Python - Analiza danych z modułem PANDAS

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

LAB - S06-L005 - metoda agg

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 **"M/F","Age"** i nazwij go **sex_age**
4. Korzystając z metody **agg()** wyznacz w jednej instrukcji wartość średnią dla kolumn **TotalSeconds** i **HalfSeconds**
5. W ten sam sposób wyznacz sumę dla tych kolumn
6. Korzystając z metody **agg()** wyznacz w jednej instrukcji jednocześnie wartość dla funkcji **mean** i **sum** wyznaczonej dla kolumn **TotalSeconds** i **HalfSeconds**
7. Zdefiniuj listę zawierającą napisy: **'mean','sum','count'** i zapisz ją w zmiennej **functions**
8. Korzystając z metody **agg()** oraz zdefiniowanej listy **funtions** wyznacz w jednej instrukcji średnią, sumę i ilość wartości w kolumnach **TotalSeconds** i **HalfSeconds**

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',

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

In [4]:
```
sex_age.agg({'TotalSeconds'      : 'mean',
             'HalfSeconds'       : 'mean'}).head()
```

Out[4]:

		TotalSeconds	HalfSeconds
M/F	Age		
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 [5]:
```
sex_age.agg({'TotalSeconds'      : 'sum',
             'HalfSeconds'       : 'sum'}).head()
```

Out[5]:

		TotalSeconds	HalfSeconds
M/F	Age		
F	18	144456.0	71999.0
	19	368447.0	179368.0
	20	608781.0	302326.0
	21	1074387.0	528993.0
	22	1910734.0	943116.0

In [6]:
```
sex_age.agg({'TotalSeconds'      : ['mean','sum'],
             'HalfSeconds'       : ['mean','sum','count']}).head()
```

Out[6]:

		TotalSeconds		HalfSeconds		
		mean	sum	mean	sum	count
M/F	Age					
F	18	16050.666667	144456.0	7999.888889	71999.0	9
	19	15351.958333	368447.0	7473.666667	179368.0	24
	20	13835.931818	608781.0	6871.045455	302326.0	44
	21	14717.630137	1074387.0	7246.479452	528993.0	73
	22	14366.421053	1910734.0	7091.097744	943116.0	133

In [7]:
```
functions = ['mean','sum','count']
```

In [8]:
```
sex_age.agg({'TotalSeconds'      : functions,
             'HalfSeconds'       : functions}).head()
```

Out[8]:

		TotalSeconds			HalfSeconds		
		mean	sum	count	mean	sum	count
M/F	Age						
F	18	16050.666667	144456.0	9	7999.888889	71999.0	9
	19	15351.958333	368447.0	24	7473.666667	179368.0	24
	20	13835.931818	608781.0	44	6871.045455	302326.0	44
	21	14717.630137	1074387.0	73	7246.479452	528993.0	73
	22	14366.421053	1910734.0	133	7091.097744	943116.0	133

In [ ]:
```
```