

Python - Analiza danych z modulem PANDAS

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

LAB - S03-L009-LAB Ranking

1. Zaimportuj moduł pandas oraz numpy i nadaj im standardowe aliasy. Do zmiennej **fortune** wczytaj zawartość pliku **Fortune_500_2017.csv**. Pobierz tylko następujące kolumny: **'Rank', 'Title', 'Employees', 'Profits', 'Assets'**. Jako indeksu użyj kolumny **'Rank'**. Wyświetl nagłówek tak utworzonego Data Frame
2. Wylicz ranking tak, aby najniższe wartości były przyznawane firmom z największą ilością pracowników (kolumna **Employees**). Wyświetl nagłówek wyznaczonej serii danych
3. Dodaj do zmiennej frame kolumnę nazwaną **'RankByEmployee'**, zawierającą powyżej wyznaczony ranking
4. Wylicz ranking tak, aby najniższe wartości były przyznawane firmom z największym zyskiem (kolumna **Profits**). Wyświetl nagłówek wyznaczonej serii danych
5. Dodaj do zmiennej frame kolumnę nazwaną **'RankByProfits'**, zawierającą powyżej wyznaczony ranking
6. Wyświetl 3 firmy, które mają największy zasób środków trwałych (kolumna **Asset**)
7. Wyświetl 3 firmy, które mają najmniejszy zasób środków trwałych (kolumna **Asset**)

Uwaga - wyżej wymienione czynności przydadzą Ci się podczas wykonywania kolejnego laboratorium, jeśli zapiszesz swoje aktualne rozwiązanie, to w następnym laboratorium czeka Cię mniej pracy ;)

Rozwiązania:

Poniżej znajdują się propozycje rozwiązań zadań. Prawdopodobnie istnieje wiele dobrych rozwiązań, dlatego jeżeli rozwiążesz 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
fortune = pd.read_csv("Fortune_500_2017.csv",
                      usecols=['Rank', 'Title', 'Employees', 'Profits', 'Assets'],
                      index_col = 'Rank')
fortune.head()
```

Out[1]:

	Title	Employees	Profits	Assets
Rank				
1	Walmart	2300000	13643.0	198825
2	Berkshire Hathaway	367700	24074.0	620854
3	Apple	116000	45687.0	321686
4	Exxon Mobil	72700	7840.0	330314
5	McKesson	68000	2258.0	56563

```
In [2]: fortune["Employees"].rank(ascending=False).head()
```

```
Out[2]: Rank
1      1.0
2      7.0
3     57.0
4     94.0
5    104.5
Name: Employees, dtype: float64
```

```
In [3]: fortune['RankByEmployee'] = fortune["Employees"].rank(ascending=False)
fortune.head()
```

```
Out[3]:
```

	Title	Employees	Profits	Assets	RankByEmployee
Rank					
1	Walmart	2300000	13643.0	198825	1.0
2	Berkshire Hathaway	367700	24074.0	620854	7.0
3	Apple	116000	45687.0	321686	57.0
4	Exxon Mobil	72700	7840.0	330314	94.0
5	McKesson	68000	2258.0	56563	104.5

```
In [4]: fortune["Profits"].rank(ascending=False).head()
```

```
Out[4]: Rank
1      11.0
2       3.0
3       1.0
4      27.0
5     102.0
Name: Profits, dtype: float64
```

```
In [5]: fortune['RankByProfits'] = fortune["Profits"].rank(ascending=False)
fortune.head()
```

```
Out[5]:
```

	Title	Employees	Profits	Assets	RankByEmployee	RankByProfits
Rank						
1	Walmart	2300000	13643.0	198825	1.0	11.0
2	Berkshire Hathaway	367700	24074.0	620854	7.0	3.0
3	Apple	116000	45687.0	321686	57.0	1.0
4	Exxon Mobil	72700	7840.0	330314	94.0	27.0
5	McKesson	68000	2258.0	56563	104.5	102.0

```
In [6]: fortune.nlargest(n=3, columns='Assets')
```

```
Out[6]:
```

	Title	Employees	Profits	Assets	RankByEmployee	RankByProfits
Rank						
20	Fannie Mae	7000	12313.0	3287968	446.5	15.0
21	J.P. Morgan Chase	243355	24733.0	2490972	20.0	2.0
26	Bank of America Corp.	208024	17906.0	2187702	29.0	6.0

```
In [7]: fortune.nsmallest(n=3, columns='Assets')
```

```
Out[7]:
```

	Title	Employees	Profits	Assets	RankByEmployee	RankByProfits
Rank						
395	A-Mark Precious Metals	83	9.3	437	500.0	434.0
247	Core-Mark Holding	7688	54.2	1497	434.0	430.0
495	Big Lots	23150	152.8	1608	266.0	403.0

```
In [ ]:
```