Python - Analiza danych z modułem PANDAS

www.udemy.com (http://www.udemy.com) (R)

LAB - S06-L004 - groupby i multiindex

1. Zaimportuj moduł pandas i numpy nadaj im standardowe aliasy. Zaimportuj też datetime, timedelta i time, możesz skorzystać z poniższych poleceń:

```
from datetime import datetime
from datetime import timedelta
import time
```

2. Do wykonania zadań będziemy korzystać z danych dotyczących maratonów. Uruchom poniższy kod, który przygotuje zmienną df o odpowiedniej strukturze:

```
df = pd.read_csv('./marathon_results_2016.csv', index_col='Bib',
usecols=['Bib','40K','Half','Pace','Age','M/F','Country',
'State','City'])

df = df[(df['40K'] != '-') & (df['Half'] != '-')]

df['40K'] = df['40K'].apply(pd.to_timedelta)
df['Half'] = df['Half'].apply(pd.to_timedelta)

df['TotalSeconds'] = df['40K'].apply(lambda x: timedelta.total_seco
nds(x))
df['HalfSeconds'] = df['Half'].apply(lambda x: timedelta.total_seco
nds(x))

df.head()
```

3. Utwórz obiekt grupy w oparciu o kolumny **"Country","City"** i nazwij go **country_city**
4. Korzystając z metody obiektu grupy wyznacz ilość obiektów w każdej utworzonej grupie
5. Wyznacz średnią z każdej grupy dla każdej wartości numerycznej
6. Pobierz dane odpowiadające za **'USA','San Francisco'**
7. Utwórz obiekt grupy w oparciu o kolumny **"M/F","Age"** i nazwij go **sex_age**
8. Korzystając z metody obiektu grupy wyznacz ilość obiektów w każdej utworzonej grupie
9. Wyznacz średnią z każdej grupy dla każdej wartości numerycznej
10. Pobierz dane odpowiadające za **"M",25**

Dane pochodzą z https://github.com/llimllib/bostonmarathon (https://github.com/llimllib/bostonmarathon) https://www.kaggle.com/rojour/boston-marathon-2016-finishers-analysis/data (https://www.kaggle.com/rojour/boston-marathon-2016-finishers-analysis/data)

Rozwiązania:

Poniżej znajdują się propozycje rozwiązań zadań. Prawdopodobnie istnieje wiele dobrych rozwiązań, dlatego jeżeli rozwiązujesz zadania samodzielnie, to najprawdopodobniej zrobisz to inaczej, może nawet lepiej :) Możesz pochwalić się swoimi rozwiązaniami w sekcji Q&A

In [1]:
```
import pandas as pd
import numpy as np
from datetime import datetime
from datetime import timedelta
import time
```

In [2]:
```
df = pd.read_csv('./marathon_results_2016.csv', index_col='Bib',
                 usecols=['Bib','40K','Half','Pace','Age','M/F',
                          'Country','State','City'])

df = df[(df['40K'] != '-') & (df['Half'] != '-')]

df['40K'] = df['40K'].apply(pd.to_timedelta)
df['Half'] = df['Half'].apply(pd.to_timedelta)

df['TotalSeconds'] = df['40K'].apply(lambda x: timedelta.total_seconds(x))
df['HalfSeconds'] = df['Half'].apply(lambda x: timedelta.total_seconds(x))

df.head()
```

Out[2]:

Bib	Age	M/F	City	State	Country	Half	40K	Pace	TotalSeconds	HalfSeconds
5	21	M	Addis Ababa	NaN	ETH	01:06:45	02:05:59	0:05:04	7559.0	4005.0
1	26	M	Ambo	NaN	ETH	01:06:46	02:05:59	0:05:06	7559.0	4006.0
6	31	M	Addis Ababa	NaN	ETH	01:06:44	02:06:47	0:05:07	7607.0	4004.0
11	33	M	Kitale	NaN	KEN	01:06:46	02:06:47	0:05:07	7607.0	4006.0
14	23	M	Eldoret	NaN	KEN	01:06:46	02:08:11	0:05:11	7691.0	4006.0

In [3]:
```
country_city = df.groupby(["Country","City"])
```

In [4]:
```
country_city.size().head(10)
```

Out[4]:
```
Country  City
ALB      Tirana                             1
AND      La Massana                         1
ARG      Avellaneda                         1
         Bernal                             1
         Buenos Aires                      10
         Caba                               1
         Ciudad Autonoma De Buenos Aires    2
         Ciudad De Buenos Aires             2
         Cordoba                            2
         Gonnet                             1
dtype: int64
```

In [5]:
```
country_city.mean().head(10)
```

Out[5]:

		Age	TotalSeconds	HalfSeconds
Country	**City**			
ALB	**Tirana**	48.0	12730.0	6133.0
AND	**La Massana**	47.0	11306.0	5684.0
ARG	**Avellaneda**	41.0	13223.0	6776.0
	Bernal	27.0	10844.0	5454.0
	Buenos Aires	47.1	12801.4	6386.9
	Caba	47.0	15280.0	6831.0
	Ciudad Autonoma De Buenos Aires	44.5	12973.5	6343.5
	Ciudad De Buenos Aires	46.0	13798.5	6747.0
	Cordoba	25.5	12098.5	6056.0
	Gonnet	53.0	11953.0	6222.0

In [6]:
```
country_city.get_group(('USA','San Francisco')).head()
```

Out[6]:

Bib	Age	M/F	City	State	Country	Half	40K	Pace	TotalSeconds	HalfSeconds
796	28	M	San Francisco	CA	USA	01:18:12	02:29:21	0:06:02	8961.0	4692.0
36	30	M	San Francisco	CA	USA	01:14:21	02:30:00	0:06:05	9000.0	4461.0
1267	27	M	San Francisco	CA	USA	01:19:24	02:33:24	0:06:11	9204.0	4764.0
644	51	M	San Francisco	CA	USA	01:19:29	02:37:08	0:06:22	9428.0	4769.0
827	33	M	San Francisco	CA	USA	01:18:52	02:38:13	0:06:24	9493.0	4732.0

In [7]:
```
sex_age = df.groupby(["M/F","Age"])
```

In [8]:
```
sex_age.size().head()
```

Out[8]:
```
M/F  Age
F    18       9
     19      24
     20      44
     21      73
     22     133
dtype: int64
```

In [9]:
```
sex_age.mean().head()
```

Out[9]:

M/F	Age	TotalSeconds	HalfSeconds
F	**18**	16050.666667	7999.888889
	19	15351.958333	7473.666667
	20	13835.931818	6871.045455
	21	14717.630137	7246.479452
	22	14366.421053	7091.097744

In [10]:
```
sex_age.get_group(("M",25)).head()
```

Out[10]:

Bib	Age	M/F	City	State	Country	Half	40K	Pace	TotalSeconds	HalfSeconds
271	25	M	Hull	MA	USA	01:12:57	02:23:40	0:05:48	8620.0	4377.0
200	25	M	Colorado Springs	CO	USA	01:15:01	02:25:27	0:05:51	8727.0	4501.0
124	25	M	Auckland	NaN	NZL	01:12:31	02:24:55	0:05:53	8695.0	4351.0
452	25	M	Philadelphia	PA	USA	01:14:31	02:27:25	0:05:58	8845.0	4471.0
216	25	M	Chicago	IL	USA	01:17:30	02:29:15	0:06:01	8955.0	4650.0

In [ ]: