

Основне академске студије  
Информациони инжењеринг

Основи рачунарске интелигенције

# Рад с подацима у језику Python

(материјали за вежбе)

## Библиотека *pandas* (*Python Data Analysis Library*)

посвећена раду над подацима

подршка за једнодимензионалне и дводимензионалне низове података

ослањање на библиотеку *NumPy*

подршка за разне операције над подацима

## Библиотека *pandas*

### ОСНОВНИ ПОДАЦИ

водећи аутор Вес Макини

настанак 2008. и 2009. године

слободна за употребу

актуелна верзија 1.4.1

Интернет сајт

<https://pandas.pydata.org/>

## Припреме за коришћење

### инсталација библиотеке

библиотека доступна у оквиру дистрибуције *Anaconda*

могућа засебна инсталација

помоћу *pip*

помоћу *conda*

### учитавање библиотеке

```
import pandas as pd
```

коришћење псеудонима **pd**

## Главне структуре података

### серија

једнодимензионални низ елемената

слична низу из библиотеке *NumPy*

слична речнику уграђеном у језику *Python*

поседује придружене ознаке елемената (индекс)

### скуп података (оквир података)

дводимензионална структура

табела података

може поседовати придружене ознаке редова (индекс) и ознаке колоне

## Главне структуре података – основне радње

### формирање

серија

функција **Series(...)**

скуп података

функција **DataFrame(...)**

### приступање

оператор **[]**

### приступање по вредности кључа

функција **get(...)**

## Главне структуре података – основне радње

### измена

функција **update(...)**

функција **assign(...)**

скуп података

додавање колоне

функција **insert(...)**

скуп података

додавање колоне на дату позицију

### уклањање елемената

функција **drop(...)**

## Главне структуре података – особине

### особина **index**

серија

ознаке елемената

скуп података

ознаке редова

### особина **dtype**

серија

тип елемената

### особина **dtypes**

серија

тип елемената

скуп података

типови колона

### особина **array**

серија

садржај

## Главне структуре података – особине

особина **at**

приступање појединачном елементу по ознаци

особина **iat**

приступање појединачном елементу по позицији

особина **loc**

приступање по ознаци или путем логичких вредности

особина **iloc**

приступање по позицији

## Главне структуре података – особине

особина **empty**

показатељ да ли је структура података празна

особина **shape**

облик

особина **ndim**

број димензија

особина **size**

број елемената

...

## Серија – формирање

```
pd.Series({10: "F2", 20: "F1", 30: "F3"})
```

```
10    F2
```

```
20    F1
```

```
30    F3
```

```
dtype: object
```

```
pd.Series([1, 6, 2], index=[1, 2, 3])
```

```
1     1
```

```
2     6
```

```
3     2
```

```
dtype: int64
```

## Серија – формирање

```
pd.Series("x", index=[1, 2, 3])
```

```
1    x
```

```
2    x
```

```
3    x
```

```
dtype: object
```

```
import numpy as np
```

```
pd.Series(np.array([2.1, 4.3, 6.5, 8.7]))
```

```
0    2.1
```

```
1    4.3
```

```
2    6.5
```

```
3    8.7
```

```
dtype: float64
```

## Серија – приступање

```
sa = pd.Series({10: "F2", 20: "F1", 30: "F3"})
```

```
10      F2
```

```
20      F1
```

```
30      F3
```

```
dtype: object
```

```
sa[20]
```

```
F1
```

```
sa.get(40)
```

```
None
```

## Серија – приступање

```
sb = pd.Series(np.array([2.1, 4.3, 6.5, 8.7]))
```

```
0    2.1
```

```
1    4.3
```

```
2    6.5
```

```
3    8.7
```

```
dtype: float64
```

```
sb[1:4:2]
```

```
1    4.3
```

```
3    8.7
```

```
dtype: float64
```

## Серија – измена

```
sc = pd.Series(np.array([5, 2, 1]))
```

```
0    5
```

```
1    2
```

```
2    1
```

```
dtype: int64
```

```
sc.update(pd.Series(np.array([4, 6]),  
                    index=[1, 2]))
```

```
0    5
```

```
1    4
```

```
2    6
```

```
dtype: int64
```

## Серија – уклањање

```
sd = pd.Series(np.array([75, 3, 25, 22]),  
               index=["S", "P", "T", "F"])
```

```
S      75
```

```
P       3
```

```
T      25
```

```
F      22
```

```
dtype: int64
```

```
sd.drop(labels=["F", "S"], inplace=True)
```

```
P       3
```

```
T      25
```

```
dtype: int64
```

Серија – коришћење особина

```
se = pd.Series({1: "A", 3: "CDE", 5: "EFGHI"})
```

```
1          A
```

```
3         CDE
```

```
5        EFGHI
```

```
dtype: object
```

```
se.index
```

```
Int64Index([1, 3, 5], dtype='int64')
```

Серија – коришћење особина

**se.dtype**

object

**se.array**

<PandasArray>

['A', 'CDE', 'EFGHI']

Length: 3, dtype: object

Серија – коришћење особина

```
se.loc[1]
```

```
A
```

```
se.loc[1:3]
```

```
1      A
```

```
3     CDE
```

```
dtype: object
```

Серија – коришћење особина

```
se.iloc[1]
```

```
CDE
```

```
se.iloc[1:3]
```

```
3      CDE
```

```
5     EFGHI
```

```
dtype: object
```

Серија – коришћење особина

**se.empty**

False

**se.shape**

(3,)

**se.ndim**

1

**se.size**

3

## Скуп података – формирање

```
ра = pd.DataFrame(  
    {"ka": [100, 200, 300],  
     "kb": [6.1, 1.4, 3.5]})
```

	ka	kb
0	100	6.1
1	200	1.4
2	300	3.5

## Скуп података – формирање

```
pb = pd.DataFrame(  
    [{"k1": "g", "k2": True},  
     {"k1": "j", "k2": False},  
     {"k1": "t", "k2": True}])
```

	k1	k2
0	g	True
1	j	False
2	t	True

## Скуп података – формирање

```
pc = pd.DataFrame(  
    {"ka": pd.Series(["p", "s", "r"]),  
     "kb": pd.Series([5, 2, 7])})
```

	ka	kb
0	p	5
1	s	2
2	r	7

## Скуп података – формирање

```
pd = pd.DataFrame(  
    {"ka": ["u", "x", "h", "e"],  
     "kb": [6, 2, 3, 5],  
     "kc": [8.3, 3.2, 5.4, 9.8]},  
    index=["ra", "rb", "rc", "rd"])
```

	ka	kb	kc
ra	u	6	8.3
rb	x	2	3.2
rc	h	3	5.4
rd	e	5	9.8

## Скуп података – приступање

```
pd["kc"]
```

```
ra      8.3
```

```
rb      3.2
```

```
rc      5.4
```

```
rd      9.8
```

```
Name: kc, dtype: float64
```

```
pd[["kc", "kb"]]
```

```
      kc  kb
```

```
ra  8.3  6
```

```
rb  3.2  2
```

```
rc  5.4  3
```

```
rd  9.8  5
```

## Скуп података – приступање

**pd[0:2]**

	ka	kb	kc
ra	u	6	8.3
rb	x	2	3.2

**pd[0::2]**

	ka	kb	kc
ra	u	6	8.3
rc	h	3	5.4

## Скуп података – приступање

```
pd[pd["ka"] != "h"]
```

	ka	kb	kc
ra	u	6	8.3
rb	x	2	3.2
rd	e	5	9.8

```
pd.get(["ka", "kb"])
```

	ka	kb
ra	u	6
rb	x	2
rc	h	3
rd	e	5

## Скуп података – мењање

```
pd.at["rb", "ka"] = "t"
```

```
pd.iat[0, 2] = 9.1
```

	ka	kb	kc
ra	u	6	9.1
rb	t	2	3.2
rc	h	3	5.4
rd	e	5	9.8

Скуп података – мењање

```
pd["kd"] = "d"
```

```
pd.insert(0, "k", 100)
```

	k	ka	kb	kc	kd
ra	100	u	6	9.1	d
rb	100	t	2	3.2	d
rc	100	h	3	5.4	d
rd	100	e	5	9.8	d

Скуп података – уклањање

```
pd = pd.drop(["rc"])  
pd.drop(columns=["k"], inplace=True)
```

	ka	kb	kc	kd
ra	u	6	9.1	d
rb	t	2	3.2	d
rd	e	5	9.8	d

Скуп података – коришћење особина

```
pd.loc[["rb", "ra"], ["kc", "ka"]]
```

	kc	ka
rb	3.2	t
ra	9.1	u

```
pd.loc[pd["kb"] > 2.7, ["kc", "kb"]]
```

	kc	kb
ra	9.1	6
rd	9.8	5

Скуп података – коришћење особина

```
pd.iloc[1:3, 1]
```

```
rb      2
```

```
rd      5
```

```
Name: kb, dtype: int64
```

Скуп података – коришћење особина

```
pd.loc["rd", "kd"] = "a"
```

```
pd.loc["ra":"rb", "kb"] = 7
```

	ka	kb	kc	kd
ra	u	7	9.1	d
rb	t	7	3.2	d
rd	e	5	9.8	a

Скуп података – коришћење особина

**pd.dtypes**

ka object

kb int64

kc float64

kd object

dtype: object

## Скуп података – коришћење особина

**pd.empty**

False

**pd.shape**

(3, 4)

**pd.ndim**

2

**pd.size**

12

## Главне структуре података – додатне радње

провера заступљеност логичких вредности

функција **any(...)**

провера да ли је барем нека вредност истинита

функција **all(...)**

провера да ли су све вредности истините

издвајање карактеристичних делова

функција **head(...)**

издвајање почетних делова

функција **tail(...)**

издвајање крајњих делова

## Главне структуре података – додатне радње

рад с недостајућим вредностима

функција **dropna(...)**

уклањање недостајућих вредности

функција **fillna(...)**

замена недостајућих вредности

функција **isna(...)**

провера присуства недостајућих вредности

функција **isnull(...)**

провера присуства недостајућих вредности

## Главне структуре података – додатне радње математичке и статистичке радње

функција **count(...)**

пребројавање елемената

функција **describe(...)**

израчунавање дескриптивних статистика

функција **mean(...)**

израчунавање аритметичке средине

функција **median(...)**

израчунавање медијане

функција **rank(...)**

израчунавање бројчаних рангова

функција **sum(...)**

израчунавање збира елемената

...

## Главне структуре података – додатне радње

разне радње

функција **agg(...)**

агрегирање података путем одабраних радњи

функција **apply(...)**

извршавање одабране функције над деловима структуре података

функција **combine(...)**

спајање структура података

функција **concat(...)**

спајање структура података

функција **groupby(...)**

груписање података

функција **merge(...)**

спајање структура података по узору на спајање у релационим базама података

функција **transform(...)**

извршавање одабране функције над структуром података уз задржавање облика

## Подаци

скуп података о притужбама на авионску буку за подручје Сан Франциска (САД)

постоји 4749 појава и 5 обележја (година, месец, заједница, укупан број притужби и укупан број позивалаца)

свака појава обухвата збирне податке о броју притужби и броју позивалаца за један месец једне године у једној заједници

извор скупа података

скуп података *Aircraft Noise Complaint Data*

матична Интернет страна

<https://data.sfgov.org/Transportation/Aircraft-Noise-Complaint-Data/q3xd-hfi8>

у оквиру портала отворених података Сан Франциска (САД)

подаци доступни у више формата

може се користити формат *CSV* (*CSV* датотека доступна кроз опцију *Export*)

скуп података креиран 9. 4. 2016. а најновије повезано ажурирање 19. 12. 2020.

за одељење које је објавило скуп података назначен је *Airport (SFO)*

за власника скупа података назначен је *NoiseAbatementOffice@flysfo.com*

лиценца и могућности коришћења скупа података

*Open Data Commons Public Domain Dedication and License*

<http://opendatacommons.org/licenses/pddl/1.0/>

## Задатак 1.

Учитати скуп података из одговарајуће *CSV* датотеке!  
Проверити садржај учитаних података и типове колона!

## Задатак 2.

Утврдити број редова у учитаном скупу података!

## Задатак 3.

Утврдити да ли постоје недостајуће вредности у скупу података и, ако да, у којим колонама! У случају да недостајуће вредности постоје, уклонити из скупа података све редове у којима су такве вредности присутне и утврдити који је нови број редова!

Задатак 4.

Утврдити колико се различитих заједница помиње у скупу података!

Задатак 5.

Утврдити који је најмањи, просечни и највећи број притужби на буку у једном месецу! Просечну вредност по потреби заокружити на две децимале!

Задатак 6.

Пронаћи податке за случај највећег броја притужби на буку у једном месецу!

## Задатак 7.

Израчунати дескриптивне статистике за број притужби у месецу за заједницу Пало Алто!

## Задатак 8.

Израчунати просечан број позивалаца у месецу по појединачним заједницама!

## Задатак 9.

Издвојити подскуп података који се односи само на случајеве из 2018. године и проверити његов садржај!

## Задатак 10.

Из подскупа података за 2018. годину уклонити колону о заједници и након тога проверити које су колоне преостале у том подскупу података!

## Задатак 11.

У подскупу података за 2018. годину изменити садржај колоне о месецима тако да уместо броја буде коришћен скраћени назив месеца! Након измене, проверити типове колоне и садржај подскупа података!

## Задатак 12.

Над подскупом података за 2018. годину извршити агрегирање тако да за сваки месец те године буду доступни збирни подаци израчунати на нивоу свих заједница!

## Основна литература

pandas. Python data analysis library. Internet:  
<https://pandas.pydata.org/>

pandas. pandas documentation. Internet:  
<https://pandas.pydata.org/docs/>