jupyter Project 19 Fictitious Names Last Checkpoint: 15 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Fictitious Names

Introduction:

This time you will create a data again

Special thanks to Chris Albon for sharing the dataset and materials. All the credits to this exercise belongs to him.

In order to understand about it go here.

Step 1. Import the necessary libraries

In [1]:
```
import pandas as pd
import numpy as np
```

Step 2. Create the 3 DataFrames based on the following raw data

In [5]:
```
raw_data_1 = {
        'subject_id': ['1', '2', '3', '4', '5'],
        'first_name': ['Alex', 'Amy', 'Allen', 'Alice', 'Ayoung'],
        'last_name': ['Anderson', 'Ackerman', 'Ali', 'Aoni', 'Atiches']}

raw_data_2 = {
        'subject_id': ['4', '5', '6', '7', '8'],
        'first_name': ['Billy', 'Brian', 'Bran', 'Bryce', 'Betty'],
        'last_name': ['Bonder', 'Black', 'Balwner', 'Brice', 'Btisan']}

raw_data_3 = {
        'subject_id': ['1', '2', '3', '4', '5', '7', '8', '9', '10', '11'],
        'test_id': [51, 15, 15, 61, 16, 14, 15, 1, 61, 16]}
```

Step 3. Assign each to a variable called data1, data2, data3

In [8]:
```
d1 = pd.DataFrame(raw_data_1, columns = ['subject_id', 'first_name', 'last_name'])
d2 = pd.DataFrame(raw_data_2, columns = ['subject_id', 'first_name', 'last_name'])
d3 = pd.DataFrame(raw_data_3, columns = ['subject_id','test_id'])
```

In [9]:
```
d1
```

Out[9]:

	subject_id	first_name	last_name
0	1	Alex	Anderson
1	2	Amy	Ackerman
2	3	Allen	Ali
3	4	Alice	Aoni
4	5	Ayoung	Atiches

In [10]:
```
d2
```

Out[10]:

	subject_id	first_name	last_name
0	4	Billy	Bonder
1	5	Brian	Black
2	6	Bran	Balwner
3	7	Bryce	Brice
4	8	Betty	Btisan

In [11]:
```
d3
```

Out[11]:

	subject_id	test_id
0	1	51
1	2	15
2	3	15
3	4	61
4	5	16
5	7	14
6	8	15
7	9	1
8	10	61
9	11	16

Step 4. Join the two dataframes along rows and assign all_data

In [22]:
```
ad = pd.concat([d1,d2])
ad
```

Out[22]:

	subject_id	first_name	last_name
0	1	Alex	Anderson
1	2	Amy	Ackerman
2	3	Allen	Ali
3	4	Alice	Aoni
4	5	Ayoung	Atiches
0	4	Billy	Bonder
1	5	Brian	Black
2	6	Bran	Balwner
3	7	Bryce	Brice
4	8	Betty	Btisan

In [24]:
```
ad.describe()
```

Out[24]:

	subject_id	first_name	last_name
count	10	10	10
unique	8	10	10
top	4	Alex	Anderson
freq	2	1	1

Step 5. Join the two dataframes along columns and assing to all_data_col

In [15]:
```
adc = pd.concat([d1,d2], axis=1)
adc
```

Out[15]:

	subject_id	first_name	last_name	subject_id	first_name	last_name
0	1	Alex	Anderson	4	Billy	Bonder
1	2	Amy	Ackerman	5	Brian	Black
2	3	Allen	Ali	6	Bran	Balwner
3	4	Alice	Aoni	7	Bryce	Brice
4	5	Ayoung	Atiches	8	Betty	Btisan

Step 6. Print data3

In [25]:
```
d3
```

Out[25]:

	subject_id	test_id
0	1	51
1	2	15
2	3	15
3	4	61
4	5	16
5	7	14
6	8	15
7	9	1
8	10	61
9	11	16

In [26]:
```
d3.describe()
```

Out[26]:

	test_id
count	10.000000
mean	26.500000
std	22.122136
min	1.000000
25%	15.000000
50%	15.500000
75%	42.250000
max	61.000000

Step 7. Merge all_data and data3 along the subject_id value

In [17]:
```
pd.merge(ad, d3, on="subject_id")
```

Out[17]:

	subject_id	first_name	last_name	test_id
0	1	Alex	Anderson	51
1	2	Amy	Ackerman	15
2	3	Allen	Ali	15
3	4	Alice	Aoni	61
4	4	Billy	Bonder	61
5	5	Ayoung	Atiches	16
6	5	Brian	Black	16
7	7	Bryce	Brice	14
8	8	Betty	Btisan	15

Step 8. Merge only the data that has the same 'subject_id' on both data1 and data2

In [18]:
```
pd.merge(d1, d2, on='subject_id', how='inner')
```

Out[18]:

	subject_id	first_name_x	last_name_x	first_name_y	last_name_y
0	4	Alice	Aoni	Billy	Bonder
1	5	Ayoung	Atiches	Brian	Black

Step 9. Merge all values in data1 and data2, with matching records from both sides where available.

In [19]:
```
pd.merge(d1, d2, on='subject_id', how='outer')
```

Out[19]:

	subject_id	first_name_x	last_name_x	first_name_y	last_name_y
0	1	Alex	Anderson	NaN	NaN
1	2	Amy	Ackerman	NaN	NaN
2	3	Allen	Ali	NaN	NaN
3	4	Alice	Aoni	Billy	Bonder
4	5	Ayoung	Atiches	Brian	Black
5	6	NaN	NaN	Bran	Balwner
6	7	NaN	NaN	Bryce	Brice
7	8	NaN	NaN	Betty	Btisan