Python - Analiza danych z modułem PANDAS

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

LAB - S06-L003 - agregacje

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. W zmiennej **group_city** zapisz wynik grupowania data frame **df** ze względu na kolumnę **City**
4. Korzystając z odpowiedniej funkcji agregującej wyznacz wartość średnią dla każdej grupy
5. Ponownie wyznacz wartość średnią, ale tym razem jawnie wskaż, że średnia ma być wyliczana tylko dla kolumn **"TotalSeconds","HalfSeconds"**
6. Ile czasu w sumie biegali uczestnicy maratonów w poszczególnych miastach? Korzystając z funkcji sum wyznacz sumę czasu **TotalSeconds** dla każdego miasta oddzielnie
7. W zmiennej **group_age** zapisz wynik grupowania data frame **df** ze względu na kolumnę **Age**
8. Zastosuj agregację count() dla każdej grupy w **group_age**
9. A ilu uczestników w różnym wieku zakończyło bieg? W tym celu policz ile osób w każdym wieku ma podaną wartość w kolumnie **TotalSeconds**

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]:
```
group_city = df.groupby(by="City")
```

In [4]:
```
group_city.mean().head()
```

Out[4]:

City	Age	TotalSeconds	HalfSeconds
0851 Oslo	39.0	11724.0	5731.0
20832	35.0	11640.0	5757.0
34-120 Andrychow	43.5	14111.0	6699.5
5700 Svendborg	58.0	13498.0	6552.0
95630	46.0	13043.0	6625.0

In [5]:
```
group_city[["TotalSeconds","HalfSeconds"]].mean().head()
```

Out[5]:

City	TotalSeconds	HalfSeconds
0851 Oslo	11724.0	5731.0
20832	11640.0	5757.0
34-120 Andrychow	14111.0	6699.5
5700 Svendborg	13498.0	6552.0
95630	13043.0	6625.0

In [6]:
```
group_city["TotalSeconds"].sum().head()
```

Out[6]:
```
City
0851 Oslo           11724.0
20832               11640.0
34-120 Andrychow    28222.0
5700 Svendborg      13498.0
95630               13043.0
Name: TotalSeconds, dtype: float64
```

In [7]:
```
group_age = df.groupby('Age')
```

In [8]:
```
group_age.count().head()
```

Out[8]:

Age	M/F	City	State	Country	Half	40K	Pace	TotalSeconds	HalfSeconds
18	22	22	22	22	22	22	22	22	22
19	41	41	41	41	41	41	41	41	41
20	84	84	82	84	84	84	84	84	84
21	159	159	155	159	159	159	159	159	159
22	226	226	222	226	226	226	226	226	226

In [9]:
```
group_age["TotalSeconds"].count().head()
```

Out[9]:
```
Age
18     22
19     41
20     84
21    159
22    226
Name: TotalSeconds, dtype: int64
```