

# Python - Analiza danych z modulem PANDAS

[www.udemy.com](http://www.udemy.com) (<http://www.udemy.com>) (R)

## LAB - S06-L002 - metoda groupby()

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

df.head()
```

3. W zmiennej **group\_city** zapisz wynik grupowania data frame **df** ze względu na kolumnę **City**
4. Korzystając z odpowiedniego atrybutu zmiennej **group\_city** wyświetl informacje o grupach
5. Wyświetl nagłówki danych z grupy **"San Francisco"**
6. Korzystając ze znanych Ci metod wyznaczania wartości średniej dla kolumny w DataFrame wyznacz średni czas biegu (kolumna **40K**) dla biegaczy z **San Francisco**
7. Wykonaj podobne obliczenia dla biegaczy z **"34-120 Andrychow"**
8. Wyświetl pierwszy wiersz z każdej grupy
9. W zmiennej **group\_age** zapisz wynik grupowania data frame **df** ze względu na kolumnę **Age**
10. Wyznacz średni czas biegu na 40km (kolumna **40K**) dla 20- i 40-latków
11. Wyświetl pierwszy wiersz z każdej grupy

Dane pochodzą z <https://github.com/llimllib/bostonmarathon>

(<https://github.com/llimllib/bostonmarathon>) <https://www.kaggle.com/rojour/boston-marathon-2016->

## 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]:

|     | Age | M/F | City        | State | Country | Half     | 40K      | Pace    | TotalSeconds | HalfSeconds |
|-----|-----|-----|-------------|-------|---------|----------|----------|---------|--------------|-------------|
| Bib |     |     |             |       |         |          |          |         |              |             |
| 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.groups
```

```
'Ada': Index(['5282', '16778', '22181'], dtype='object', name='Bib'),
'Adams Basin': Index(['6287'], dtype='object', name='Bib'),
'Addis Ababa': Index(['5', '6', '7', '4', 'F14'], dtype='object', name='Bib'),
'Adel': Index(['4332'], dtype='object', name='Bib'),
'Adelaide': Index(['15426', '10168', '20505'], dtype='object', name='Bib'),
'Adelaide, South Australia': Index(['19362'], dtype='object', name='Bib'),
'Adelphi': Index(['10244'], dtype='object', name='Bib'),
'Adliswil': Index(['1699', '11601'], dtype='object', name='Bib'),
'Adolfo Lopez Edo Mex': Index(['18673'], dtype='object', name='Bib'),
'Adrian': Index(['6219', '22586', '21429', '23263', '18382'], dtype='object', name='Bib'),
'Adstock': Index(['14927'], dtype='object', name='Bib'),
'Aelvsjoe': Index(['12066'], dtype='object', name='Bib'),
'Afton': Index(['1010', '19079'], dtype='object', name='Bib'),
'Agassiz': Index(['10343'], dtype='object', name='Bib'),
'Agawam': Index(['6193'], dtype='object', name='Bib'),
'Ageo': Index(['21558'], dtype='object', name='Bib'),
'Agoura Hills': Index(['24813', '29867'], dtype='object', name='Bib'),
...
```

```
In [5]: group_city.get_group("San Francisco").head()
```

Out[5]:

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

```
In [6]: group_city.get_group("San Francisco")["40K"].mean()
```

Out[6]: Timedelta('0 days 03:25:35.335025')

```
In [7]: group_city.get_group("34-120 Andrychow")["40K"].mean()
```

Out[7]: Timedelta('0 days 03:55:11')

```
In [8]: group_city.first().head()
```

Out[8]:

|                     | Age | M/F | State | Country | Half     | 40K      | Pace    | TotalSeconds | HalfSeconds |
|---------------------|-----|-----|-------|---------|----------|----------|---------|--------------|-------------|
| City                |     |     |       |         |          |          |         |              |             |
| 0851 Oslo           | 39  | F   | NaN   | NOR     | 01:35:31 | 03:15:24 | 0:07:55 | 11724.0      | 5731.0      |
| 20832               | 35  | M   | MD    | USA     | 01:35:57 | 03:14:00 | 0:07:51 | 11640.0      | 5757.0      |
| 34-120<br>Andrychow | 44  | F   | NaN   | POL     | 01:58:39 | 03:55:09 | 0:09:28 | 14109.0      | 7119.0      |
| 5700<br>Svendborg   | 58  | M   | NaN   | DEN     | 01:49:12 | 03:44:58 | 0:09:05 | 13498.0      | 6552.0      |
| 95630               | 46  | F   | CA    | USA     | 01:50:25 | 03:37:23 | 0:08:44 | 13043.0      | 6625.0      |

```
In [9]: group_age = df.groupby('Age')
```

```
In [10]: group_age.get_group(20)["40K"].mean()
```

Out[10]: Timedelta('0 days 03:36:54.583333')

```
In [11]: group_age.get_group(40)["40K"].mean()
```

Out[11]: Timedelta('0 days 03:35:22.724557')

```
In [12]: group_age.first().head()
```

Out[12]:

|     | M/F | City        | State | Country | Half     | 40K      | Pace    | TotalSeconds | HalfSeconds |
|-----|-----|-------------|-------|---------|----------|----------|---------|--------------|-------------|
| Age |     |             |       |         |          |          |         |              |             |
| 18  | M   | Needham     | MA    | USA     | 01:22:35 | 02:38:30 | 0:06:23 | 9510.0       | 4955.0      |
| 19  | M   | Plainfield  | IL    | USA     | 01:25:28 | 02:42:30 | 0:06:33 | 9750.0       | 5128.0      |
| 20  | M   | Addis Ababa | MI    | ETH     | 01:06:45 | 02:14:23 | 0:05:26 | 8063.0       | 4005.0      |
| 21  | M   | Addis Ababa | MA    | ETH     | 01:06:45 | 02:05:59 | 0:05:04 | 7559.0       | 4005.0      |
| 22  | M   | Eugene      | OR    | USA     | 01:13:53 | 02:24:35 | 0:05:52 | 8675.0       | 4433.0      |

```
In [ ]:
```

