

INTELIGENTNI SISTEMI

as. ms Vladimir Jocić
as. ms Adrian Milaković



PANDAS
biblioteka

09

PANDAS BIBLIOTEKA

Šta je pandas biblioteka?

- **Pan(el) da(ta)s** je softverska biblioteka napisana za programski jezik Pajton namenjena analizi podataka.
- Dizajnirana je za manipulaciju nad tabelama podataka (*DataFrame*, 2D) i sekvencama podataka (*Series*, 1D) i podržava:
 - Funkcije za prenos podataka između različitih ekspresivnih struktura podataka biblioteke i velikog broja formata fajlova.
 - Intuitivne operacije za filtriranje, indeksiranje, spajanje, izmenu ugrađenih struktura podataka, oblikovanje, pristup podskupu podataka itd.
 - Mogućnost za primenu različitih operacija nad bibliotečkim strukturama podataka, grupisanje po određenom delu strukture itd.
- Predstavlja nadgradnju na popularnu Python biblioteku, koja pruža podršku za višedimenzione nizove i matematičke funkcije – NumPy.

PANDAS BIBLIOTEKA

Kako preuzeti pandas biblioteku?

- Pandas nije standardna Python biblioteka. Moguće je preuzeti je na više različitih načina:
 - Iz PyCharm Python okruženja: File -> Settings -> Project: *Project Name* -> Python Interpreter, a zatim izabrati dugme **+** za preuzimanje novog paketa. Zatim je potrebno pretražiti **pandas** i instalirati paket na dugme **Install Package**. Na dugme **-** moguće je ukloniti instalirani paket.
 - Iz PyCharm Python Terminala komandom **pip install pandas**.
 - Pisanjem naredbe **import pandas** u fajlu sa Pajton izvornim kodom, a zatim prelaskom mišem preko naredbe (*hover*) iz tooltip prozora izabrati opciju **Install package pandas**.
 - Korišćenjem popularnog package manager-a **Anaconda**.

PODRŽANI TIPOVI PODATAKA

Da bi se omogućio uvoz pandas biblioteke i započeo rad sa funkcionalnostima koje pruža neophodno je napisati sledeću naredbu:

```
import pandas as pd
```

gde `pd` predstavlja konvencijski alias (drugo ime) za ovaj modul/biblioteku.

Ova biblioteka definiše dve najčešće korišćene strukture podataka: `Series` i `DataFrame`.

Series predstavlja jednodimenzioni niz podataka, kojima se pristupa na osnovu labela, koje su definisane osom labela (`index`). Ukoliko želimo da kreiramo objekat ovog tipa neophodno je napisati sledeću naredbu:

```
s = pd.Series(data=[1, 4, 7], index=[0, 1, 2], dtype=int, name='ime')
```

data je obavezan argument i predstavlja podatke serije.

index je opcioni argument i predstavlja labele, na osnovu kojih se pristupa podacima. Mora biti iste dužine kao i argument `data`.

dtype je opcioni argument i predstavlja tip podataka serije. Može se zaključiti iz `data`.

name je opcioni argument i predstavlja ime serije. Može se zaključiti iz `data`.

SERIES - ATRIBUTI

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>s = pd.Series(data=[1, 4, 7], index=[0, 1, 2], dtype=int, name='ime')</pre>	
Indeksi serije	<pre>print(s.index.values) ekv. print(s.keys().values)</pre>	[0 1 2]
Vrednosti serije	<pre>print(s.values)</pre>	[1 4 7]
Tip podataka serije	<pre>print(s.dtype)</pre>	int32
Dimenzije serije (torka)	<pre>print(s.shape)</pre>	(3,)
Broj podataka	<pre>print(s.size)</pre>	3
Da li postoje NaN-ovi	<pre>print(s.hasnans)</pre>	False
Da li je serija prazna	<pre>print(s.empty)</pre>	False
Ime serije	<pre>print(s.name)</pre>	ime

SERIES - KONVERZIJE

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>s = pd.Series(data=[1, 4, 7], index=[0, 1, 2], dtype=int, name='ime')</pre>	
Konverzija podataka u tip uz vraćanje kopije	<pre>sc = s.astype(dtype=float, copy=True)</pre>	
Ispis kopije serije	<pre>print(sc)</pre>	<pre>0 1.0 1 4.0 2 7.0 Name: ime, dtype: float64</pre>
Kreiranje kopije (rekurzivno, duboka)	<pre>sc = s.copy(deep=True)</pre>	
Konverzija u NumPy ndarray	<pre>print(s.to_numpy())</pre>	<pre>[1 4 7]</pre>
Konverzija u listu	<pre>print(s.to_list())</pre>	<pre>[1, 4, 7]</pre>
Konverzija u boolean (samo ako je size == 1)	<pre>print(s.bool())</pre>	<pre>ValueError</pre>

SERIES - INDEKSIRANJE

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>s = pd.Series(data=[1, 4, 7], index=['a', 'b', 'c'], dtype=int, name='ime')</pre>	
Dohvatanje vrednosti	<pre>print(s.get('c'))</pre>	7
Pristup jednoj vrednosti preko ključa (get, set)	<pre>s.at['c'] = 9 s.at['d'] = 11 print(s.at['c'],s.at['d'])</pre>	9 11
Pristup jednoj vrednosti preko indeksa (get, set). Ne sme se vršiti set po indeksu van opsega.	<pre>s.iat[2] = 9 print(s.iat[2]) # ne sme s.iat[3] = 11</pre>	9
Pristup vrednosti preko labela indeksa (get, set)	<pre>s.loc['a'] = 11 print(s.loc['a'])</pre>	11
Pristup vrednosti preko labela indeksa (get, set)	<pre>s.loc[['a', 'b']] = 1, 1 ekv. s.loc[['a', 'b']] = 1 print(s.loc[['a', 'b']])</pre>	a 1 b 1 Name: ime, dtype: int32
Pristup vrednosti preko opsega labela (get, set)	<pre>s.loc['a':'b'] = [1, 2] print(s.loc['a':'b'])</pre>	a 1 b 2 Name: ime, dtype: int32

SERIES - INDEKSIRANJE, ITERIRANJE

Objašnjenje	Kod	Izlaz
Pristup vrednosti preko indeksa (get, set). Ne sme se vršiti set po indeksu van opsega.	<pre>s.iloc[0] = 11 print(s.iloc[0])</pre>	11
Pristup vrednostima preko liste indeksa (get, set). Ne sme se vršiti set po indeksu van opsega.	<pre>s.iloc[[0, 1]] = 11, 22 print(s.iloc[[0, 1]])</pre>	a 11 b 22 Name: ime, dtype: int32
Pristup vrednostima preko opsega indeksa (get, set).	<pre>s.iloc[0:2] = 11, 11 # ekv. s.iloc[0:2] = 11 print(s.iloc[0:2])</pre>	a 11 b 11 Name: ime, dtype: int32
Pristup vrednostima preko niza logičkih vrednosti (get, set).	<pre>s.iloc[[True, False, False]] = 11 print(s.iloc[[True, False, True]])</pre>	a 11 c 7 Name: ime, dtype: int32
Uklanjanje vrednosti sa indeksa uz njeno vraćanje.	<pre>print(s.pop(0))</pre>	1
Pristup parovima indeks, vrednost.	<pre>print(*s.items()) # ekv. print(*s.iteritems())</pre>	('a', 1) ('b', 4) ('c', 7)

SERIES – BINARNE OPERATORSKE FUNKCIJE

Objašnjenje	Kod	Izlaz
Kreiranje serija	<pre>s1 = pd.Series(data=[3, 4, 7], index=['a', 'b', 'c'], dtype=int, name='ime1') s2 = pd.Series(data=[6, 4, 1], index=['a', 'b', 'c'], dtype=int, name='ime2')</pre>	
Sumiranje dve serije (fill_value menja NaN vrednosti)	<pre>print(s1.add(s2, fill_value=3).values) # isto kao (s1 + s2).values</pre>	[9 8 8]
Oduzimanje dve serije	<pre>print(s1.sub(s2).values) # isto kao (s1 - s2).values</pre>	[-3 0 6]
Množenje dve serije	<pre>print(s1.mul(s2).values) # isto kao (s1 * s2).values</pre>	[18 16 7]
Deljenje dve serije	<pre>print(s1.div(s2).values) # isto kao (s1 / s2).values</pre>	[0.5 1. 7.]
Moduo podataka dve serije	<pre>print(s1.mod(s2).values) # isto kao (s1 % s2).values</pre>	[3 0 0]
Stepenovanje podataka dve serije	<pre>print(s1.pow(s2).values) # isto kao (s1 ** s2).values</pre>	[729 256 7]

SERIES – BINARNE OPERATORSKE FUNKCIJE

Objašnjenje	Kod	Izlaz
<p>Kreiranje serija - Neophodno je napisati naredbu:</p> <pre>import numpy as np</pre> <p>kako bi se izvršio uvoz modula numpy, alias np. np.nan – Not a number</p>	<pre>s1 = pd.Series(data=[3, 4, np.nan], index=['a', 'b', 'c']) s2 = pd.Series(data=[6, 4, 1], index=['a', 'b', 'c'])</pre>	
<p>Sumiranje dve serije (fill_value zamenjuje nan)</p>	<pre>print(s1.radd(s2, fill_value=1).values)</pre>	[9. 8. 2.]
<p>Oduzimanje dve serije (od desne se oduzima leva)</p>	<pre>print(s1.rsub(s2, fill_value=1).values)</pre>	[3. 0. 0.]
<p>Množenje dve serije</p>	<pre>print(s1.rmul(s2, fill_value=1).values)</pre>	[18. 16. 1.]
<p>Deljenje dve serije (leva deli desnu)</p>	<pre>print(s1.rdiv(s2, fill_value=1).values)</pre>	[2. 1. 1.]
<p>Moduo podataka dve serije</p>	<pre>print(s1.rmod(s2, fill_value=1).values)</pre>	[0. 0. 0.]
<p>Stepenovanje podataka dve serije</p>	<pre>print(s1.rpow(s2, fill_value=1).values)</pre>	[216. 256. 1.]

SERIES – BINARNE OPERATORSKE FUNKCIJE

Objašnjenje	Kod	Izlaz
Kreiranje serija	<pre>s1 = pd.Series(data=[3, 4, 2], index=['a', 'b', 'c']) s2 = pd.Series(data=[6, 4, 1], index=['a', 'b', 'c'])</pre>	
Upoređivanje dve serije na manje	<pre>print(s1.lt(s2).values) # isto kao (s1 < s2).values</pre>	[True False False]
Upoređivanje dve serije na veće	<pre>print(s1.gt(s2).values) # isto kao (s1 > s2).values</pre>	[False False True]
Upoređivanje dve serije na manje i jednako	<pre>print(s1.le(s2).values) # isto kao (s1 <= s2).values</pre>	[True True False]
Upoređivanje dve serije na veće i jednako	<pre>print(s1.ge(s2).values) # isto kao (s1 >= s2).values</pre>	[False True True]
Upoređivanje dve serije na različito	<pre>print(s1.ne(s2).values) # isto kao (s1 != s2).values</pre>	[True False True]
Upoređivanje dve serije na jednakost	<pre>print(s1.eq(s2).values) # isto kao (s1 == s2).values</pre>	[False True False]

SERIES – BINARNE OPERATORSKE FUNKCIJE

Objašnjenje	Kod	Izlaz
Kreiranje serija	<pre>s1 = pd.Series(data=[3, 4, 2], index=['a', 'b', 'c']) s2 = pd.Series(data=[6.14, 4.123, 1.5], index=['a', 'b', 'c'])</pre>	
Kombinovanje dve serije (func je funkcija sa 2 argumenta (npr. minimum); prvi argument su elementi s1, drugi elementi s2)	<pre>print(s1.combine(s2, func=min).values)</pre>	[3. 4. 1.5]
Zaokruživanje svake vrednosti u seriji na broj decimala	<pre>print(s2.round(1).values)</pre>	[6.1 4.1 1.5]
Skalarni proizvod elemenata dve serije	<pre>print(s1.dot(s2))</pre>	37.912
Upoređivanje serija na jednakost elemenata	<pre>print(s1.equals(s2))</pre>	False

SERIES - OPERACIJE NAD SERIJAMA

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>s = pd.Series(data=[3, 4, 2], index=['a', 'b', 'c'])</pre>	
Primena funkcije nad elementima serije	<pre>print(s.apply(np.square).values)</pre>	[9 16 4]
Primena funkcije nad elementima serije uz prosleđivanje argumenta	<pre>print(s.apply(pow, args=(3,)).values)</pre>	[27 64 8]
Agregacija elemenata serije	<pre>print(s.agg(min))</pre>	2
Agregacije elemenata serije uz više funkcija (rezultat je serija)	<pre>print(s.agg((min, max)))</pre>	min 2 max 4 dtype: int64
Transformacija elemenata serije	<pre>print(s.transform(np.square).values)</pre>	[9 16 4]
Transformacija elemenata serije	<pre>print(s.map(np.square).values)</pre>	[9 16 4]
Grupisanje na elemente sa indeksom 'b' i ostale, a zatim srednja vrednost	<pre>print(s.groupby(lambda x: x == 'b').mean())</pre>	False 2.5 True 4.0 dtype: float64

SERIES - IZRAČUNAVANJA, OPIS

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>s = pd.Series(data=[4, 0, -3, 3, -1])</pre>	
Apsolutna vrednost elemenata serije	<pre>print(s.abs().values)</pre>	[4 0 3 3 1]
Provera na istinitost svih elemenata serije	<pre>print(s.all())</pre>	False
Provera na istinitost barem jednog elementa serije	<pre>print(s.any())</pre>	True
Provera elemenata serije na opseg vrednosti	<pre>print(s.between(left=1, right=5))</pre>	0 True 1 False 2 False 3 True 4 False dtype: bool
Odsecanje elemenata serije na vrednost	<pre>print(s.clip(lower=2, upper=4).values)</pre>	[4 2 2 3 2]
Broj elemenata serije koji su različiti od NaN	<pre>print(s.count())</pre>	5

SERIES - IZRAČUNAVANJA, OPIS

Objašnjenje	Kod	Izlaz
Kumulativni maksimum (maksimum do tog elementa)	<pre>print(s.cummax().values)</pre>	<pre>[4 4 4 4 4]</pre>
Kumulativni minimum (minimum do tog elementa)	<pre>print(s.cummin().values)</pre>	<pre>[4 0 -3 -3 -3]</pre>
Kumulativni proizvod (zaključno sa tim elementom)	<pre>print(s.cumprod().values)</pre>	<pre>[4 0 0 0 0]</pre>
Kumulativna suma (zaključno sa tim elementom)	<pre>print(s.cumsum().values)</pre>	<pre>[4 4 1 4 3]</pre>
Generisanje opisne statistike serije: count - broj mean - srednja vrednost std - stand. Devijacija min - minimalna vrednost max - maksimalna vrednost percentiles - opciono, 20% vrednosti < -1,4 50% je medijana 80% vrednosti < 3.2	<pre>print(s.describe(percentiles=[0.2, 0.5, 0.8]))</pre>	<pre>count 5.000000 mean 0.600000 std 2.880972 min -3.000000 20% -1.400000 50% 0.000000 80% 3.200000 max 4.000000 dtype: float64</pre>

SERIES - IZRAČUNAVANJA, OPIS

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>s = pd.Series(data=[1, 3, 5, 3, 7])</pre>	
Razlika svakog elementa i onog periods unazad	<pre># periods je opciono, podraz. 1 print(s.diff(periods=1).values)</pre>	[nan 2. 2. -2. 4.]
Enumeracija elemenata serije	<pre>val, ind = s.factorize() print('V:', val) print('I:', ind.values)</pre>	V: [0 1 2 1 3] I: [1 3 5 7]
Element sa maksimalnom vrednošću	<pre>print(s.max())</pre>	7
Element sa minimalnom vrednošću	<pre>print(s.min())</pre>	1
Aritmetička sredina elemenata serije	<pre>print(s.mean())</pre>	3.8
Medijana serije, element srednji po vrednosti	<pre>print(s.median())</pre>	3.0
Najčešće pojavljivane vrednosti	<pre>print(s.mode())</pre>	0 3 dtype: int64
Suma svih elemenata serije	<pre>print(s.sum())</pre>	19

SERIES - IZRAČUNAVANJA, OPIS

Objašnjenje	Kod	Izlaz
Promena vrednosti u odnosu na prethodnu vrednost (koliko puta)	<pre>print(s.pct_change())</pre>	<pre>0 NaN 1 2.000000 2 0.666667 3 -0.400000 4 1.333333 dtype: float64</pre>
N najvećih elemenata serije, opadajuće	<pre>print(s.nlargest(n=3).values)</pre>	<pre>[7 5 3]</pre>
N najmanjih elemenata serije, rastuće	<pre>print(s.nsmallest(n=2).values)</pre>	<pre>[1 3]</pre>
Proizvod elemenata serije	<pre>print(s.product()) # isto kao print(s.prod())</pre>	<pre>315</pre>
Jedinstveni elementi serije	<pre>print(s.unique())</pre>	<pre>[1 3 5 7]</pre>
Broj jedinstvenih elemenata serije	<pre>print(s.nunique())</pre>	<pre>4</pre>
Seriya koja predstavlja broj pojavljivanja svih različitih elemenata serije	<pre>print(s.value_counts(sort=True, ascending=True))</pre>	<pre>1 1 5 1 7 1 3 2 dtype: int64</pre>

SERIES - SELEKCIJA

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>s = pd.Series(data=[1, 3, 5, 3, 7])</pre>	
Kreiranje serije od druge serije izbacivanjem zadatih labela indeksa	<pre>print(s.drop([0, 2, 4]))</pre>	<pre>1 3 3 3 dtype: int64</pre>
Kreiranje serije izbacivanjem duplikata (ostaje poslednja pojava)	<pre>print(s.drop_duplicates(keep='last').values)</pre>	<pre>[1 5 3 7]</pre>
Označava duplikate serije (ostaje prva pojava)	<pre>print(s.duplicated(keep='first').values)</pre>	<pre>[False False False True False]</pre>
Vraća prvih n vrednosti (podraz. 5). Negativno n vraća sve vrednosti osim poslednjih n.	<pre>print(s.head(n=3).values)</pre>	<pre>[1 3 5]</pre>
Vraća poslednjih n vrednosti (podraz. 5). Negativno n vraća sve vrednosti osim prvih n.	<pre>print(s.tail(n=2).values)</pre>	<pre>[3 7]</pre>
Indeks najvećeg elementa i najmanjeg elementa	<pre>print(s.idxmax()) print(s.idxmin())</pre>	<pre>4 0</pre>

SERIES – SELEKCIJA, REINDEKSIRANJE

Objašnjenje	Kod	Izlaz
Provera da li se elementi serije nalaze u zadatim vrednostim	<pre>print(s.isin([1, 2, 3]).values)</pre>	<pre>[True True False True False]</pre>
Reindeksiranje elemenata serije (na novim indeksima se nalazi vrednost NaN ili fill_value)	<pre>s = s.reindex(index=s.index - 2, fill_value=-1) print(s)</pre>	<pre>-2 -1 -1 -1 0 1 1 3 2 5 dtype: int64</pre>
Očuvanje redosleda elemenata uz promenu labela indeksa (arange vraća niz vrednosti u opsegu [-2, 3) sa korakom 1)	<pre>s.index = np.arange(-2, 3) print(s) # isto kao: s = s.set_axis(np.arange(-2, 3))</pre>	<pre>-2 1 -1 3 0 5 1 3 2 7 dtype: int64</pre>
Preimenovanje labela indeksa	<pre>s = s.rename(lambda x: x - 2)</pre>	<pre>ind -2 1 -1 3</pre>
Preimenovanje imena serije	<pre>s = s.rename('ime')</pre>	<pre>0 5 1 3</pre>
Preimenovanje imena indeksa	<pre>s = s.rename_axis('ind') print(s)</pre>	<pre>2 7 Name: ime, dtype: int64</pre>

SERIES – SELEKCIJA, REINDEKSIRANJE

Objašnjenje	Kod	Izlaz
Uzorkovanje podataka n - broj podataka ili frac - procenat podataka random_state - za reproducibilne situacije	<pre>sam = s.sample(n=3, random_state=12) # isto kao i: sam = s.sample(frac=0.6, random_state=12) print(sam)</pre>	<pre>0 1 1 3 2 5 dtype: int64</pre>
Dohvatanje elemenata serije sa pozicija	<pre>s = s.take([1, 2]) # ekv. s = s.filter([1, 2]) print(s)</pre>	<pre>1 3 2 5 dtype: int64</pre>
Odsecanje elemenata serije pre i posle indeksa	<pre>s = s.truncate(before=2, after=4) print(s)</pre>	<pre>2 5 3 3 4 7 dtype: int64</pre>
Izmena vrednosti kod kojih uslov nije ispunjen	<pre>s = s.where(s > 3, -1) print(s.values)</pre>	<pre>[-1 -1 5 -1 7]</pre>
Izmena vrednosti kod kojih je uslov ispunjen	<pre>s = s.mask(s > 3, -1) print(s.values)</pre>	<pre>[1 3 -1 3 -1]</pre>
Dodavanje prefiksa i sufiksa labelama indeksa	<pre>s = s.add_prefix('p') s = s.add_suffix('s') print(s.index.values)</pre>	<pre>['p0s' 'p1s' 'p2s' 'p3s' 'p4s']</pre>

SERIES - UPRAVLJANJE NaN VREDNOSTIMA

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>s = pd.Series(data=[1, np.nan, 5, np.nan, 7])</pre>	
Popunjavanje NaN vrednosti (sinonim za s.bfill()) Koristi prvu sledeću vrednost različitu od NaN za popunu NaN vrednosti	<pre>s = s.backfill() #ekv. s = s.fillna(method='backfill') #ekv. s = s.replace(np.nan, method='bfill') print(s.values)</pre>	[1. 5. 5. 7. 7.]
Eliminacija NaN vrednosti	<pre>s = s.dropna() print(s)</pre>	0 1.0 2 5.0 4 7.0 dtype: float64
Popunjavanje NaN vrednosti (sinonim za s.pad()) Koristi prvu prethodnu vrednost različitu od NaN za popunu NaN vrednosti	<pre>s = s.ffill() # ekv. s = s.fillna(method='ffill') # ekv. s = s.replace(np.nan, method='ffill') print(s.values)</pre>	[1. 1. 5. 5. 7.]
Popunjavanje NaN vrednosti određenom vrednošću	<pre>s = s.fillna(0) print(s.values)</pre>	[1. 0. 5. 0. 7.]

SERIES - UPRAVLJANJE NaN VREDNOSTIMA

Objašnjenje	Kod	Izlaz
Popunjavanje NaN vrednosti koristeći linearnu interpolaciju	<pre>s = s.interpolate() print(s.values)</pre>	[1. 3. 5. 6. 7.]
Detektovanje NaN vrednosti	<pre>s = s.isna() # ekv. s = s.isnull() print(s.values)</pre>	[False True False True False]
Detektovanje vrednosti koje nisu NaN	<pre>s = s.notna() # ekv. s = s.notnull() print(s.values)</pre>	[True False True False True]
Zamena vrednosti vrednošću	<pre>s = s.replace(np.nan, 0) print(s.values)</pre>	[1. 0. 5. 0. 7.]
Zamena liste vrednosti vrednošću	<pre>s = s.replace([1, 5], 0) print(s.values)</pre>	[0. nan 0. nan 7.]
Zamena liste vrednosti listom vrednosti	<pre>s = s.replace([1, np.nan], [2, 0]) print(s.values)</pre>	[2. 0. 5. 0. 7.]
Zamena liste vrednosti rečnikom vrednosti	<pre>s = s.replace({np.nan: 0, 1: 2}) print(s.values)</pre>	[2. 0. 5. 0. 7.]

SERIES - OBLIKOVANJE, SORTIRANJE

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>s = pd.Series(data=[1, 5, 3, 7, np.nan])</pre>	
Labele indeksa u sortiranom poretku prema vrednostima serije (-1 za NaN)	<pre>s = s.argsort() print(s.values)</pre>	<pre>[0 2 1 3 -1]</pre>
Labela indeksa najmanjeg elementa	<pre>print(s.argmin())</pre>	0
Labela indeksa najvećeg elementa	<pre>print(s.argmax())</pre>	3
Sortiranje serije prema vrednosti elemenata serije (ascending označava poredak rastuće/opadajuće)	<pre>s = s.sort_values(ascending=False) print(s.index.values) print(s.values)</pre>	<pre>[3 1 2 0 4] [7. 5. 3. 1. nan]</pre>
Sortiranje serije prema vrednosti labela indeksa (ascending označava poredak rastuće/opadajuće)	<pre>s = s.sort_index(ascending=False) print(s.index.values) print(s.values)</pre>	<pre>[4 3 2 1 0] [nan 7. 3. 5. 1.]</pre>
numpy ndarray vrednosti	<pre>print(s.ravel())</pre>	<pre>[1. 5. 3. 7. nan]</pre>

SERIES - OBLIKOVANJE, SORTIRANJE

Objašnjenje	Kod	Izlaz
Ekspanzija elemenata serije koji su lista. Postavljanje novih labela indeksa: s = s.set_axis(np.arange(0, len(s))) ili ignore_index=True	<pre>s = pd.Series(data=[[1, 2], [3, 4], 5]) s = s.explode(ignore_index=False) print(s.index.values) print(s.values)</pre>	<pre>[0 0 1 1 2] [1 2 3 4 5]</pre>
Kreiranje serije	<pre>s = pd.Series(data=[1, 3, 5, 7])</pre>	
Pronalaženje pozicije za umetanje vrednosti u sortiranu seriju	<pre>print(s.searchsorted(2)) print(s.searchsorted(5, side='left')) print(s.searchsorted(5, side='right'))</pre>	<pre>1 2 3</pre>
Ponavljanje elemenata serije (koliko puta)	<pre>s = s.repeat(2) print(s.index.values) print(s.values)</pre>	<pre>[0 0 1 1 2 2 3 3 4 4] [1 1 2 2 3 3 5 5 7 7]</pre>
Ponavljanje elemenata serije (koji indeks koliko puta)	<pre>s = s.repeat([2, 0, 0, 0, 1]) print(s.index.values) print(s.values)</pre>	<pre>[0 0 4] [1 1 7]</pre>

SERIES - OBLIKOVANJE, SORTIRANJE

Objašnjenje	Kod	Izlaz
Kreiranje skalara na osnovu serije sa samo jednim elementom. Ukoliko serija ima više elemenata operacija je bez efekta.	<pre>s = pd.Series(data=[1, 2]) print(s.squeeze()) s = pd.Series(data=[1]) print(s.squeeze())</pre>	<pre>0 1 1 2 dtype: int64 1</pre>
Krairanje pogleda na seriju. Nije u pitanju kopija.	<pre>s = pd.Series(data=[1, 2]) sv = s.view(dtype=s.dtype) sv.loc[0] = 13 print(s)</pre>	<pre>0 13 1 2 dtype: int64</pre>
Kreiranje multi-indeksa serije prilikom kreiranja serije	<pre>ind = [[1, 2], [3, 4]] s = pd.Series(data=[1, 5, 3, 7], index=pd.MultiIndex .from_product(ind, names=['one', 'two'])) print(s) print(s[1][4])</pre>	<pre>one two 1 3 1 4 5 2 3 3 4 7 dtype: int64 5</pre>
Obrtanje redosleva nivoa multi-indeksa	<pre>lvl = [1, 0] s = s.reorder_levels(lvl) print(s)</pre>	<pre>two one 3 1 1 4 1 5 3 2 3 4 2 7 dtype: int64</pre>

SERIES – KOMBINOVANJE, POREĐENJE

Objašnjenje	Kod	Izlaz
Kreiranje serija	<pre>s = pd.Series(data=[1, 2], index=[0, 1]) p = pd.Series(data=[3, 4], index=[0, 1])</pre>	
Konkatenacija serija	<pre>s = s.append(p) print(s.index.values) print(s.values)</pre>	<pre>[0 1 0 1] [1 2 3 4]</pre>
Konkatenacija serija (ignorisanje indeksa)	<pre>s = s.append(p, ignore_index=True) print(s.index.values) print(s.values)</pre>	<pre>[0 1 2 3] [1 2 3 4]</pre>
Poređenje dve serije uz prikaz razlika (mogu se porediti samo identično labelirane serije)	<pre>print(s.compare(p))</pre>	<pre> self other 0 1 3 1 2 4</pre>
Modifikacija serije vrednostima druge serije (samo oni indeksi koji su prisutni u drugoj seriji)	<pre>p.loc[2] = 5 s.update(p) print(s)</pre>	<pre>0 3 1 4 dtype: int64</pre>

PODRŽANI TIPOVI PODATAKA

DataFrame predstavlja dvodimenzionu (tabelarnu) strukturu podataka, kojoj se pristupa na osnovu labela, koje su definisane osama labela (vrste i kolone). Svaka kolona sadrži vrednosti jednog atributa, a svaki red predstavlja skup vrednosti svih atributa (kolona). Ukoliko želimo da kreiramo objekat ovog tipa neophodno je napisati sledeću naredbu:

```
df = pd.DataFrame(data=[[1, 2, 3], [4, 5, 6], [7, 8, 9]],  
                  index=[0, 1, 2], columns=['a', 'b', 'c'], dtype=int)
```

data je obavezan argument i predstavlja podatke (lista listi, numpy ndarray, rečnik itd.).

index je opcioni argument i predstavlja labele, na osnovu kojih se pristupa redovima tabele. Mora biti iste dužine kao i argument data.

columns je opcioni argument i predstavlja labele, na osnovu kojih se pristupa kolonama tabele. Mora biti iste dužine kao i svaki element argument data (podlista liste).

dtype je opcioni argument i predstavlja tip podataka serije. Može se zaključiti iz data.

Može se formirati pomoću rečnika (labele kolona su ključevi rečnika):

```
df = pd.DataFrame({'a': [1, 4, 7], 'b': [2, 5, 8], 'c': [3, 6, 9]})
```

DATAFRAME - ATRIBUTI

Objašnjenje	Kod	Izlaz
Kreiranje DataFrame-a	<pre>df = pd.DataFrame(data=[[1, 2, 3], [4, 5, 6], [7, 8, 9]], index=[0, 1, 2], columns=['a', 'b', 'c'], dtype=int)</pre>	
Labele indeksa (redova)	<pre>print(df.index.values)</pre>	<pre>[0 1 2]</pre>
Labele kolona	<pre>print(df.columns.values)</pre>	<pre>['a' 'b' 'c']</pre>
Tipovi podataka	<pre>print(df.dtypes)</pre>	<pre>a int32 b int32 c int32 dtype: object</pre>
Info (koncizan pregled)	<pre>print(df.info())</pre>	<pre><class 'pandas.core.frame.DataFrame'> Int64Index: 3 entries, 0 to 2 Data columns (total 3 columns): # Column Non-Null Count Dtype --- - 0 a 3 non-null int32 1 b 3 non-null int32 2 c 3 non-null int32 dtypes: int32(3) memory usage: 60.0 bytes</pre>
Vrednosti reprezentovane numpy nizom	<pre>print(df.values)</pre>	<pre>[[1 2 3] [4 5 6] [7 8 9]]</pre>

DATAFRAME - ATRIBUTI

Objašnjenje	Kod	Izlaz
Podskup vrednosti odgovarajućeg tipa; vraća se DataFrame	<pre>print(df.select_dtypes(include=[int]))</pre>	<pre> a b c 0 1 2 3 1 4 5 6 2 7 8 9</pre>
Lista osa	<pre>print(df.axes)</pre>	<pre>[Int64Index([0, 1, 2], dtype='int64'), Index(['a', 'b', 'c'], dtype='object')]</pre>
Broj osa	<pre>print(df.ndim)</pre>	2
Broj elemenata DataFrame-a	<pre>print(df.size)</pre>	9
Oblik DataFrame-a kao torka	<pre>print(df.shape)</pre>	(3, 3)
Memorijsko zauzeće DataFrame-a u bajtovima	<pre>print(df.memory_usage())</pre>	<pre>Index 24 a 12 b 12 c 12 dtype: int64</pre>
Da li je DataFrame prazan (dimenzija bilo koje ose je 0)	<pre>print(df.empty)</pre>	False

DATAFRAME - KONVERZIJE

Objašnjenje	Kod	Izlaz
Konverzija elemenata u zadati tip podataka (vraća se kopija)	<pre>df = df.astype(dtype=float) print(df)</pre>	<pre> a b c 0 1.0 2.0 3.0 1 4.0 5.0 6.0 2 7.0 8.0 9.0</pre>
Konverzija određenih kolona u zadate tipove	<pre>df = df.astype({'a': float, 'b': bool}) print(df)</pre>	<pre> a b c 0 1.0 True 3 1 4.0 True 6 2 7.0 True 9</pre>
Kreiranje kopije DataFrame-a (deep=True označava duboku kopiju)	<pre>copy = df.copy(deep=False) print(copy)</pre>	<pre> a b c 0 1 2 3 1 4 5 6 2 7 8 9</pre>
Konvertovanje DataFrame-a sa jednom vrednošću u bool tip (DataFrame sa više vrednosti proizvodi ValueError)	<pre>print(pd.DataFrame({'a': [True]}).bool())</pre>	<pre>True</pre>

DATAFRAME - INDEKSIRANJE, ITERACIJA

Objašnjenje	Kod	Izlaz
Kreiranje DataFrame-a	<pre>df = pd.DataFrame(data=[[1, 2], [3, 4], [5, 6]], index=[0, 1, 2], columns=['a', 'b'])</pre>	
Dohvatanje prvih n redova (podrazumevano 5). Negativno n vraća sve redove osim poslednjih n.	<pre>print(df.head()) print(df.head(1))</pre>	<pre> a b 0 1 2 1 3 4 2 5 6 a b 0 1 2</pre>
Pristup jednoj vrednosti preko labela/ključa (get, set)	<pre>df.at[0, 'b'] = -1 print(df.at[0, 'b']) print(df.at[1, 'a'])</pre>	<pre>-1 3</pre>
Pristup jednoj vrednosti preko celobrojnih indeksa (get, set). Ne sme se vršiti set po indeksu van opsega.	<pre>df.iat[0, 1] = -1 print(df.iat[0, 1]) print(df.iat[1, 0]) # ne sme df.iat[1, 3] = 11</pre>	<pre>-1 3</pre>

DATAFRAME - INDEKSIRANJE, ITERACIJA

Objašnjenje	Kod	Izlaz
Pristup vrednosti preko labela osa (get, set)	<pre>df.loc[0, 'b'] = 11 print(df.loc[0, 'b'])</pre>	11
Pristup jednom redu (seriji) preko labela indeksa (get, set)	<pre>print(df.loc[0])</pre>	a 1 b 2 Name: 0, dtype: int64
Pristup jednom delu DataFrame-a preko liste labela redova i kolona (get, set)	<pre>df.loc[[0, 1], ['a', 'b']] = 0 print(df)</pre>	a b 0 0 0 1 0 0 2 5 6
Pristup delu DataFrame-a preko opsega labela (get, set)	<pre>print(df.loc[0:1, 'a':'b'])</pre>	a b 0 1 2 1 3 4
Pristup redovima DataFrame-a preko bool vrednosti (get, set)	<pre>print(df.loc[[True, False, True]])</pre>	a b 0 1 2 2 5 6
Pristup delu kolone 'a' čiji su elementi u koloni 'b' veći od 3 (get, set)	<pre>print(df.loc[df['b'] > 3, ['a']]) # ekv. print(df.loc[[False, True, True], ['a']])</pre>	a 1 3 2 5

DATAFRAME - INDEKSIRANJE, ITERACIJA

Objašnjenje	Kod	Izlaz
Pristup delu DataFrame-a čiji su elementi kolone 'a' jednaki 1 (parametar lambda funkcije je DataFrame)	<pre>print(df.loc[lambda d: d['a'] == 1])</pre>	<pre> a b 0 1 2</pre>
Postavljanje vrednosti celoj koloni 'a' (: označava sve redove)	<pre>df.loc[:, 'a'] = 0 print(df)</pre>	<pre> a b 0 0 2 1 0 4 2 0 6</pre>
Pristup jednom redu (seriji) preko celobrojnog indeksa (get, set). Ne sme se vršiti set po indeksu van opsega.	<pre>df.iloc[0] = 0 print(df)</pre>	<pre> a b 0 0 0 1 3 4 2 5 6</pre>
Pristup redovima preko liste indeksa (get, set). Ne sme se vršiti set po indeksu van opsega.	<pre>df.iloc[[0, 1]] = 0 print(df)</pre>	<pre> a b 0 0 0 1 0 0 2 5 6</pre>

DATAFRAME - INDEKSIRANJE, ITERACIJA

Objašnjenje	Kod	Izlaz
Pristup redovima preko opsega indeksa. Opseg ne obuhvata gornju granicu (get, set). Ne sme se vršiti set po indeksu van opsega.	<pre>df.iloc[:2] = 0 print(df)</pre>	<pre> a b 0 0 0 1 0 0 2 5 6</pre>
Pristup redovima DataFrame-a preko bool vrednosti (get, set). Ne sme se vršiti set po indeksu van opsega.	<pre>print(df.iloc[[True, False, True]])</pre>	<pre> a b 0 1 2 2 5 6</pre>
Pristup redovima DataFrame-a čije su labela parna vrednost (parametar lambda funkcije je DataFrame).	<pre>print(df.iloc[lambda d: d.index % 2 == 0])</pre>	<pre> a b 0 1 2 2 5 6</pre>
Pristup jednom elementu DataFrame-a. Ne sme se vršiti set po indeksu van opsega.	<pre>df.iloc[0, 0] = 0 print(df)</pre>	<pre> a b 0 0 2 1 3 4 2 5 6</pre>

DATAFRAME - INDEKSIRANJE, ITERACIJA

Objašnjenje	Kod	Izlaz
<p>Pristup delu DataFrame-a. Ne sme se vršiti set po indeksu van opsega.</p>	<pre>df.iloc[[0, 2], [0, 1]] = 0 print(df)</pre>	<pre> a b 0 0 0 1 3 4 2 0 0</pre>
<p>Pristup delu DataFrame-a po opsegu celobrojnih indeksa. Opseg ne obuhvata gornju granicu (get, set). Ne sme se vršiti set po indeksu van opsega.</p>	<pre>df.iloc[0:2, 0:2] = 0 print(df)</pre>	<pre> a b 0 0 0 1 0 0 2 5 6</pre>
<p>Pristup kolonama DataFrame-a pomoću liste bool vrednosti.</p>	<pre>df.iloc[:, [True, False]] = 0 print(df)</pre>	<pre> a b 0 0 2 1 0 4 2 0 6</pre>
<p>Pristup koloni 'a' DataFrame-a (parametar lambda funkcije je DataFrame). Lambda funkcija utvrđuje kojim kolonama se pristupa.</p>	<pre>df.iloc[:, lambda d: d.columns == 'a'] = 0 print(df)</pre>	<pre> a b 0 0 2 1 0 4 2 0 6</pre>

DATAFRAME - INDEKSIRANJE, ITERACIJA

Objašnjenje	Kod	Izlaz
Umetanje vrednosti kao nove kolone na zadatu lokaciju (greška je umetnuti kolonu van opsega indeksa ili ukoliko postoji kolona zadanog imena)	<pre>df.insert(loc=0, column='c', value=0) print(df)</pre>	<pre> c a b 0 0 1 2 1 0 3 4 2 0 5 6</pre>
Umetanje liste vrednosti kao nove kolone na zadatu lokaciju	<pre>df.insert(loc=0, column='c', value=[0, 0, 0]) print(df)</pre>	<pre> c a b 0 0 1 2 1 0 3 4 2 0 5 6</pre>
Umetanje serije kao nove kolone na zadatu lokaciju	<pre>df.insert(loc=0, column='c', value=pd.Series([0, 0, 0])) print(df)</pre>	<pre> c a b 0 0 1 2 1 0 3 4 2 0 5 6</pre>
Pristup parovima (labela kolone, serija)	<pre>for cn, s in df.items(): print(cn, ':', s.values) # ekv. df.iteritems()</pre>	<pre>a : [1 3 5] b : [2 4 6]</pre>
Pristup parovima (labela vrste, serija)	<pre>for rn, s in df.iterrows(): print(rn, ':', s.values)</pre>	<pre>0 : [1 2] 1 : [3 4] 2 : [5 6]</pre>

DATAFRAME - INDEKSIRANJE, ITERACIJA

Objašnjenje	Kod	Izlaz
Pristup labelama kolona	<pre>print(df.keys().values)</pre>	<pre>['a' 'b']</pre>
Pristup redovima DataFrame-a kao torkama (index označava da li torku čini i labela reda)	<pre>for row in df.itertuples(index=True): print(row) print(pd.Series(data=row).values)</pre>	<pre>Pandas(Index=0, a=1, b=2) [0 1 2] Pandas(Index=1, a=3, b=4) [1 3 4] Pandas(Index=2, a=5, b=6) [2 5 6]</pre>
Uklanja kolonu i vraća je kao seriju. Labela kolone koja se uklanja se zadaje (greška je ukoliko kolona ne postoji).	<pre>print(df.pop('a')) # isto kao i: del df['a'] print(df)</pre>	<pre>0 1 1 3 2 5 Name: a, dtype: int64 b 0 2 1 4 2 6</pre>
Dohvatanje poslednjih n redova (podraz. 5) Negativno n vraća sve redove osim prvih n.	<pre>print(df.tail()) print(df.tail(1))</pre>	<pre> a b 0 1 2 1 3 4 2 5 6 a b 2 5 6</pre>

DATAFRAME - INDEKSIRANJE, ITERACIJA

Objašnjenje	Kod	Izlaz
Dohvatanje kolone po labeli. Ukoliko kolone zadate labelom nema, a podraz. vrednost nije navedena, vraća None. U suprotnom vraća podraz. vrednost.	<pre>print(df.get(key='a', default=0))</pre>	<pre>0 1 1 3 2 5 Name: a, dtype: int64</pre>
DataFrame bool vrednosti prisutnosti elemenata u zadanom skupu	<pre>print(df.isin([2, 5]))</pre>	<pre> a b 0 False True 1 False False 2 True False</pre>
Izmena vrednosti kod kojih je uslov ispunjen. Menja objekat df i vraća None.	<pre>df.mask(df % 2 == 1, 0, inplace=True) print(df)</pre>	<pre> a b 0 0 2 1 0 4 2 0 6</pre>
Izmena vrednosti kod kojih uslov nije ispunjen. Vraća se kopija objekta df.	<pre>print(df.where(df % 2 == 1, 0), inplace=False)</pre>	<pre> a b 0 1 0 1 3 0 2 5 0</pre>

DATAFRAME - BIN. OPERATORSKE FUNKCIJE

Objašnjenje	Kod	Izlaz
Kreiranje DataFrame-ova	<pre>d1 = pd.DataFrame(data=[[1], [2], [3]], index=[0, 1, 2], columns=['a']) d2 = pd.DataFrame(data=[[2], [4], [6]], index=[0, 1, 2], columns=['a'])</pre>	
Sumiranje dva DataFrame-a (fill_value menja NaN vrednosti)	<pre>print(d1.add(d2, fill_value=0)) # isto kao d1 + d2</pre>	<pre>a 0 3 1 6 2 9</pre>
Oduzimanje dva DataFrame-a	<pre>print(d1.sub(d2)) # isto kao d1 - d2</pre>	<pre>a 0 -1 1 -2 2 -3</pre>
Množenje dva DataFrame-a	<pre>print(d1.mul(d2)) # isto kao d1 * d2</pre>	<pre>a 0 2 1 8 2 18</pre>
Deljenje dva DataFrame-a	<pre>print(d1.div(d2)) # isto kao d1 / d2</pre>	<pre>a 0 0.5 1 0.5 2 0.5</pre>
Moduo podataka dva DataFrame-a	<pre>print(d1.div(d2)) # isto kao d1 % d2</pre>	<pre>a 0 1 1 2 2 3</pre>

DATAFRAME - BIN. OPERATORSKE FUNKCIJE

Objašnjenje	Kod	Izlaz
Stepenovanje podataka dva DataFrame-a	<pre>print(d1.pow(d2)) # isto kao d1 ** d2</pre>	<pre> a 0 1 1 16 2 729</pre>
Sumiranje dva DataFrame-a (fill_value zamenjuje nan)	<pre>print(d1.radd(d2))</pre>	<pre> a 0 3 1 6 2 9</pre>
Oduzimanje dva DataFrame-a (od desne se oduzima leva)	<pre>print(d1.rsub(d2))</pre>	<pre> a 0 1 1 2 2 3</pre>
Množenje dva DataFrame-a	<pre>print(d1.rmul(d2))</pre>	<pre> a 0 2 1 8 2 18</pre>
Deljenje dva DataFrame-a (leva deli desnu)	<pre>print(d1.rdiv(d2))</pre>	<pre> a 0 2.0 1 2.0 2 2.0</pre>
Moduo podataka dva DataFrame-a	<pre>print(d1.rmod(d2))</pre>	<pre> a 0 0 1 0 2 0</pre>
Stepenovanje podataka dva DataFrame-a	<pre>print(d1.rpow(d2))</pre>	<pre> a 0 2 1 16 2 216</pre>

DATAFRAME - BIN. OPERATORSKE FUNKCIJE

Objašnjenje	Kod	Izlaz
Upoređivanje dva DataFrame-a na manje	<pre>print(d1.lt(d2)) # isto kao d1 < d2</pre>	<pre>a 0 True 1 True 2 True</pre>
Upoređivanje dva DataFrame-a na veće	<pre>print(d1.gt(d2)) # isto kao d1 > d2</pre>	<pre>a 0 False 1 False 2 False</pre>
Upoređivanje dva DataFrame-a na manje i jednako	<pre>print(d1.le(d2)) # isto kao d1 <= d2</pre>	<pre>a 0 True 1 True 2 True</pre>
Upoređivanje dva DataFrame-a na veće i jednako	<pre>print(d1.ge(d2)) # isto kao d1 >= d2</pre>	<pre>a 0 False 1 False 2 False</pre>
Upoređivanje dva DataFrame-a na različito	<pre>print(d1.ne(d2)) # isto kao d1 != d2</pre>	<pre>a 0 False 1 False 2 False</pre>
Upoređivanje dva DataFrame-a na jednakost	<pre>print(d1.eq(d2)) # isto kao d1 == d2</pre>	<pre>a 0 False 1 False 2 False</pre>

DATAFRAME - BIN. OPERATORSKE FUNKCIJE

Objašnjenje	Kod	Izlaz
Kreiranje DataFrame-ova	<pre>d1 = pd.DataFrame(data=[[3, 4, 1], [2, 5, 0], [4, 1, 0]]) d2 = pd.DataFrame(data=[[2, 3, 1], [4, 2, 1], [6, 3, 1]])</pre>	
Kombinovanje kolona dva DataFrame-a (argumenti lambda funkcije su dve serije, koji predstavljaju kolone prvog i drugog DataFrame-a)	<pre>print(d1.combine(d2, lambda s1, s2: s1 if s1.sum() < s2.sum() else s2))</pre>	<pre> 0 1 2 0 3 3 1 1 2 2 0 2 4 3 0</pre>
Matrični proizvod elemenata dva DataFrame-a	<pre>print(d1.dot(d2))</pre>	<pre> 0 1 2 0 28 20 8 1 24 16 7 2 12 14 5</pre>
Upoređivanje dva DataFrame-a na jednakost	<pre>print(d1.equals(d2))</pre>	False

DATAFRAME – OPERACIJE NAD TABELAMA

Objašnjenje	Kod	Izlaz
Kreiranje DataFrame-a	<pre>d = pd.DataFrame(data=[[1, 2, 3], [4, 5, 6], [7, 8, 9]])</pre>	
Primena funkcije nad elementima DataFrame-a uz prosleđivanje argumenata	<pre>print(d.apply(pow, args=(2,)))</pre>	<pre> 0 1 2 0 1 4 9 1 16 25 36 2 49 64 81</pre>
Primena funkcije nad kolonama (axis=0)	<pre>print(d.apply(np.sum, axis=0))</pre>	<pre>0 12 1 15 2 18 dtype: int64</pre>
Primena funkcije nad redovima (axis=1)	<pre>print(d.apply(np.sum, axis=1))</pre>	<pre>0 6 1 15 2 24 dtype: int64</pre>
Rezultati u obliku liste se ekspanuju u kolone	<pre>f = lambda r: [-v for v in r] * 2 df = d.apply(f, axis=1, result_type='expand') print(df)</pre>	<pre> 0 1 2 3 4 5 0 -1 -2 -3 -1 -2 -3 1 -4 -5 -6 -4 -5 -6 2 -7 -8 -9 -7 -8 -9</pre>
Povratna vrednost je objekat serije	<pre>f = lambda r: [-v for v in r] * 2 ser = d.apply(f, axis=1, result_type='reduce') print(ser)</pre>	<pre>0 [-1, -2, -3, -1, -2, -3] 1 [-4, -5, -6, -4, -5, -6] 2 [-7, -8, -9, -7, -8, -9] dtype: object</pre>

DATAFRAME - OPERACIJE NAD TABELAMA

Objašnjenje	Kod	Izlaz
Vrši se broadcast rezultata (zadržava se oblik)	<pre>f = lambda r: [-v for v in r] df = d.apply(f, axis=1, result_type='broadcast') print(df)</pre>	<pre> 0 1 2 0 -1 -2 -3 1 -4 -5 -6 2 -7 -8 -9</pre>
Primena funkcije nad elementima DataFrame-a	<pre>print(d.applymap(np.square))</pre>	<pre> 0 1 2 0 0 1 4 9 1 16 25 36 2 49 64 81</pre>
Primena funkcije nad DataFrame-om (lančanje)	<pre>print(d.pipe(pow,4).pipe(np.sqrt). pipe(lambda x: -x + 1))</pre>	<pre> 0 1 2 0 0.0 -3.0 -8.0 1 -15.0 -24.0 -35.0 2 -48.0 -63.0 -80.0</pre>
Agregacija elemenata serije po kolonama (axis=0) ili vrstama (axis=1)	<pre>print(d.agg(min, axis=0).values) print(d.agg(max, axis=1).values)</pre>	<pre>[1 2 3] [3 6 9]</pre>
Agregacije elemenata serije uz više funkcija (rezultat je DataFrame)	<pre>print(d.agg((min, max), axis=0))</pre>	<pre> 0 1 2 min 1 2 3 max 7 8 9</pre>

DATAFRAME - OPERACIJE NAD TABELAMA

Objašnjenje	Kod	Izlaz
Transformacija elemenata DataFrame-a	<pre>print(d.transform(lambda x: x + 1))</pre>	<pre> 0 1 2 0 2 3 4 1 5 6 7 2 8 9 10</pre>
Transformacija elemenata DataFrame-a	<pre>print(s.map(np.square).values)</pre>	<pre>[9 16 4]</pre>
Grupisanje po redovima (axis=0), a zatim suma	<pre>print(d.groupby(by=lambda x: x % 2 == 0, axis=0).sum())</pre>	<pre> 0 1 2 False 4 5 6 True 8 10 12</pre>
Grupisanje po kolonama (axis=1), a zatim suma	<pre>print(d.groupby(by=lambda x: x % 2 == 0, axis=1).sum())</pre>	<pre> False True 0 2 4 1 5 10 2 8 16</pre>
Grupisanje po vrednosti elemenata 0. kolone, a zatim suma and 2. kolonom	<pre>print(d.groupby(0, axis=1)[2].sum())</pre>	<pre>0 1 3 4 6 7 9 Name: 2, dtype: int64</pre>

DATAFRAME - IZRAČUNAVANJA, OPIS

Objašnjenje	Kod	Izlaz
Kreiranje DataFrame-a	<pre>d = pd.DataFrame(data=[[1, 2, 0], [3, 4, 5], [6, 7, 8]])</pre>	
Apsolutna vrednost elemenata DataFrame-a	<pre>print(d.abs())</pre>	<pre> 0 1 2 0 1 2 0 1 3 4 5 2 6 7 8</pre>
Provera na istinitost svih elemenata po kolonama	<pre>print(s.all(axis=0))</pre>	<pre>0 True 1 True 2 False dtype: bool</pre>
Provera na istinitost svih elemenata po redovima	<pre>print(s.all(axis=1))</pre>	<pre>0 False 1 True 2 True dtype: bool</pre>
Provera na istinitost barem jednog elementa po kolonama	<pre>print(s.any(axis=0))</pre>	<pre>0 True 1 True 2 True dtype: bool</pre>
Provera na istinitost barem jednog elementa po redovima	<pre>print(s.any(axis=1))</pre>	<pre>0 True 1 True 2 True dtype: bool</pre>

DATAFRAME - IZRAČUNAVANJA, OPIS

Objašnjenje	Kod	Izlaz
Odsecanje elemenata na opseg vrednosti	<pre>print(d.clip(lower=2, upper=4))</pre>	<pre>0 1 2 0 2 2 2 1 3 4 4 2 4 4 4</pre>
Određivanje koeficijenta korelacije kolona	<pre>print(d.corr())</pre>	<pre>0 1 2 0 1.000000 1.000000 0.966823 1 1.000000 1.000000 0.966823 2 0.966823 0.966823 1.000000</pre>
Određivanje kovarijanse kolona	<pre>print(d.cov())</pre>	<pre>0 1 2 0 6.333333 6.333333 9.833333 1 6.333333 6.333333 9.833333 2 9.833333 9.833333 16.333333</pre>
Broj NaN vrednosti po kolonama (axis=0) i vrstama (axis=1)	<pre>print(d.count(axis=0).values) print(d.count(axis=1).values)</pre>	<pre>[3 3 3] [3 3 3]</pre>
Kumulativni maksimum po kolonama (axis=0) i vrstama (axis=1)	<pre>print(d.cummax(axis=1))</pre>	<pre>0 1 2 0 1 2 2 1 3 4 5 2 6 7 8</pre>
Kumulativni minimum po kolonama (axis=0) i vrstama (axis=1)	<pre>print(d.cummin(axis=0))</pre>	<pre>0 1 2 0 1 2 0 1 1 2 0 2 1 2 0</pre>

DATAFRAME - IZRAČUNAVANJA, OPIS

Objašnjenje	Kod	Izlaz
Kumulativni proizvod po kolonama (axis=0) ili vrstama (axis=1)	<pre>print(d.cumprod(axis=1))</pre>	<pre>0 1 2 0 1 2 0 1 3 12 60 2 6 42 336</pre>
Kumulativni suma po kolonama (axis=0) ili vrstama (axis=1)	<pre>print(d.cumsum(axis=0))</pre>	<pre>0 1 2 0 1 2 0 1 4 6 5 2 10 13 13</pre>
Generisanje opisne statistike serije: count - broj mean - srednja vrednost std - stand. Devijacija min - minimalna vrednost max - maksimalna vrednost percentiles - opciono, 20% vrednosti < -1,4 50% je medijana 80% vrednosti < 3.2	<pre>print(d.describe(percentiles=[0.2, 0.5, 0.8]))</pre>	<pre>count 0 1 2 count 3.000000 3.000000 3.000000 mean 3.333333 4.333333 4.333333 std 2.516611 2.516611 4.041452 min 1.000000 2.000000 0.000000 20% 1.800000 2.800000 2.000000 50% 3.000000 4.000000 5.000000 80% 4.800000 5.800000 6.800000 max 6.000000 7.000000 8.000000</pre>

DATAFRAME - IZRAČUNAVANJA, OPIS

Objašnjenje	Kod	Izlaz
Razlika svakog elementa i onog periods unazad (opciono, podr. 1) po kolonama (axis=0) i vrstama (axis=1)	<pre>print(d.diff(periods=1, axis=0))</pre>	<pre> 0 1 2 0 NaN NaN NaN 1 2.0 2.0 5.0 2 3.0 3.0 3.0</pre>
Apsolutna vrednost standardne devijacije po kolonama (axis=0) i vrstama (axis=1)	<pre>print(d.mad(axis=0))</pre>	<pre>0 1.777778 1 1.777778 2 2.888889 dtype: float64</pre>
Element sa maksimalnom vrednošću po kolonama (axis=0) i vrstama (axis=1)	<pre>print(d.max(axis=0).values)</pre>	<pre>[6 7 8]</pre>
Element sa minimalnom vrednošću po kolonama (axis=0) i vrstama (axis=1)	<pre>print(d.min(axis=0).values)</pre>	<pre>[1 2 0]</pre>
Aritmetička sredina po kolonama (axis=0) i vrstama (axis=1)	<pre>print(d.mean(axis=0).values)</pre>	<pre>[3.33333333 4.33333333 4.33333333]</pre>
Medijana po kolonama (axis=0) i vrstama (axis=1)	<pre>print(s.median(axis=0).values)</pre>	<pre>[3. 4. 5.]</pre>
Najčešće pojavljivane vrednosti	<pre>print(d.mode(axis=0)[0].values)</pre>	<pre>[1 3 6]</pre>

DATAFRAME - IZRAČUNAVANJA, OPIS

Objašnjenje	Kod	Izlaz
Promena vrednosti u odnosu na prethodnu vrednost (koliko puta)	<pre>print(d.pct_change(periods=1))</pre>	<pre> 0 1 2 0 NaN NaN NaN 1 2.0 1.00 inf 2 1.0 0.75 0.6</pre>
Proizvod elemenata po kolonama (axis=0) ili vrstama (axis=1)	<pre>print(d.product(axis=0)) isto kao print(d.prod(axis=0))</pre>	<pre>0 18 1 56 2 0 dtype: int64</pre>
Rang elemenata po vrednosti po kolonama (axis=0) ili vrstama (axis=1)	<pre>print(d.rank(axis=0, ascending=True))</pre>	<pre> 0 1 2 0 1.0 1.0 1.0 1 2.0 2.0 1.0 2 3.0 3.0 3.0</pre>
Zaokruživanje vrednosti na broj decimala	<pre>print(d.round(2))</pre>	<pre> 0 1 2 0 1 2 0 1 3 4 5 2 6 7 8</pre>
Suma elemenata po kolonama (axis=0) ili vrstama (axis=1)	<pre>print(d.sum(axis=0))</pre>	<pre>0 10 1 13 2 13 dtype: int64</pre>

DATAFRAME - IZRAČUNAVANJA, OPIS

Objašnjenje	Kod	Izlaz
Broj različitih vrednosti po kolonama (axis=0) ili vrstama (axis=1)	<pre>print(d.nunique(axis=0).values)</pre>	<pre>0 3 1 3 2 3 dtype: int64</pre>
Broj jedinstvenih redova u DataFrame-u	<pre>print(d.value_counts())</pre>	<pre>0 1 2 6 7 8 1 3 4 5 1 1 2 0 1 dtype: int64</pre>
Broj jedinstvenih redova u DataFrame-u koristeći podskup kolona	<pre>print(d.value_counts(subset=[0]))</pre>	<pre>6 1 3 1 1 1 dtype: int64</pre>
Standardno odstupanje po kolonama (axis=0) ili vrstama (axis=1)	<pre>print(d.std(axis=0))</pre>	<pre>0 2.516611 1 2.516611 2 4.041452 dtype: float64</pre>
Varijansa po kolonama (axis=0) ili vrstama (axis=1)	<pre>print(d.var(axis=0))</pre>	<pre>0 6.333333 1 6.333333 2 16.333333 dtype: float64</pre>

DATAFRAME - SELEKCIJA

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>d = pd.DataFrame(data=[[1, 2, 0], [1, 2, 0], [3, 4, 5]])</pre>	
Kreiranje DataFrame-a od drugog izbacivanjem zadatih labela vrsta (index) ili kolona (columns)	<pre>print(d.drop(index=[0], columns=[1]))</pre>	<pre>0 2 1 1 0 2 3 5</pre>
Kreiranje DataFrame-a izbacivanjem duplikata redova (ostaje poslednja pojava)	<pre>print(d.drop_duplicates(keep='last'))</pre>	<pre>0 1 2 1 1 2 0 2 3 4 5</pre>
Označava duplikate redove (ostaje prva pojava)	<pre>print(d.duplicated(keep='last').values)</pre>	<pre>[True False False]</pre>
Vraća prvih n vrednosti (podraz. 5). Negativno n vraća sve vrednosti osim poslednjih n.	<pre>print(d.head(n=2))</pre>	<pre>0 1 2 0 1 2 0 1 1 2 0</pre>
Vraća poslednjih n vrednosti (podraz. 5). Negativno n vraća sve vrednosti osim prvih n.	<pre>print(d.tail(n=2))</pre>	<pre>0 1 2 1 1 2 0 2 3 4 5</pre>

DATAFRAME - SELEKCIJA

Objašnjenje	Kod	Izlaz
Indeks najvećeg elementa po kolonama (axis=0)	<pre>print(s.idxmax(axis=0))</pre>	<pre>0 2 1 2 2 2 dtype: int64</pre>
Indeks najmanjeg elementa po vrstama (axis=1)	<pre>print(s.idxmin(axis=1))</pre>	<pre>0 2 1 2 2 0 dtype: int64</pre>
Podskup kolona (axis=1) ili vrsta (axis=0), čije su labele navedene u items	<pre>print(d.filter(items=[0, 2], axis=1))</pre>	<pre> 0 2 0 1 0 1 1 0 2 3 0</pre>
Reindeksiranje vrsta DataFrame-a (na novim indeksima se nalazi vrednost NaN ili fill_value)	<pre>d = d.reindex(index=d.index - 1, fill_value=-1) print(d)</pre>	<pre> 0 1 2 -1 -1 -1 -1 0 1 2 0 1 1 2 0</pre>
Reindeksiranje kolona DataFrame-a (na novim indeksima se nalazi najbliža vrednost)	<pre>d = d.reindex(columns=d.columns - 1, method='nearest') print(d)</pre>	<pre> -1 0 1 0 1 1 2 1 1 1 2 2 3 3 4</pre>

DATAFRAME - SELEKCIJA, REINDEKSIRANJE

Objašnjenje	Kod	Izlaz
Očuvanje redosleda elemenata uz promenu labela vrsti (arange vraća niz vrednosti u opsegu [4, 7) sa korakom 1)	<pre>d.index = np.arange(4, 7) print(d) # isto kao i: d = d.set_axis(np.arange(4, 7), axis=0)</pre>	<pre> 0 1 2 4 1 2 0 5 1 2 0 6 3 4 5</pre>
Očuvanje redosleda elemenata uz promenu labela kolona (arange vraća niz vrednosti u opsegu [4, 7) sa korakom 1)	<pre>d.columns = np.arange(4, 7) print(d) # isto kao i: d = d.set_axis(np.arange(4, 7), axis=1)</pre>	<pre> 4 5 6 0 1 2 0 1 1 2 0 2 3 4 5</pre>
Preimenovanje labela redova Preimenovanje imena ose	<pre>d = d.rename(index=lambda x: x + 2) d = d.rename_axis('ime', axis=0) print(d)</pre>	<pre> 0 1 2 ime 2 1 2 0 3 1 2 0 4 3 4 5</pre>
Preimenovanje labela kolona Preimenovanje imena ose	<pre>d = d.rename(columns=lambda x: x + 2) d = d.rename_axis('ime', axis=1) print(d)</pre>	<pre>ime 2 3 4 0 1 2 0 1 1 2 0 2 3 4 5</pre>

DATAFRAME - SELEKCIJA, REINDEKSIRANJE

Objašnjenje	Kod	Izlaz
Uzorkovanje podataka (redova) n - broj podataka ili frac - procenat podataka random_state - za reproducibilne situacije axis - 0 za vrste, 1 za kolone	<pre>sam = d.sample(n=1, random_state=15, axis=0) # isto kao i: sam = d.sample(frac=0.33, random_state=15) print(sam)</pre>	<pre>0 1 2 2 3 4 5</pre>
Dohvatanje elemenata serije sa pozicija redova (axis=0) ili kolona (axis=1)	<pre>d = d.take([1, 2], axis=1) # isto kao i: d = d.filter([1, 2], axis=1) print(d)</pre>	<pre>1 2 0 2 0 1 2 0 2 4 5</pre>
Odsecanje redova (axis=0) ili kolona (axis=1) DataFrame-a pre i posle zadatih labela	<pre>d = d.truncate(before=1, after=2, axis=0) print(d)</pre>	<pre>0 1 2 1 1 2 0 2 3 4 5</pre>
Dodavanje prefiksa i sufiksa labelama kolona	<pre>d = d.add_prefix('c') d = d.add_suffix('s') print(d)</pre>	<pre> c0s c1s c2s 0 1 2 0 1 1 2 0 2 3 4 5</pre>

DATAFRAME - UPRAVLJANJE NaN VRED.

Objašnjenje	Kod	Izlaz
Kreiranje DataFrame-a	<pre>d = pd.DataFrame(data=[[1, 2], [np.nan, 2], [3, np.nan]])</pre>	
Popunjavanje NaN vrednosti (sinonim za s.bfill()) Koristi prvu sledeću vrednost u koloni (axis=0) ili vrsti (axis=1) različitu od NaN za popunu NaN vrednosti	<pre>d = d.backfill(axis=0) # isto kao i: d = d.fillna(method='backfill', axis=0) # isto kao i: d = d.replace(np.nan, method='bfill') print(d)</pre>	<pre> 0 1 0 1.0 2.0 1 3.0 2.0 2 3.0 NaN</pre>
Eliminacija svih redova (axis=0) ili kolona (axis=1) koji imaju sve (all) ili barem jednu (any) NaN vrednost	<pre>print(d.dropna(axis=0, how='any')) # za izbacivanje redova samo jedne kolone "column": print(d.dropna(axis=0, subsets=[column]))</pre>	<pre> 0 1 0 1.0 2.0</pre>
Popunjavanje NaN vrednosti (sinonim za s.pad()) Koristi prvu prethodnu vrednost u koloni (axis=0) ili vrsti (axis=1) različitu od NaN za popunu NaN vrednosti	<pre>d = d.ffill(axis=0) # isto kao i: d = d.fillna(method='ffill', axis=0) # isto kao i: d = d.replace(np.nan, method='ffill') print(d)</pre>	<pre> 0 1 0 1.0 2.0 1 1.0 2.0 2 3.0 2.0</pre>

SERIES - UPRAVLJANJE NaN VREDNOSTIMA

Objašnjenje	Kod	Izlaz
Popunjavanje NaN vrednosti određenom vrednošću	<pre>print(d.fillna(0))</pre>	<pre>0 1 0 1.0 2.0 1 0.0 2.0 2 3.0 0.0</pre>
Popunjavanje NaN vrednosti koristeći linearnu interpolaciju	<pre>print(d.interpolate())</pre>	<pre>0 1 0 1.0 2.0 1 2.0 2.0 2 3.0 2.0</pre>
Detektovanje NaN vrednosti	<pre>print(d.isna()) # isto kao i: print(d.isnull())</pre>	<pre>0 1 0 False False 1 True False 2 False True</pre>
Detektovanje vrednosti koje nisu NaN	<pre>print(d.notna()) # isto kao i: print(d.notnull())</pre>	<pre>0 1 0 True True 1 False True 2 True False</pre>
Zamena vrednosti vrednošću	<pre>print(d.replace(np.nan, 0))</pre>	<pre>0 1 0 1.0 2.0 1 0.0 2.0 2 3.0 0.0</pre>

DATAFRAME - UPRAVLJANJE NaN VRED.

Objašnjenje	Kod	Izlaz
Zamena liste vrednosti vrednošću	<pre>print(d.replace(to_replace=[np.nan, 2], value=0))</pre>	<pre>0 1 0 1.0 0.0 1 0.0 0.0 2 3.0 0.0</pre>
Zamena liste vrednosti listom vrednosti	<pre>print(d.replace(to_replace=[np.nan, 2], value=[0, 1]))</pre>	<pre>0 1 0 1.0 1.0 1 0.0 1.0 2 3.0 0.0</pre>
Zamena parova vrednosti (vrednost: zamena)	<pre>print(d.replace(to_replace={np.nan: 2, 2: 1}))</pre>	<pre>0 1 0 1.0 1.0 1 2.0 1.0 2 3.0 2.0</pre>
Zamena vrednosti u kolonama određenom vrednošću (NaN u koloni labele 0 i 2 u koloni labele 1 vrednošću 1)	<pre>print(d.replace(to_replace={0: np.nan, 1: 2}, value=1))</pre>	<pre>0 1 0 1.0 1.0 1 1.0 1.0 2 3.0 NaN</pre>
Zamena vrednosti u kolonama određenim vrednostima (NaN u koloni labele 0 i 2 u koloni labele 1 vrednošću 1 u koloni 0 i 0 u koloni 1)	<pre>print(d.replace(to_replace={0: np.nan, 1: 2}, value={0: 1, 1: 0}))</pre>	<pre>0 1 0 1.0 0.0 1 1.0 0.0 2 3.0 NaN</pre>

DATAFRAME – OBLIKOVANJE, SORTIRANJE

Objašnjenje	Kod	Izlaz
Kreiranje serije	<pre>d = pd.DataFrame(data=[[3, 1], [1, 2], [3, 3]])</pre>	
Sortiranje vrsta prema vrednosti elemenata po navedenim kolonama (axis=0)	<pre>print(d.sort_values(by=[0, 1], axis=0, ascending=True))</pre>	<pre>0 1 1 1 2 0 3 1 2 3 3</pre>
Sortiranje kolona prema vrednosti elemenata po navedenim vrstama (axis=1)	<pre>print(d.sort_values(by=[2, 0], axis=1, ascending=True))</pre>	<pre>1 0 0 1 3 1 2 1 2 3 3</pre>
Sortiranje vrsta prema vrednosti labela koristeći ključ za sortiranje	<pre>print(d.sort_index(axis=0, ascending=True, key=lambda x: x % 2 == 1))</pre>	<pre>0 1 0 3 1 2 3 3 1 1 2</pre>
Sortiranje kolona prema vrednosti labela	<pre>print(d.sort_index(axis=1, ascending=False))</pre>	<pre>1 0 0 1 3 1 2 1 2 3 3</pre>

DATAFRAME - OBLIKOVANJE, SORTIRANJE

Objašnjenje	Kod	Izlaz
Dohvatanje n redova sa najvećim vrednostima po kolonama navedenim u columns	<pre>print(d.nlargest(n=2, columns=[0], keep='first'))</pre>	<pre>0 1 0 3 1 2 3 3</pre>
Dohvatanje n redova sa najmanjim vrednostima po kolonama navedenim u columns	<pre>print(d.nsmallest(n=2, columns=[0], keep='first'))</pre>	<pre>0 1 1 1 2 0 3 1</pre>
Transponovani DataFrame (svojstvo T)	<pre>print(d.T) # isto kao i: print(d.transpose())</pre>	<pre>0 1 2 0 3 1 3 1 1 2 3</pre>
Transformisanje elemenata DataFrame-a koji su lista uz reindeksiranje	<pre>d = pd.DataFrame(data=[[1, 2], 1], [3, 4]) print(d.explode(column=0, ignore_index=True))</pre>	<pre>0 1 0 1 1 1 2 1 2 3 4</pre>
Kreiranje skalara na osnovu DataFrame-a sa samo jednim elementom. Ukoliko DataFrame ima više elemenata operacija je bez efekta.	<pre>d = pd.DataFrame(data=[1, 2]) print(d.squeeze()) d = pd.DataFrame(data=[1]) print(d.squeeze())</pre>	<pre>0 1 1 2 Name: 0, dtype: int64 1</pre>

SERIES - OBLIKOVANJE, SORTIRANJE

Objašnjenje	Kod	Izlaz
Kreiranje multi-indeksa DataFrame-a prilikom kreiranja DataFrame-a	<pre>d=pd.DataFrame(index=pd.MultiIndex.from_tuples(((1, 2), (1, 3))), data=[[1, 2], [3, 4]])</pre>	<pre> 0 1 1 2 1 2 3 3 4</pre>
Pristup elementu DataFrame-a preko multi-indeksa	<pre>print(d.loc[(1, 3), 0])</pre>	<pre>3</pre>
Obrtanje redosleva nivoa multi-indeksa	<pre>print(d.reorder_levels(order=[1, 0], axis=0))</pre>	<pre> 0 1 2 1 1 2 3 1 3 4</pre>

DATAFRAME - KOMBINOVANJE, POREĐENJE

Objašnjenje	Kod	Izlaz
Kreiranje DataFrame-ova	<pre>d1 = pd.DataFrame(data=[[1, 2, 3], [4, 5, 6]]) d2 = pd.DataFrame(data=[[7, 8, 9, 0]])</pre>	
Dodavanje redova jednog DataFrame-a na kraj drugog (nove kolone dodavanog DataFrame se dodaju)	<pre>print(d1.append(d2, ignore_index=True))</pre>	<pre> 0 1 2 3 0 1 2 3 NaN 1 4 5 6 NaN 2 7 8 9 0.0</pre>
Dodavanje novih kolona u DataFrame (ime nove kolone je argument)	<pre>print(d1.assign(p1=lambda df: df[0] * 2 + 1))</pre>	<pre> 0 1 2 p1 0 1 2 3 3 1 4 5 6 9</pre>
Poređenje dva DataFrame-a uz prikaz razlika (mogu se porediti samo identično labelirani DataFrame-ovi)	<pre>t = d1.copy() t.loc[0, 0] = 0 t.loc[1, 2] = 12 print(d1.compare(t))</pre>	<pre> 0 2 self other self other 0 1.0 0.0 NaN NaN 1 NaN NaN 6.0 12.0</pre>
Unija kolona 2 DataFrame-a rsuffix - sufiks kolona koje se preklapaju how - koristi se indeks pozivaoca (left) ili argumenta (right)	<pre>print(d1.join(d2, rsuffix='_', how='left'))</pre>	<pre> 0 1 2 0_ 1_ 2_ 3 0 1 2 3 7.0 8.0 9.0 0.0 1 4 5 6 NaN NaN NaN NaN</pre>

DATAFRAME - KOMBINOVANJE, POREĐENJE

Objašnjenje	Kod	Izlaz
Kreiranje DataFrame-ova	<pre>d1 = pd.DataFrame(data=[[1, 2], [4, 5]]) d2 = pd.DataFrame(data=[[1, 2, 3]])</pre>	
Merge dva DataFrame-a (koriste se labela vrsta pozivaoca)	<pre>print(d1.merge(d2, how='left'))</pre>	<pre> 0 1 2 0 0 1 2 3.0 1 4 5 NaN</pre>
Merge dva DataFrame-a (koriste se labela vrsta pozivanog)	<pre>print(d1.merge(d2, how='right'))</pre>	<pre> 0 1 2 0 0 1 2 3</pre>
Merge dva DataFrame-a (koristi se unija labela vrsta)	<pre>print(d1.merge(d2, how='outer'))</pre>	<pre> 0 1 2 0 0 1 2 3.0 1 4 5 NaN</pre>
Merge dva DataFrame-a (koristi se presek labela vrsta)	<pre>print(d1.merge(d2, how='inner'))</pre>	<pre> 0 1 2 0 0 1 2 3</pre>
Merge dva DataFrame-a (koriste se labela vrsta pozivaoca, merge nad kolonom 0)	<pre>print(d1.merge(d2, how='left', on=[0]))</pre>	<pre> 0 1_x 1_y 2 0 0 1 2 2.0 3.0 1 4 5 NaN NaN</pre>

DATAFRAME - KOMBINOVANJE, POREĐENJE

Objašnjenje	Kod	Izlaz
Kreiranje DataFrame-ova	<pre>d1 = pd.DataFrame(data=[[1, 2], [4, 5]]) d2 = pd.DataFrame(data=[[2, 7, 3]])</pre>	
Ažuriranje vrednosti DataFrame-a pozivaoca vrednostima DataFrame argumenta	<pre>d1.update(d2) print(d1)</pre>	<pre> 0 1 0 2.0 7.0 1 4.0 5.0</pre>

Dodatni tutorijali:

- 10 minute pandas:

https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html

- Getting started tutorial:

https://pandas.pydata.org/pandas-docs/stable/getting_started/intro_tutorials/index.html

- Vizuelizacija:

https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

PITANJA?

<http://ri4es.etf.rs/>

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.