# Python - Analiza danych z modułem PANDAS

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

# LAB - S03-L010-LAB Filtrowanie Data Frame

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. **Punkty 2 - 5 wykonywałeś już w poprazednim laboratorium. Możesz śmiało skorzystać z poprzednich rozwiązań, ale jak masz ochotę - możesz jeszcze raz przypomnieć sobie zasady budowy rankingów :) Jeżeli korzystasz z gotowego rozwiązania przeskocz do punktu 6.**
2. Wylicz ranking tak, aby najniższe wartości były przyznawane firmom z największą ilośćią 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. Utwórz zmienną **isEmployeesRankLess10**, która ma przechowywać serię wartości True/False odpowiadających na pytanie czy firma znajduje się w pierwszej dzisiątce w rankingu firm o największym zatrudnieniu. (True - firma jest w pierwszej dziesiątce, False - jest na dalszej pozycji)
7. Utwórz zmienną **isEmployeesRankFirst10**, która ma przechowywać serię wartości True/False odpowiadających na pytanie czy firma znajduje się w pierwszej dzisiątce w rankingu firm o największym zatrudnieniu. (True - firma jest w pierwszej dziesiątce, False - jest na dalszej pozycji)
8. Utwórz zmienną **isProfitRankFirst10****, która ma przechowywać serię wartości True/False odpowiadających na pytanie czy firma znajduje się w pierwszej dzisiątce w rankingu firm o największym zysku. (True - firma jest w pierwszej dziesiątce, False - jest na dalszej pozycji)
9. Korzystając ze zmiennych z poprzednch punktów, wyświetl firmy, które jednocześnie są w pierwszej dziesiatce pod względem ilości pracowników i wielkości zysku. (tutaj możesz próbować odpowiedzieć na pytanie czy zysk firmy jest powiązany z ilością pracowników)
10. Utwórz zmienną **isEmployeesRankMore400**, która ma przechowywać serię wartości True/False odpowiadających na pytanie czy firma znajduje się w rankingu firm o największym zatrudnieniu na pozycji 400 lub dalej. (True - firma jest w ostatniej setce, False - jest na dalszej pozycji)
11. Korzystając ze zmiennych z poprzednch punktów, wyświetl firmy, które jednocześnie są w ostatniej setce firm pod względem ilości pracowników i pierwszej dzisiątce wielkości zysku. (tutaj możesz próbować odpowiedzieć na pytanie jakie firmy mają zysk wypracowany przez małą liczbę pracowników)

# 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
fortune = pd.read_csv("Fortune_500_2017.csv",
                      usecols=['Rank','Title','Employees','Profits','Assets'],
                      index_col = 'Rank')
fortune.head()
```

Out[1]:

| Rank | Title | Employees | Profits | Assets |
|---|---|---|---|---|
| 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]:

| Rank | Title | Employees | Profits | Assets | RankByEmployee |
|---|---|---|---|---|---|
| 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]:

| Rank | Title | Employees | Profits | Assets | RankByEmployee | RankByProfits |
|---|---|---|---|---|---|---|
| 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]:
```
isEmployeesRankFirst10 = fortune.RankByEmployee <= 10
```

In [7]:
```
isProfitRankFirst10 = fortune.RankByProfits <= 10
```

In [8]:
```
fortune[isEmployeesRankFirst10 & isProfitRankFirst10]
```

Out[8]:

| Rank | Title | Employees | Profits | Assets | RankByEmployee | RankByProfits |
|---|---|---|---|---|---|---|
| 2 | Berkshire Hathaway | 367700 | 24074.0 | 620854 | 7.0 | 3.0 |

In [9]:
```
isEmployeesRankMore400 = fortune.RankByEmployee >=400
```

In [10]:
```
fortune[isEmployeesRankMore400 & isProfitRankFirst10]
```

Out[10]:

| Rank | Title | Employees | Profits | Assets | RankByEmployee | RankByProfits |
|---|---|---|---|---|---|---|
| 148 | Altria Group | 8300 | 14239.0 | 45932 | 424.0 | 10.0 |

In [11]:
```
fortune.head()
```

Out[11]:

| Rank | Title | Employees | Profits | Assets | RankByEmployee | RankByProfits |
|---|---|---|---|---|---|---|
| 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 [ ]: