

Python - Analiza danych z modulem PANDAS

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

LAB - S02-L017 - Modyfikacja serii danych

1. Zaimportuj moduł **pandas** i nadaj mu standardowy alias. Do zmiennej **surveys** zapisz data series pobierając wartości z pliku **StackOverflowDeveloperSurvey.csv** kolumnę **Salary**. Ponieważ nie wszyscy ankietowani podali wysokość swojej pensji, usuń te wpisy, które są puste (skorzystaj z metody **dropna()** bez żadnych dodatkowych argumentów). Wyświetl pięć pierwszych pozycji tej serii.
2. Interesuje Cię ile osób podało wysokość swojej wypłaty? Wyświetl informację od ilości elementów w serii **surveys**
3. Symulujemy podwyżkę pensji o **3%**. Utwórz nową serię **surveysIncrease**, której wartością będą kwoty z serii **surveys** pomnożone przez **0.03**. Wyświetl nagłówek nowej serii.
4. Utwórz nową serię **surveysAfterIncrease**, której wartością będzie suma **surveys** i **surveysIncrease**. Wyświetl nagłówek.
5. Zmieniamy trochę temat. Do zmiennej **surveysTime** wczytaj kolumnę **HoursOutside** z pliku **StackOverflowDeveloperSurvey2018.csv**. Ponieważ plik jest duży, dodaj parametr **low_memory=False**. Również skorzystaj z metody **dropna()**, aby wyeliminować puste wartości. Wyświetl nagłówek.
6. Interesuje Cię ile czasu programiści spędzają na "świeżym powietrzu"? Uruchom na rzecz **surveysTime** metodę, która dla każdej unikalnej pozycji z serii wyświetli ile razy ta pozycja występowała w serii. Do której grupy należysz ;) ?
7. Po zaimportowaniu danych, można je normalizować. Zmień wielkość liter na małe w całej serii **surveysTime**. Zmiany mają rzeczywiście być zapisane w **surveysTime**. Wyświetl nagłówek.
8. Zmieniamy zdanie. Chcemy, aby teksty były zapisane wielkimi literami. Tym razem wykorzystaj do tego wyrażenie lambda. Zmiany znowu mają być zapisane do **surveysTime**. Wyświetl nagłówek.
9. Napisz funkcję **ChangeDescription**, która jako argument przyjmie napis. Jeżeli w tej zmiennej znajduje się tekst **'LESS THAN 30 MINUTES'** to zwróć **'LESS THAN HALF HOUR'**, a w przeciwnym razie zwróć oryginalną wartość przekazaną w argumencie.
10. Przetestuj funkcję przekazując do niej różne napisy.
11. Zastosuj funkcję **ChangeDescription** do serii **surveysTime**. Wyświetl nagłówek
12. Wyciągnij wniosek na temat "wychodzenia na świeże powietrze"

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
surveys = pd.read_csv("StackOverflowDeveloperSurvey.csv", usecols=["Salary"])
surveys.head()
```

```
Out[1]: 2      113750.0
      14      100000.0
      17      130000.0
      18       82500.0
      22      100764.0
      Name: Salary, dtype: float64
```

```
In [2]: len(surveys)
```

```
Out[2]: 12891
```

```
In [3]: surveysIncrease = surveys * 0.03
surveysIncrease.head()
```

```
Out[3]: 2      3412.50
      14      3000.00
      17      3900.00
      18      2475.00
      22      3022.92
      Name: Salary, dtype: float64
```

```
In [4]: surveysAfterIncrease = surveys + surveysIncrease
surveysAfterIncrease.head()
```

```
Out[4]: 2      117162.50
      14      103000.00
      17      133900.00
      18      84975.00
      22      103786.92
      Name: Salary, dtype: float64
```

```
In [5]: surveysTime = pd.read_csv("StackOverflowDeveloperSurvey2018.csv",
                                usecols=["HoursOutside"], squeeze=True, low_memory_increased=True)
surveysTime.head()
```

```
Out[5]: 0      1 - 2 hours
      1      30 - 59 minutes
      3  Less than 30 minutes
      4      1 - 2 hours
      5      30 - 59 minutes
      Name: HoursOutside, dtype: object
```

```
In [6]: surveysTime.value_counts()
```

```
Out[6]: 1 - 2 hours      27788
      30 - 59 minutes    24002
      Less than 30 minutes 11223
      3 - 4 hours        7186
      Over 4 hours       1825
      Name: HoursOutside, dtype: int64
```

```
In [7]: surveysTime = surveysTime.str.lower()
surveysTime.head()
```

```
Out[7]: 0          1 - 2 hours
1          30 - 59 minutes
3    less than 30 minutes
4          1 - 2 hours
5          30 - 59 minutes
Name: HoursOutside, dtype: object
```

```
In [8]: surveysTime = surveysTime.apply(lambda desc: desc.upper())
surveysTime.head()
```

```
Out[8]: 0          1 - 2 HOURS
1          30 - 59 MINUTES
3    LESS THAN 30 MINUTES
4          1 - 2 HOURS
5          30 - 59 MINUTES
Name: HoursOutside, dtype: object
```

```
In [9]: def ChangeDescription(desc):
        if desc == 'LESS THAN 30 MINUTES':
            return 'LESS THAN HALF HOUR'
        else:
            return desc
```

```
In [10]: print(ChangeDescription('LESS THAN 30 MINUTES'))
print(ChangeDescription('1 HOUR'))
```

```
LESS THAN HALF HOUR
1 HOUR
```

```
In [11]: surveysTime = surveysTime.apply(ChangeDescription)
surveysTime.head()
```

```
Out[11]: 0          1 - 2 HOURS
1          30 - 59 MINUTES
3    LESS THAN HALF HOUR
4          1 - 2 HOURS
5          30 - 59 MINUTES
Name: HoursOutside, dtype: object
```

;) świeże powietrze musi być bardzo szkodliwe, skoro tak mała liczba programistów, którzy wyszli na więcej niż 4 godziny przeżyła....

```
In [ ]:
```