

[Lab 2] Instructions

Name: Nguyễn Thành Trung – MSSV: 19522431

Link github: [Data_Mining/Week2 at main · Shu2301/Data_Mining \(github.com\)](https://github.com/Shu2301/Data_Mining)

The screenshot displays a Jupyter Notebook with the following content:

Cell [1]: Imports libraries and loads data.

```
%matplotlib inline
import numpy as np
import pandas as pd

df = pd.read_csv("PastHires.csv")
df.head()
```

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
0	10	Y	4	BS	N	N	Y
1	0	N	0	BS	Y	Y	Y
2	7	N	6	BS	N	N	N
3	2	Y	1	MS	Y	N	Y
4	20	N	2	PhD	Y	N	N

Cell [2]: Displays the first 10 rows of the dataset.

```
df.head(10)
```

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
0	10	Y	4	BS	N	N	Y
1	0	N	0	BS	Y	Y	Y
2	7	N	6	BS	N	N	N
3	2	Y	1	MS	Y	N	Y
4	20	N	2	PhD	Y	N	N
5	0	N	0	PhD	Y	Y	Y
6	5	Y	2	MS	N	Y	Y
7	3	N	1	BS	N	Y	Y
8	15	Y	5	BS	N	N	Y

Cell [3]: Displays the last 4 rows of the dataset.

```
df.tail(4)
```

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
9	0	N	0	BS	N	N	N
10	1	N	1	PhD	Y	N	N
11	4	Y	1	BS	N	Y	Y
12	0	N	0	PhD	Y	N	Y

Cell [4]: Displays the shape of the DataFrame.

```
df.shape
```

(13, 7)

Cell [5]: Displays the size of the DataFrame.

```
df.size
```

91

Cell [6]: Displays the length of the DataFrame.

```
len(df)
```

13

Cell [7]: Displays the columns of the DataFrame.

```
df.columns
```

Index(['Years Experience', 'Employed?', 'Previous employers', 'Level of Education', 'Top-tier school', 'Interned', 'Hired'], dtype='object')

Cell [8]: Displays the 'Hired' column.

```
df['Hired']
```

	Hired
0	Y
1	Y
2	N

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

11 Y
12 Y
Name: Hired, dtype: object

[9] df['Hired'][:5]

0 giây

0 Y
1 Y
2 N
3 Y
4 N
Name: Hired, dtype: object

[10] df['Hired'][5]

0 giây

'Y'

[11] df[['Years Experience', 'Hired']]

0 giây

	Years Experience	Hired
0	10	Y
1	0	Y
2	7	N
3	2	Y
4	20	N
5	0	Y
6	5	Y
7	3	Y
8	15	Y
9	0	N

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

11 4 Y
12 0 Y

[12] df[['Years Experience', 'Hired']][:5]

0 giây

	Years Experience	Hired
0	10	Y
1	0	Y
2	7	N
3	2	Y
4	20	N

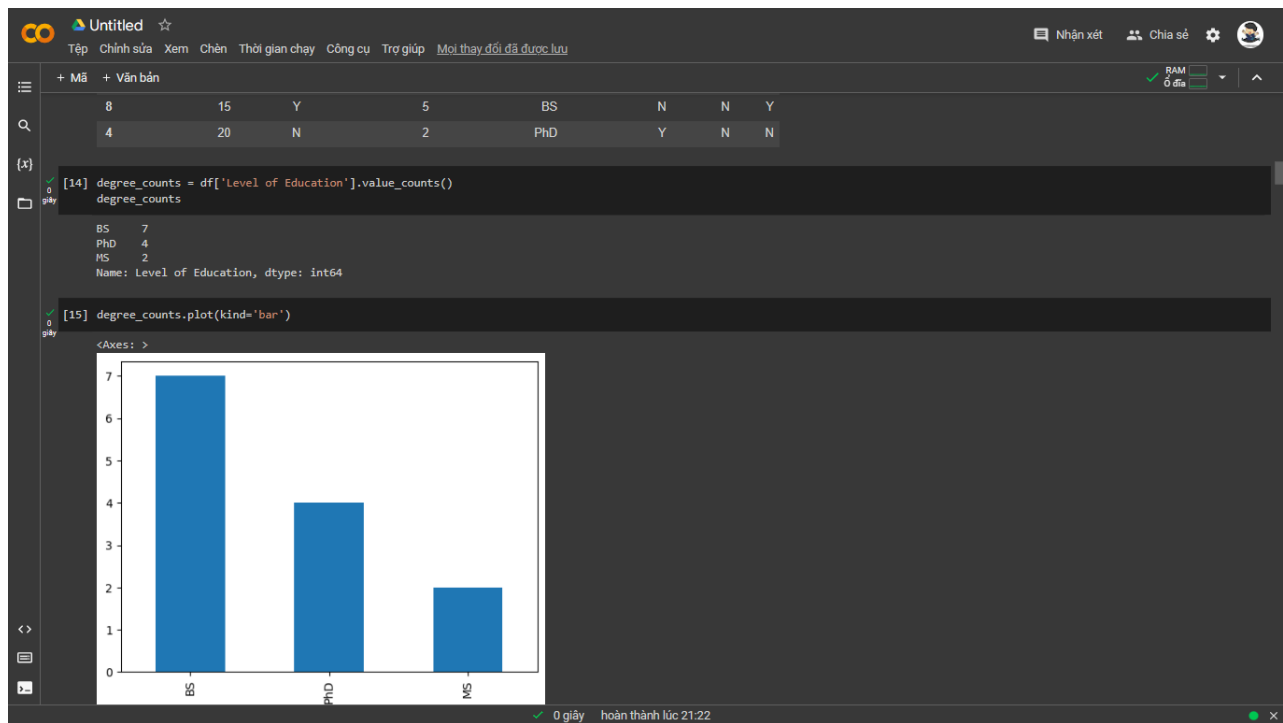
[13] df.sort_values(['Years Experience'])

0 giây

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
1	0	N	0	BS	Y	Y	Y
5	0	N	0	PhD	Y	Y	Y
9	0	N	0	BS	N	N	N
12	0	N	0	PhD	Y	N	Y
10	1	N	1	PhD	Y	N	N
3	2	Y	1	MS	Y	N	Y
7	3	N	1	BS	N	Y	Y
11	4	Y	1	BS	N	Y	Y
6	5	Y	2	MS	N	Y	Y
2	7	N	6	BS	N	N	N

0 giây

hoàn thành lúc 21:22



Untitled

Tệp | Chính sửa | Xem | Chèn | Thời gian chạy | Công cụ | Trợ giúp | Mọi thay đổi đã được lưu

+ Mã + Văn bản

RAM 0 giây

```
[16] import numpy as np
import pandas as pd
```

```
[17] labels = ['a','b','c']
my_list = [10,20,30]
arr = np.array([10,20,30])
d = {'a':10,'b':20,'c':30}
```

```
[18] pd.Series(data=my_list)
0    10
1    20
2    30
dtype: int64
```

```
[19] pd.Series(data=my_list, index=labels)
a     10
b     20
c     30
dtype: int64
```

```
[20] pd.Series(my_list, labels)
a     10
b     20
c     30
dtype: int64
```

```
[21] pd.Series(arr)
0     10
1     20
2     30
dtype: int64
```

0 giây | hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

RAM
Ổ đĩa

[22] pd.Series(arr, labels)

0 giây

a 10
b 20
c 30
dtype: int64

[23] pd.Series(d)

0 giây

a 10
b 20
c 30
dtype: int64

[24] pd.Series(data=labels)

0 giây

0 a
1 b
2 c
dtype: object

[25] pd.Series([sum,print,len])

0 giây

0 <built-in function sum>
1 <built-in function print>
2 <built-in function len>
dtype: object

[26] ser1=pd.Series([1,2,3,4],index = ['USA','Germany','USSR','Japan'])

0 giây

[27] ser1

0 giây

USA 1
Germany 2
USSR 3
Japan 4
dtype: int64

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

RAM
Ổ đĩa

[28] ser2=pd.Series([1,2,3,4],index = ['USA','Germany','Italy','Japan'])

0 giây

[29] ser2

0 giây

USA 1
Germany 2
Italy 3
Japan 4
dtype: int64

[30] ser1['USA']

0 giây

1

[31] ser1+ser2

0 giây

Germany 4.0
Italy NaN
Japan 8.0
USA 2.0
USSR NaN
dtype: float64

[32] import pandas as pd
import numpy as np

0 giây

[33] from numpy.random import randn
np.random.seed(101)

0 giây

[34] df = pd.DataFrame(randn(5,4),index='A B C D E'.split(), columns='W X Y Z'.split())

0 giây

[35] df

0 giây

W X Y Z

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

[36] df['W']

A 2.706850
B 0.651118
C -2.018168
D 0.188695
E 0.190794
Name: W, dtype: float64

[37] df[['W', 'Z']]

	W	Z
A	2.706850	0.503826
B	0.651118	0.605965
C	-2.018168	-0.589001
D	0.188695	0.955057
E	0.190794	0.683509

[38] df.W

A 2.706850
B 0.651118
C -2.018168
D 0.188695
E 0.190794
Name: W, dtype: float64

[39] type(df['W'])

pandas.core.series.Series

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

[40] df['new'] = df['W'] + df['Y']
df

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

[41] df.drop('new', axis=1)
df

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

[42] df.drop('new', axis=1, inplace=True)
df

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

[43] df.drop('E',axis=0)

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057

0 giây

[44] df.loc['A']

W	2.706850
X	0.628133
Y	0.907969
Z	0.503826

Name: A, dtype: float64

0 giây

[45] df.iloc[2]

W	-2.018168
X	0.740122
Y	0.528813
Z	-0.589001

Name: C, dtype: float64

0 giây

[46] df.loc['B','Y']

-0.8480769834036315

0 giây

[47] df.loc[['A','B'],['W','Y']]

	W	Y
A	2.706850	0.907969

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

[48] df

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

0 giây

[49] df>0

	W	X	Y	Z
A	True	True	True	True
B	True	False	False	True
C	False	True	True	False
D	True	False	False	True
E	True	True	True	True

0 giây

[50] df[df>0]

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	NaN	NaN	0.605965
C	NaN	0.740122	0.528813	NaN

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

RAM
Ổ đĩa

[51] df[df['W']>0]

0 giây

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

[52] df[df['W']>0]['Y']

0 giây

A	0.907969
B	-0.848077
D	-0.933237
E	2.605967

Name: Y, dtype: float64

[53] df[df['W']>0][['Y','X']]

0 giây

	Y	X
A	0.907969	0.628133
B	-0.848077	-0.319318
D	-0.933237	-0.758872
E	2.605967	1.978757

[54] df[(df['W']>0) & (df['Y']>1)]

0 giây

	W	X	Y	Z
--	---	---	---	---

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

RAM
Ổ đĩa

[55] df

0 giây

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

[56] df.reset_index()

0 giây

index	W	X	Y	Z	
0	A	2.706850	0.628133	0.907969	0.503826
1	B	0.651118	-0.319318	-0.848077	0.605965
2	C	-2.018168	0.740122	0.528813	-0.589001
3	D	0.188695	-0.758872	-0.933237	0.955057
4	E	0.190794	1.978757	2.605967	0.683509

[57] newind='CA NY WY OR CO'.split()

0 giây

[58] df['States'] = newind

0 giây

[59] df

0 giây

	W	X	Y	Z	States
--	---	---	---	---	--------

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

```
[60] df.set_index('States')
```

	W	X	Y	Z
States				
CA	2.706850	0.628133	0.907969	0.503826
NY	0.651118	-0.319318	-0.848077	0.605965
WY	-2.018168	0.740122	0.528813	-0.589001
OR	0.188695	-0.758872	-0.933237	0.955057
CO	0.190794	1.978757	2.605967	0.683509

```
[61] df
```

	W	X	Y	Z	States
A	2.706850	0.628133	0.907969	0.503826	CA
B	0.651118	-0.319318	-0.848077	0.605965	NY
C	-2.018168	0.740122	0.528813	-0.589001	WY
D	0.188695	-0.758872	-0.933237	0.955057	OR
E	0.190794	1.978757	2.605967	0.683509	CO

```
[62] df.set_index('States', inplace = True)
```

```
df
```

	W	X	Y	Z
States				
CA	2.706850	0.628133	0.907969	0.503826

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

```
[63] outside = ['G1','G1', 'G1','G2', 'G2','G2']
inside = [1,2,3,1,2,3]
hier_index = list(zip(outside,inside))
hier_index=pd.MultiIndex.from_tuples(hier_index)
```

```
[64] hier_index
```

```
MultiIndex([(('G1', 1),
              ('G1', 2),
              ('G1', 3),
              ('G2', 1),
              ('G2', 2),
              ('G2', 3)),
            ])
```

```
[65] df = pd.DataFrame(np.random.randn(6,2),index=hier_index,columns=['A','B'])
df
```

		A	B
G1	1	0.302665	1.693723
	2	-1.706086	-1.159119
	3	-0.134841	0.390528
G2	1	0.166905	0.184502

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

30.6387870.329646

[66] df.loc['G1']

0 giây

	A	B
1	0.302665	1.693723
2	-1.706086	-1.159119
3	-0.134841	0.390528

[67] df.loc['G1'].loc[1]

0 giây

A	0.302665
B	1.693723

Name: 1, dtype: float64

[68] df.index.names

0 giây

FrozenList([None, None])

[69] df.index.names = ['Group', 'Num']

0 giây

df

		A	B
G1	1	0.302665	1.693723
	2	-1.706086	-1.159119
	3	-0.134841	0.390528
G2	1	0.166905	0.184502
	2	0.807706	0.072960

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

0 giây

[70] df.xs('G1')

0 giây

	A	B
1	0.302665	1.693723
2	-1.706086	-1.159119
3	-0.134841	0.390528

[71] df.xs('G1')

0 giây

	A	B
1	0.302665	1.693723
2	-1.706086	-1.159119
3	-0.134841	0.390528

[72] df.xs(['G1',1])

0 giây

<ipython-input-72-c549ee06ce91>:1: FutureWarning: Passing lists as key for xs is deprecated and will be removed in a future version. Pass key as a tuple instead.

df.xs(['G1',1])

A	0.302665
B	1.693723

Name: (G1, 1), dtype: float64

[73] df.xs(1,level='Num')

0 giây

	A	B
1	0.302665	1.693723
2	-1.706086	-1.159119
3	-0.134841	0.390528

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

0 giây

RAM
Ổ đĩa

[73] df.xs(1, level='num')

0 giây

	A	B
Group		
G1	0.302665	1.693723
G2	0.166905	0.184502

[74] import numpy as np
import pandas as pd

0 giây

[75] df = pd.DataFrame({'A':[1,2,np.nan], 'B':[5,np.nan,np.nan], 'C':[1,2,3]})
df

0 giây

	A	B	C
0	1.0	5.0	1
1	2.0	NaN	2
2	NaN	NaN	3

[76] df.dropna()

0 giây

	A	B	C
0	1.0	5.0	1

[77] df.dropna(axis=1)

0 giây

	C
0	1
1	2

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

0 giây

RAM
Ổ đĩa

[78] df.dropna(thresh=2)

0 giây

	A	B	C
0	1.0	5.0	1
1	2.0	NaN	2

[79] df.fillna(value='FILL VALUE')

0 giây

	A	B	C
0	1.0	5.0	1
1	2.0	FILL VALUE	2
2	FILL VALUE	FILL VALUE	3

[80] df['A'].fillna(value=df['A'].mean())

0 giây

	A
0	1.0
1	2.0
2	1.5

Name: A, dtype: float64

[81] import pandas as pd
data = {'Company':['GOOG', 'GOOG', 'MSFT', 'MSFT', 'FB', 'FB'],
 'Person': ['Sam', 'Charlie', 'Amy', 'Vanessa', 'Carl', 'Sarah'],
 'Sales':[200,120,340,124,243,350]}

0 giây

[82] df=pd.DataFrame(data)
df

0 giây

	Company	Person	Sales
--	---------	--------	-------

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

RAM
Ổ đĩa

0

GOOG

Sam

200

1

GOOG

Charlie

120

2

MSFT

Amy

340

3

MSFT

Vanessa

124

4

FB

Carl

243

5

FB

Sarah

350

0 giây

[83] df.groupby('Company')

<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7fec00185b50>

0 giây

[84] by_comp = df.groupby("Company")

0 giây

[85] by_comp.mean()

Sales

Company

FB296.5

GOOG160.0

MSFT232.0

0 giây

[86] df.groupby('Company').mean()

Sales

Company

FB296.5

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

RAM
Ổ đĩa

MSFT232.0

0 giây

[87] by_comp.min()

PersonSales

Company

FBCarl243

GOOGCharlie120

MSFTAmy124

0 giây

[88] by_comp.max()

PersonSales

Company

FBSarah350

GOOGSam200

MSFTVanessa340

0 giây

[89] by_comp.count()

PersonSales

Company

FB22

GOOG22

MSFT22

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

0 giây

✓

by_comp.describe()

Sales

	count	mean	std	min	25%	50%	75%	max
Company								
FB	2.0	296.5	75.660426	243.0	269.75	296.5	323.25	350.0
GOOG	2.0	160.0	56.568542	120.0	140.00	160.0	180.00	200.0
MSFT	2.0	232.0	152.735065	124.0	178.00	232.0	286.00	340.0

[91] by_comp.describe().transpose()

0 giây

✓

	Company	FB	GOOG	MSFT
Sales count	2.000000	2.000000	2.000000	
mean	296.500000	160.000000	232.000000	
std	75.660426	56.568542	152.735065	
min	243.000000	120.000000	124.000000	
25%	269.750000	140.000000	178.000000	
50%	296.500000	160.000000	232.000000	
75%	323.250000	180.000000	286.000000	
max	350.000000	200.000000	340.000000	

[92] by_comp.describe().transpose()['GOOG']

0 giây

✓

Sales	count	mean	std	min	25%
	2.000000	160.000000	56.568542	120.000000	140.000000

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

0 giây

✓

import pandas as pd

[94] df1 = pd.DataFrame({'A': ['A0', 'A1', 'A2', 'A3'],
 'B': ['B0', 'B1', 'B2', 'B3'],
 'C': ['C0', 'C1', 'C2', 'C3'],
 'D': ['D0', 'D1', 'D2', 'D3']},
 index=[0,1,2,3])

0 giây

✓

[95] df2 = pd.DataFrame({'A': ['A4', 'A5', 'A6', 'A7'],
 'B': ['B4', 'B5', 'B6', 'B7'],
 'C': ['C4', 'C5', 'C6', 'C7'],
 'D': ['D4', 'D5', 'D6', 'D7']},
 index=[4,5,6,7])

0 giây

✓

[96] df3 = pd.DataFrame({'A': ['A8', 'A9', 'A10', 'A11'],
 'B': ['B8', 'B9', 'B10', 'B11'],
 'C': ['C8', 'C9', 'C10', 'C11'],
 'D': ['D8', 'D9', 'D10', 'D11']},
 index=[8,9,10,11])

0 giây

✓

[97] df1

0 giây

✓

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

[98] df2

0 giây

✓

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

0 giây

✓

RAM
Ổ đĩa

0 giây

df1

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

0 giây

[99] df3

	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

0 giây

[100] pd.concat([df1,df1,df3])

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2

0 giâyhoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

0 giây

✓

RAM
Ổ đĩa

0 giây

[101] pd.concat([df1,df2,df3],axis=1)

	A	B	C	D	A	B	C	D	A	B	C	D
0	A0	B0	C0	D0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	A1	B1	C1	D1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	A2	B2	C2	D2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	A3	B3	C3	D3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	A4	B4	C4	D4	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	A5	B5	C5	D5	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	A6	B6	C6	D6	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	A7	B7	C7	D7	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A8	B8	C8	D8
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A9	B9	C9	D9
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A10	B10	C10	D10
11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A11	B11	C11	D11

0 giây

[102] left=pd.DataFrame({'key':['K0','K1','K2','K3'],
 'A':['A0','A1','A2','A3'],
 'B':['B0','B1','B2','B3']})
right=pd.DataFrame({'key':['K0','K1','K2','K3'],
 'C':['C0','C1','C2','C3'],
 'D':['D0','D1','D2','D3']})

0 giây

[103] left

key	A	B
-----	---	---

0 giâyhoàn thành lúc 21:22

The screenshot displays a Jupyter Notebook interface with a dark theme. The notebook is titled "Untitled" and has tabs for "Mã" and "Văn bản". The interface includes a top bar with navigation icons and a bottom bar showing the current cell's execution time and status.

The notebook contains the following code cells and their outputs:

```
[103] left
```

	key	A	B
0	K0	A0	B0
1	K1	A1	B1
2	K2	A2	B2
3	K3	A3	B3

```
[104] right
```

	key	C	D
0	K0	C0	D0
1	K1	C1	D1
2	K2	C2	D2
3	K3	C3	D3

```
[105] pd.merge(left,right,how='inner',on='key')
```

	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K2	A2	B2	C2	D2
3	K3	A3	B3	C3	D3

```
[106] left=pd.DataFrame({'key1':['K0','K0','K1','K2'],  
                        'key2':['K0','K1','K1','K1']})
```

```
[107] pd.merge(left, right, on=['key1','key2'])
```

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A2	B2	C1	D1
2	K1	K0	A2	B2	C2	D2

```
[108] pd.merge(left,right,how='outer', on=['key1','key2'])
```

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K0	K1	A1	B1	NaN	NaN
2	K1	K0	A2	B2	C1	D1
3	K1	K0	A2	B2	C2	D2
4	K2	K1	A3	B3	NaN	NaN
5	K2	K0	NaN	NaN	C3	D3

```
[109] pd.merge(left,right,how='left', on=['key1','key2'])
```

The image displays a Jupyter Notebook interface with two tabs. The top tab, titled 'Untitled', shows a series of pandas operations:

- Initial data frames:

	K0	K1	A0	B0	C0	D0
0	K0	K1	A1	B1	NaN	NaN
2	K1	K0	A2	B2	C1	D1
3	K1	K0	A2	B2	C2	D2
4	K2	K1	A3	B3	NaN	NaN
- Code cell [110]: `pd.merge(left, right, how='right', on=['key1', 'key2'])`
- Resulting merged DataFrame:

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A2	B2	C1	D1
2	K1	K0	A2	B2	C2	D2
3	K2	K0	NaN	NaN	C3	D3
- Code cell [111]:

```
left = pd.DataFrame({'A':['A0','A1','A2'],'B':['B0','B1','B2']],index=['K0','K1','K2'])
right = pd.DataFrame({'C':['C0','C2','C3'],'D':['D0','D2','D3']],index=['K0','K2','K3'])
```
- Code cell [112]: `left.join(right)`
- Resulting joined DataFrame:

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2
- Code cell [113]: `left.join(right, how='outer')`
- Resulting outer-joined DataFrame:

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2
K3	NaN	NaN	C3	D3

The bottom tab, also titled 'Untitled', shows further pandas operations:

- Code cell [114]:

```
import pandas as pd
df = pd.DataFrame({'col1':[1,2,3,4], 'col2':[444,555,666,444], 'col3':['abc',
'def','ghi','xyz']})
df.head()
```
- Resulting DataFrame:

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz
- Code cell [115]: `df['col2'].unique()`
- Result: `array([444, 555, 666])`
- Code cell [116]: `df['col2'].nunique()`
- Result: `3`
- Code cell [117]: `df['col2'].value_counts()`

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

RAM
Ổ đĩa

555 1
666 1
Name: col2, dtype: int64

[118] newdf = df[(df['col1']>2) & (df['col2']==444)]
[119] newdf
col1 col2 col3
3 4 444 xyz

[120] def times2(x):
return x*2
[121] df['col1'].apply(times2)
0 2
1 4
2 6
3 8
Name: col1, dtype: int64

[122] df['col3'].apply(len)
0 3
1 3
2 3
3 3
Name: col3, dtype: int64

[123] df['col1'].sum()
10

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

RAM
Ổ đĩa

[124] del df['col1']
[125] df
col2 col3
0 444 abc
1 555 def
2 666 ghi
3 444 xyz

[126] df.columns
Index(['col2', 'col3'], dtype='object')

[127] df.index
RangeIndex(start=0, stop=4, step=1)

[128] df
col2 col3
0 444 abc
1 555 def
2 666 ghi
3 444 xyz

[129] df.sort_values(by='col2')

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

RAM
Ổ đĩa

	col2	col3
0	444	abc
3	444	xyz
1	555	def
2	666	ghi

[130] df.isnull()

	col2	col3
0	False	False
1	False	False
2	False	False
3	False	False

[131] df.dropna()

	col2	col3
0	444	abc
1	555	def
2	666	ghi
3	444	xyz

[132] import numpy as np

[133] df = pd.DataFrame({'col1': [1, 2, 3, np.nan],
'col2': [np.nan, 555, 666, 444],
'col3': ['abc', 'def', 'ghi', 'xyz']})

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã+ Văn bản

RAM
Ổ đĩa

	col1	col2	col3
0	1.0	NaN	abc
1	2.0	555.0	def
2	3.0	666.0	ghi
3	NaN	444.0	xyz

[134] df.isnull()

	col1	col2	col3
0	False	True	False
1	False	False	False
2	False	False	False
3	True	False	False

[135] df.dropna()

	col1	col2	col3
1	2.0	555.0	def
2	3.0	666.0	ghi

[136] df.fillna("FILL")

	col1	col2	col3
0	1.0	FILL	abc
1	2.0	555.0	def

0 giây

hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

✓ 0 giây

RAM
Ổ đĩa

[137] data={'A':['foo','foo','foo','bar','bar','bar'],
 'B':['one','one','two','two','one','one'],
 'C':['x','y','x','y','x','y'],
 'D':[1,3,2,5,4,1]}
df = pd.DataFrame(data)

[138] df

	A	B	C	D
0	foo	one	x	1
1	foo	one	y	3
2	foo	two	x	2
3	bar	two	y	5
4	bar	one	x	4
5	bar	one	y	1

[139] df

	A	B	C	D
0	foo	one	x	1
1	foo	one	y	3
2	foo	two	x	2
3	bar	two	y	5
4	bar	one	x	4
5	bar	one	y	1

0 giây hoàn thành lúc 21:22

Untitled

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpMọi thay đổi đã được lưu

+ Mã + Văn bản

✓ 0 giây

RAM
Ổ đĩa

[140] df.pivot_table(values='D', index=['A','B'], columns=['C'])

		C	x	y
bar	one	4.0	1.0	
	two	NaN	5.0	
foo	one	1.0	3.0	
	two	2.0	NaN	

[141] import numpy as np
import pandas as pd

[142] df = pd.read_csv('PastHires.csv')

df

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
0	10	Y	4	BS	N	N	Y
1	0	N	0	BS	Y	Y	Y
2	7	N	6	BS	N	N	N
3	2	Y	1	MS	Y	N	Y
4	20	N	2	PhD	Y	N	N
5	0	N	0	PhD	Y	Y	Y
6	5	Y	2	MS	N	Y	Y
7	3	N	1	BS	N	Y	Y
8	15	Y	5	BS	N	N	Y
9	0	N	0	BS	N	N	N

0 giây hoàn thành lúc 21:22

The image displays two screenshots of a Jupyter Notebook environment. The top screenshot shows a notebook named "19522431-Lab2.ipynb" with a file explorer on the left containing "Mã" and "Văn bản". The main area contains two code cells: Cell [143] imports 'df' as 'to_csv' with index=False, and Cell [149] reads an Excel file 'job-market.xlsx'. Below the code is a preview of a DataFrame with 13 columns: Id, Title, Company, Date, Location, Area, Classification, SubClassification, Requirement, FullDescription, LowestSalary, and HighestSalary. It lists several retail jobs from Aldi Stores and other companies. The bottom screenshot shows another instance of the same notebook. Cell [150] imports pandas and reads 'job-market.csv'. Cell [151] prints the first 10 rows of the DataFrame. This view includes detailed information about the first 10 rows, such as company names (Aldi Stores, LB Creative Pty Ltd, Anaconda Group Pty Ltd, Commonwealth Bank), locations (Sydney, Brisbane, Melbourne), and classifications (Retail & Consumer Products, Call Centre & Customer Service).

19522431-Lab2.ipynb

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpKhông lưu được kể từ 22:58

+ Mã+ Văn bản

✓
0 giây

[152] import pandas as pd
df = pd.read_csv('job-market.csv')

✓
0 giây

[153] df.drop_duplicates(inplace=True)
df.dropna(inplace=True)

✓
0 giây

[154] import pandas as pd
import matplotlib.pyplot as plt

✓
1 giây

[155] df = pd.read_csv('job-market.csv')

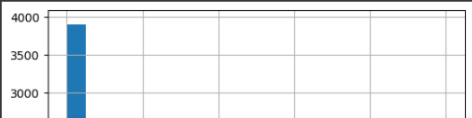
✓
0 giây

[156] print(df['LowestSalary'].describe())

count10099.000000
mean24.578671
std20.387614
min0.000000
25%0.000000
50%30.000000
75%40.000000
max50.000000
Name: LowestSalary, dtype: float64

✓
0 giây

[157] df['LowestSalary'].hist(bins=20)

<Axes: >


Không thể lưu tự động. Tệp này đã được cập nhật từ xa hoặc trong một thẻ khác. [Hiển thị sự khác biệt](#)

✓
0 giây

hoàn thành lúc 23:00

19522431-Lab2.ipynb

TệpChỉnh sửaXemChènThời gian chạyCông cụTrợ giúpKhông lưu được kể từ 22:58

+ Mã+ Văn bản


✓
0 giây

[158] print(df['HighestSalary'].describe())

count10099.000000
mean42.290326
std11.458869
min30.000000
25%30.000000
50%40.000000
75%50.000000
max60.000000
Name: HighestSalary, dtype: float64

✓
0 giây

[159] plt.scatter(df['LowestSalary'], df['HighestSalary'])

<matplotlib.collections.PathCollection at 0x7fcf814e1df0>


Không thể lưu tự động. Tệp này đã được cập nhật từ xa hoặc trong một thẻ khác. [Hiển thị sự khác biệt](#)

✓
0 giây

hoàn thành lúc 23:00

