |

Giải thích thêm:

* *frac*: Chỉ định số bản ghi sẽ trả về ở mỗi lần random. Nếu bằng 1, tức là random ngẫu nhiên tất cả các bản ghi.
* *.reset\_index():*Sắp xếp lại cột chỉ số của dataframe.
* *drop*:với giá trị *True,*nó sẽ ngăn không cho hàm reset\_index tạo cột mới từ cột chỉ số của dataframe ban đầu.

Lưu dataframe về file csv

Thư viện pandas python cho phép bạn lưu lại dataframe chỉ với một dòng code. Quá đơn giản phải không nào?

|  |  |
| --- | --- |
| 0  1  2 | df.to\_csv('comments.csv') |

Bạn có thể mở file để xem kết quả lưu:

lưu dataframe vào file csv

Các tham số của hàm to\_csv khá tương tự với hàm read\_csv. Bạn đọc có thể xem thêm thông tin đầy đủ của hàm này [tại đây](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.to_csv.html).

Tới đây mình xin kết thúc bài hướng dẫn về thư viện pandas python. Qua bài viết này, tôi tin chắc bạn đã có được những kiến thức cần thiết và có thể làm chủ thư viện pandas trong python. Bạn cũng có thể xem một ví dụ thực tế sử dụng thư viện này tại bài viết [code thuật toán linear regression này](https://nguyenvanhieu.vn/thuat-toan-linear-regression/).

Tôi ưu bộ nhớ khi dùng pandas

Mục này mình đọc được từ bài của Anh Lê Huỳnh Đức trên Forum machinelearningcoban, xin được trích dẫn lại:

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | import numpy as np # linear algebra  import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)    def reduce\_mem\_usage(df):      """ iterate through all the columns of a dataframe and modify the data type          to reduce memory usage.      """      start\_mem = df.memory\_usage().sum() / 1024\*\*2      print('Memory usage of dataframe is {:.2f} MB'.format(start\_mem))        for col in df.columns:          col\_type = df[col].dtype            if col\_type != object and col\_type.name != 'category' and 'datetime' not in col\_type.name:              c\_min = df[col].min()              c\_max = df[col].max()              if str(col\_type)[:3] == 'int':                  if c\_min > np.iinfo(np.int8).min and c\_max < np.iinfo(np.int8).max:                      df[col] = df[col].astype(np.int8)                  elif c\_min > np.iinfo(np.int16).min and c\_max < np.iinfo(np.int16).max:                      df[col] = df[col].astype(np.int16)                  elif c\_min > np.iinfo(np.int32).min and c\_max < np.iinfo(np.int32).max:                      df[col] = df[col].astype(np.int32)                  elif c\_min > np.iinfo(np.int64).min and c\_max < np.iinfo(np.int64).max:                      df[col] = df[col].astype(np.int64)              else:                  if c\_min > np.finfo(np.float16).min and c\_max < np.finfo(np.float16).max:                      df[col] = df[col].astype(np.float16)                  elif c\_min > np.finfo(np.float32).min and c\_max < np.finfo(np.float32).max:                      df[col] = df[col].astype(np.float32)                  else:                      df[col] = df[col].astype(np.float64)          elif 'datetime' not in col\_type.name:              df[col] = df[col].astype('category')        end\_mem = df.memory\_usage().sum() / 1024\*\*2      print('Memory usage after optimization is: {:.2f} MB'.format(end\_mem))      print('Decreased by {:.1f}%'.format(100 \* (start\_mem - end\_mem) / start\_mem))        return df |

Chạy thử:

|  |  |
| --- | --- |
| 0  1  2  3  4  5  6 | train = pd.read\_csv('../input/train.csv')  test = pd.read\_csv('../input/test.csv')  sub = pd.read\_csv('../input/sample\_submission.csv')  train = reduce\_mem\_usage(train)  test = reduce\_mem\_usage(test) |