

pandas_intro

August 23, 2017

0.1 Pandas

Pandas je modul namenjen brzom i efikasnom radu sa podacima.

```
In [8]: import pandas as pd
```

Za predstavljanje podataka se koriste dve strukture: **Series** i **DataFrame**

Series su strukture koje odgovaraju jednodimenzionim nizovima (čiji elementi mogu biti raznorodni) i koje imaju pridruženi niz obeležja tzv. indeks.

DataFrame su strukture koje odgovaraju tabelarnim podacima (eng. spreadsheet-like data structures) i koje imaju indekse i vrsta i kolona.

NaN označava nedostajuću vrednost. Sa **isnull** i **notnull** se može proveriti da li je neka vrednost nedostajuća ili ne.

```
In [2]: # stampa
        #\?pd.Series
```

```
In [9]: # kreiranje serije
        points = pd.Series([(1, 1) , (2, 2), (3, 3)])
        print(type(points[0]))
        print(points)
        import numpy as np
        points1 = np.array([(1, 1) , (2, 2), (3, 3)])
        print(type(points1[1]))
```

```
<class 'tuple'>
0      (1, 1)
1      (2, 2)
2      (3, 3)
dtype: object
<class 'numpy.ndarray'>
```

```
In [10]: # ocitavanje elemenata serije
         #print(points.values)
         print(points.keys)
```

```

<bound method Series.key of 0      (1, 1)
1      (2, 2)
2      (3, 3)
dtype: object>

```

```

In [11]: # ocitavanje indeksa serije
points.index
#points.index.get_values()[0]

```

```

Out[11]: RangeIndex(start=0, stop=3, step=1)

```

```

In [12]: # dodeljivanje indeksa seriji
points = pd.Series([(1, 1), (2, 2), (3, 3)], index = ['x', 'y', 'z'])
points

```

```

Out[12]: x      (1, 1)
         y      (2, 2)
         z      (3, 3)
dtype: object

```

```

In [13]: points.index

```

```

Out[13]: Index(['x', 'y', 'z'], dtype='object')

```

```

In [41]: # pristup elementu serije
points['x']

```

```

Out[41]: (1, 1)

```

```

In [14]: # kreiranje okvira podataka
data = {
    'name': ['Anna', 'John', 'Peter', 'Linda'],
    'location': ['Berlin', 'New York', 'Belgrade', 'London'],
    'age': [24, 30, 33, 21]
}
print(type(data))

users = pd.DataFrame(data)

```

```

<class 'dict'>

```

```

In [43]: #from IPython.display import display
# moze se koristiti i display za doterani prikaz podataka
users

```

```

Out[43]:
   age  location  name
0   24    Berlin  Anna
1   30  New York  John
2   33  Belgrade  Peter
3   21    London  Linda

```


	age	location	name	email
0	24.0	Berlin	Anna	NaN
1	30.0	New York	John	NaN
x	NaN	NaN	NaN	NaN
y	NaN	NaN	NaN	NaN
z	NaN	NaN	NaN	NaN
w	NaN	NaN	NaN	NaN

Provera nedostajucih vrednosti je moguca kroz **isnull** ili **notnull**

```
In [80]: pd.isnull(users['email'])
```

```
Out[80]: 0      True
         1      True
         2      True
         3      True
         Name: email, dtype: bool
```

Primer korišćenja raspoloživih podataka

```
In [15]: # učitavanje podataka iz CSV fajla
         # podaci su preuzeti sa https://www.kaggle.com/the-guardian/extinct-languages
         # i predstavljaju informacije o jezicima koji su ugrozeni ili u izumiranju
         # Kategorije koje se razmatraju su:
         # vulnerable - jezik se uci u nekom uzem kontekstu
         # definitely endangered - jezik se ne uci kao maternji jezik
         # severely endangered - jezik govore pripadnici starijih generacija, dok g
         # critically endangered - jezik govore samo pripadnici starijih generacija
         # extinct - jezici ciji govornici vise ne postoje

         languages = pd.read_csv('data/languages.csv')

In [70]: # dodatne informacije o read_csv funkciji
         # stampa \?pd.read_csv

         # na raspolaganju su i funkcije za analiziranje Excel formata, JSON format

In [71]: # rezultat citanje je DataFrame struktura
         type(languages)

Out[71]: pandas.core.frame.DataFrame

In [72]: # ispis broja vrsti ucitanog skupa podataka
         len(languages)

Out[72]: 2722

In [1]: # ispis prvih 5 redova
         # stampa languages.head(5)
```



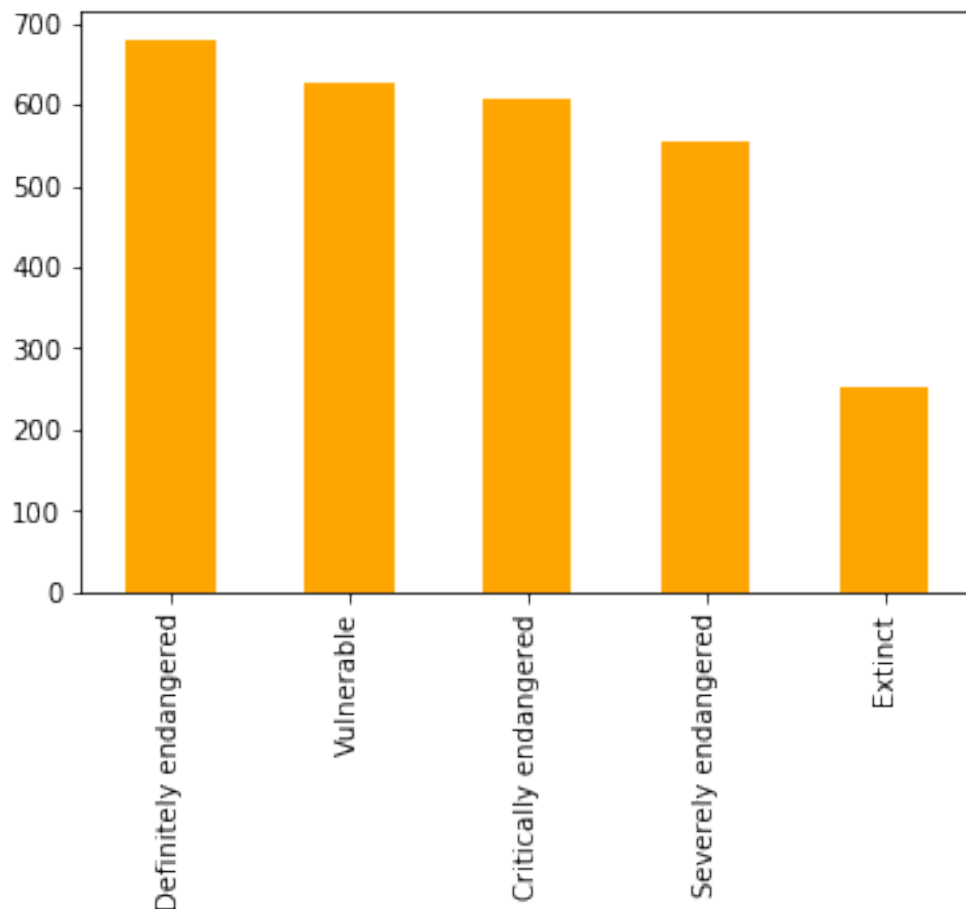
```
In [80]: # value_counts prebrojava razlicite vrednosti u kolonama  
# stampa \?pd.value_counts
```

```
In [81]: # izdvajamo informaciju o broju jezika iz svake kategorije  
# kategorije koje su nam na raspolaganju su sadrzane u koloni Degree of en  
languages['Degree of endangerment'].value_counts()
```

```
Out[81]: Definitelyendangered    680  
Vulnerable                      628  
Criticallyendangered            607  
Severelyendangered              554  
Extinct                         253  
Name: Degree of endangerment, dtype: int64
```

```
In [82]: # iscrtavamo histogram na osnovu izracunatih podataka  
languages['Degree of endangerment'].value_counts().plot(kind='bar', color=
```

```
Out[82]: <matplotlib.axes._subplots.AxesSubplot at 0xc78b48fd30>
```



```
In [84]: # iscrtavamo grafik na osnovu izracunatih podataka
         #languages['Degree of endangerment'].value_counts().plot()
         languages['Degree of endangerment'].value_counts().plot(kind='pie')
```

```
Out[84]: <matplotlib.axes._subplots.AxesSubplot at 0xc78a9f7518>
```

```
In [85]: # izdvajamo informacije samo o jezicima koji su kritično ugroženi
         critically_endangered=languages[languages['Degree of endangerment'] == 'Critically endangered']
```

```
In [6]: # i ispisujemo informacije o prvih 5 najugroženijih jezika
        # stampa critically_endangered[:5]
```

```
In [16]: # postoji i mogućnost vezivanja uslova: koriste se & |
         ce = languages['Degree of endangerment'] == 'Critically endangered'
         #print(ce)
         vu = languages['Degree of endangerment'] == 'Vulnerable'
         endangered_languages = languages[ce | vu]
         #stampa endangered_languages[100:110]
```

```
In [91]: # generisane podatke je moguće izvesti i sacuvati
         endangered_languages.to_csv('./endangered.csv')
```

```
In [92]: # stampa \!less endangered.csv
```

'less' is not recognized as an internal or external command,
operable program or batch file.

**** Dodatni materijali: ****

- <https://github.com/jvns/pandas-cookbook>
- **knjiga:** Python for Data Analysis, Wes McKinney

In []: