Python - Analiza danych z modułem PANDAS

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

LAB - S02-L012 - Pobieranie wartości po indeksie

1. Zaimportuj moduł pandas i nadaj mu standardowy alias. Do zmiennej surveys zapisz data series pobierając wartości z pliku StackOverflowDeveloperSurvey.csv kolumnę CompanySize. Wyświetl pięć pierwszych pozycji tej serii.
2. Wyświetl wartość z pozycji 3 (indeks 3)
3. Wyświetl elementy z pozycji 1-10 włącznie. Czy wartość na trzeciej pozycji jest zgodna z wynikiem z poprzedniego punktu?
4. Wyświetl wartość z pozycji 12345 (indeks 12345)
5. Wyświetl elementy z pozycji 12341 - 12350 włącznie. Czy wartość na pozycji 12345 jest zgodna z wynikiem z poprzedniego punktu?
6. Posortuj serię surveys korzystając z parametru inplace=True
7. Wyświetl wartość z pozycji 3 (indeks 3), czy to nadal ta sama wartość co poprzednio?
8. Wyświetl elementy z pozycji 1-10 włącznie. Czy wartość na trzeciej pozycji jest zgodna z wynikiem z poprzedniego punktu? Czy ten wynik nie jest dziwny??
9. Wyświetl wartość z pozycji 12345 (indeks 12345), czy to nadal ta sama wartość co poprzednio?
10. Wyświetl elementy z pozycji 12341 - 12350 włącznie. Czy wartość na trzeciej pozycji jest zgodna z wynikiem z poprzedniego punktu? Czy ten wynik nie jest dziwny??
11. Wykonaj polecenie resetujące indeks (będzie o nim mowa w dalszej części kursu, na razie weź to "na wiarę"): surveys.reset_index(drop=True,inplace=True)
12. Wykonaj ponownie polecenia z pkt 7 - 10. Czy teraz wyniki bardziej pasują?

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=["CompanySize"]).squeeze().dropna()
surveys.head()
```

Out[1]:
```
1          20 to 99 employees
2    10,000 or more employees
3    10,000 or more employees
4          10 to 19 employees
6          20 to 99 employees
Name: CompanySize, dtype: object
```

In [2]:
```
surveys[3]
```

Out[2]:
```
'10,000 or more employees'
```

In [3]:
```
surveys[1:11]
```

Out[3]:
```
2     10,000 or more employees
3     10,000 or more employees
4           10 to 19 employees
6           20 to 99 employees
7      Fewer than 10 employees
8     5,000 to 9,999 employees
10        100 to 499 employees
11        100 to 499 employees
13     Fewer than 10 employees
14    5,000 to 9,999 employees
Name: CompanySize, dtype: object
```

In [4]:
```
surveys[12345]
```

Out[4]:
```
'20 to 99 employees'
```

In [5]:
```
surveys[12341:12351]
```

Out[5]:
```
15989    1,000 to 4,999 employees
15991          20 to 99 employees
15992          10 to 19 employees
15993    1,000 to 4,999 employees
15994          20 to 99 employees
15995        100 to 499 employees
15996          10 to 19 employees
15997                I don't know
15998          20 to 99 employees
15999        100 to 499 employees
Name: CompanySize, dtype: object
```

In [6]:
```
surveys.sort_values(inplace=True)
```

In [7]:
```
surveys[3]
# tak to nadal ta sama wartość co poprzednio
```

Out[7]:
```
'10,000 or more employees'
```

In [8]:
```
surveys[12345]
# tak to nadal ta sama wartość co poprzednio
```

Out[8]:
```
'20 to 99 employees'
```

In [9]:
```
surveys[1:11]
# hmmm dziwne - wartość na pozycji 3 jest inna niż tutaj!
```

Out[9]:
```
36683    1,000 to 4,999 employees
43105    1,000 to 4,999 employees
43102    1,000 to 4,999 employees
30582    1,000 to 4,999 employees
43072    1,000 to 4,999 employees
43063    1,000 to 4,999 employees
23955    1,000 to 4,999 employees
43060    1,000 to 4,999 employees
43056    1,000 to 4,999 employees
30558    1,000 to 4,999 employees
Name: CompanySize, dtype: object
```

In [10]:
```
surveys[12341:12351]
# hmmm dziwne - wartość na pozycji 12345 jest inna niż tutaj!
```

Out[10]:
```
14554    10,000 or more employees
9429     10,000 or more employees
6384     10,000 or more employees
1102     10,000 or more employees
14549    10,000 or more employees
14547    10,000 or more employees
14570    10,000 or more employees
14575    10,000 or more employees
6410     10,000 or more employees
14576    10,000 or more employees
Name: CompanySize, dtype: object
```

In [11]:
```
surveys[[3,12345]]
# a tutaj nadal jest ta sama wartość co poprzednio
```

Out[11]:
```
3        10,000 or more employees
12345          20 to 99 employees
Name: CompanySize, dtype: object
```

In [12]:
```
surveys.reset_index(drop=True,inplace=True)
```

In [13]:
```
surveys[3]
# teraz na pozycji 3 jest inna wartość niż oryginalnie - sensowne
```

Out[13]:
```
'1,000 to 4,999 employees'
```

In [14]:
```
surveys[1:11]
# i wartość na pozycji 3 jest taka sama jak tutaj -
# teraz wszystko pasuje!
```

Out[14]:
```
1     1,000 to 4,999 employees
2     1,000 to 4,999 employees
3     1,000 to 4,999 employees
4     1,000 to 4,999 employees
5     1,000 to 4,999 employees
6     1,000 to 4,999 employees
7     1,000 to 4,999 employees
8     1,000 to 4,999 employees
9     1,000 to 4,999 employees
10    1,000 to 4,999 employees
Name: CompanySize, dtype: object
```

In [15]:
```
surveys[12345]
# teraz na pozycji 12345 jest inna wartość niż
# oryginalnie - sensowne
```

Out[15]:
```
'10,000 or more employees'
```

In [16]:
```
surveys[12341:12351]
# i wartość na pozycji 3 jest taka sama jak tutaj -
# teraz wszystko pasuje!
```

Out[16]:
```
12341    10,000 or more employees
12342    10,000 or more employees
12343    10,000 or more employees
12344    10,000 or more employees
12345    10,000 or more employees
12346    10,000 or more employees
12347    10,000 or more employees
12348    10,000 or more employees
12349    10,000 or more employees
12350    10,000 or more employees
Name: CompanySize, dtype: object
```

No cóż, w tym zadaniu zobaczyć można pewien niuans. Sortowanie zmieniło kolejność elementów, ale nie przebudowało indeksu. Część poleceń pobiera wartości dokładnie w oparciu o indeks, a inne bazują poprostu na kolejności elementów. To dlatego zwracane wyniki były pozornie sprzeczne. Wystarczyło jednak przebudować indeks i wszystko zaczęło działać!

In [ ]: