

08_missing_values

August 1, 2025

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]: penzici = pd.read_excel("../..../datasets/penzioneri.xlsx", sheet_name = "drugi")
```

```
[3]: penzici.head(10)
```

```
[3]:
```

	grad	starost	zanimanje	spol	primanja	stednja
0	Split	55.0	obrtnik	m	2500.0	10000.0
1	Zagreb	77.0	penzioner	m	1700.0	5000.0
2	Split	62.0	penzioner	m	1800.0	2000.0
3	Zagreb	NaN	obrtnik	f	3500.0	4500.0
4	Split	59.0	NaN	NaN	NaN	NaN
5	Zagreb	71.0	penzioner	m	1250.0	6500.0
6	Osijek	59.0	obrtnik	m	4000.0	12000.0
7	Pula	NaN	obrtnik	NaN	3000.0	10000.0
8	Zagreb	65.0	NaN	m	2000.0	7000.0
9	Split	52.0	obrtnik	m	5000.0	NaN

1 Dataset penzioneri

Financijski podaci o starijim građanima koje skuplja jedna financijska institucija

Pregledom podataka utvrđuje se da postoje neki podaci koji su označeni oznakom NaN

Radi se o missing vrijednostima, podacima koji jednostavno nedostaju

Možda je došlo do greške pri unosu podataka, možda podaci nikad nisu bili ni zapisani, možda su se zagubili pri migracijama podataka

```
[4]: penzici.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   grad        21 non-null    object
1   starost     18 non-null    float64
```

```

2   zanimanje  19 non-null    object
3   spol       19 non-null    object
4   primanja   20 non-null    float64
5   stednja    18 non-null    float64
dtypes: float64(3), object(3)
memory usage: 1.1+ KB

```

```
[5]: penzici.describe()
```

```

[5]:
count    starost    primanja    stednja
mean    67.111111  3882.500000  11227.777778
std     10.681974  1823.622407  8040.105761
min     52.000000  1250.000000  2000.000000
25%     59.000000  2375.000000  6275.000000
50%     63.500000  3750.000000  9600.000000
75%     74.750000  4925.000000  13500.000000
max     92.000000  8200.000000  33000.000000

```

```
[6]: penzici.isna().sum()
```

```

[6]: grad      0
     starost    3
     zanimanje  2
     spol       2
     primanja   1
     stednja    3
     dtype: int64

```

`isna().sum()` je naredba koja sumira koliko ima missing vrijednosti po stupcu

```
[7]: isna_values = penzici[penzici.isna().any(axis=1)]
```

```
[8]: isna_values
```

```

[8]:
   grad  starost  zanimanje  spol  primanja  stednja
3  Zagreb     NaN  obrtnik    f    3500.0   4500.0
4  Split     59.0     NaN    NaN     NaN     NaN
7   Pula     NaN  obrtnik    NaN    3000.0  10000.0
8  Zagreb     65.0     NaN    m    2000.0   7000.0
9  Split     52.0  obrtnik    m    5000.0     NaN
11 Pula     NaN  obrtnik    m    4500.0     NaN

```

```
[9]: penzici2 = penzici.copy()
```

Pokazat ćemo nekoliko načina rješavanja missing vrijednosti. Kako bismo sačuvali originalni dataset, radimo novu kopiju Prvi način, kao i kod outliera je brisanje

```
[10]: penzici2.dropna(how = "any")
```

```
[10]:      grad  starost  zanimanje spol  primanja  stednja
0   Split    55.0    obrtnik    m    2500.0  10000.0
1   Zagreb    77.0  penzioner    m    1700.0   5000.0
2   Split    62.0  penzioner    m    1800.0   2000.0
5   Zagreb    71.0  penzioner    m    1250.0   6500.0
6   Osijek    59.0    obrtnik    m    4000.0  12000.0
10  Zagreb    60.0  penzioner    f    1700.0   2500.0
12  Osijek    58.0    obrtnik    f    3000.0  11000.0
13  Zagreb    83.0  penzioner    m    3500.0  14000.0
14  Split    74.0    obrtnik    f    4900.0  16000.0
15  Zagreb    92.0  penzioner    m    6000.0  20000.0
16  Osijek    62.0    obrtnik    m    5200.0   6200.0
17   Pula     69.0    obrtnik    m    4700.0   9200.0
18  Osijek    76.0    obrtnik    m    4900.0   8200.0
19  Zagreb    59.0    obrtnik    f    6300.0  25000.0
20  Zagreb    75.0  penzioner    m    8200.0  33000.0
```

Kod naredbe `dropna()` mora se koristiti parametar `inplace = True` ako zaista želimo obrisati missing values. Ako nismo koristili `inplace = True`, dobili smo samo klasični upit koji bi nam pokazao kako bi izgledao dataset bez missing vrijednosti.

```
[11]: penzici2.head(10)
```

```
[11]:      grad  starost  zanimanje spol  primanja  stednja
0   Split    55.0    obrtnik    m    2500.0  10000.0
1   Zagreb    77.0  penzioner    m    1700.0   5000.0
2   Split    62.0  penzioner    m    1800.0   2000.0
3   Zagreb    NaN    obrtnik    f    3500.0   4500.0
4   Split    59.0      NaN  NaN      NaN      NaN
5   Zagreb    71.0  penzioner    m    1250.0   6500.0
6   Osijek    59.0    obrtnik    m    4000.0  12000.0
7   Pula     NaN    obrtnik  NaN    3000.0  10000.0
8   Zagreb    65.0      NaN    m    2000.0   7000.0
9   Split    52.0    obrtnik    m    5000.0      NaN
```

Redovi 3, 4, 7 i 8 su opet tu.

```
[12]: penzici2.dropna(how = "any", inplace = True)
```

```
[13]: penzici2.head(10)
```

```
[13]:      grad  starost  zanimanje spol  primanja  stednja
0   Split    55.0    obrtnik    m    2500.0  10000.0
1   Zagreb    77.0  penzioner    m    1700.0   5000.0
2   Split    62.0  penzioner    m    1800.0   2000.0
5   Zagreb    71.0  penzioner    m    1250.0   6500.0
6   Osijek    59.0    obrtnik    m    4000.0  12000.0
10  Zagreb    60.0  penzioner    f    1700.0   2500.0
```

12	Osijek	58.0	obrtnik	f	3000.0	11000.0
13	Zagreb	83.0	penzioner	m	3500.0	14000.0
14	Split	74.0	obrtnik	f	4900.0	16000.0
15	Zagreb	92.0	penzioner	m	6000.0	20000.0

Problem s brisanjem redaka koji imaju missing value je taj što možete obrisati previše

Možda želimo obrisati redak 4, korisnik iz Splita jer u njemu ima previše nedostajućih vrijednosti, ostale ćemo zadržati

Kadkada je jako teško doći do podataka, naročito recimo u medicini - ne možete samo izaći na ulicu i anketom prikupiti podatke o osobama koje boluju od karcinoma

```
[14]: penzici3 = penzici.copy()
```

```
[15]: penzici3.head(10)
```

```
[15]:
```

	grad	starost	zanimanje	spol	primanja	stednja
0	Split	55.0	obrtnik	m	2500.0	10000.0
1	Zagreb	77.0	penzioner	m	1700.0	5000.0
2	Split	62.0	penzioner	m	1800.0	2000.0
3	Zagreb	NaN	obrtnik	f	3500.0	4500.0
4	Split	59.0	NaN	NaN	NaN	NaN
5	Zagreb	71.0	penzioner	m	1250.0	6500.0
6	Osijek	59.0	obrtnik	m	4000.0	12000.0
7	Pula	NaN	obrtnik	NaN	3000.0	10000.0
8	Zagreb	65.0	NaN	m	2000.0	7000.0
9	Split	52.0	obrtnik	m	5000.0	NaN

```
[16]: penzici3
```

```
[16]:
```

	grad	starost	zanimanje	spol	primanja	stednja
0	Split	55.0	obrtnik	m	2500.0	10000.0
1	Zagreb	77.0	penzioner	m	1700.0	5000.0
2	Split	62.0	penzioner	m	1800.0	2000.0
3	Zagreb	NaN	obrtnik	f	3500.0	4500.0
4	Split	59.0	NaN	NaN	NaN	NaN
5	Zagreb	71.0	penzioner	m	1250.0	6500.0
6	Osijek	59.0	obrtnik	m	4000.0	12000.0
7	Pula	NaN	obrtnik	NaN	3000.0	10000.0
8	Zagreb	65.0	NaN	m	2000.0	7000.0
9	Split	52.0	obrtnik	m	5000.0	NaN
10	Zagreb	60.0	penzioner	f	1700.0	2500.0
11	Pula	NaN	obrtnik	m	4500.0	NaN
12	Osijek	58.0	obrtnik	f	3000.0	11000.0
13	Zagreb	83.0	penzioner	m	3500.0	14000.0
14	Split	74.0	obrtnik	f	4900.0	16000.0
15	Zagreb	92.0	penzioner	m	6000.0	20000.0
16	Osijek	62.0	obrtnik	m	5200.0	6200.0

17	Pula	69.0	obrtnik	m	4700.0	9200.0
18	Osijek	76.0	obrtnik	m	4900.0	8200.0
19	Zagreb	59.0	obrtnik	f	6300.0	25000.0
20	Zagreb	75.0	penzioner	m	8200.0	33000.0

Uvodimo parametar thresh koji nam označava koji minimum podataka s ne NaN vrijednosću nam mora ostati

Naša tablica ima 6 kolona, ako je thresh = 5, dozvoljen je jedan NaN

Ako ima dva ili više NaNova, podatak se briše

```
[17]: penzici3.dropna(thresh = 5, inplace = True)
```

```
[18]: penzici3
```

```
[18]:
```

	grad	starost	zanimanje	spol	primanja	stednja
0	Split	55.0	obrtnik	m	2500.0	10000.0
1	Zagreb	77.0	penzioner	m	1700.0	5000.0
2	Split	62.0	penzioner	m	1800.0	2000.0
3	Zagreb	NaN	obrtnik	f	3500.0	4500.0
5	Zagreb	71.0	penzioner	m	1250.0	6500.0
6	Osijek	59.0	obrtnik	m	4000.0	12000.0
8	Zagreb	65.0	NaN	m	2000.0	7000.0
9	Split	52.0	obrtnik	m	5000.0	NaN
10	Zagreb	60.0	penzioner	f	1700.0	2500.0
12	Osijek	58.0	obrtnik	f	3000.0	11000.0
13	Zagreb	83.0	penzioner	m	3500.0	14000.0
14	Split	74.0	obrtnik	f	4900.0	16000.0
15	Zagreb	92.0	penzioner	m	6000.0	20000.0
16	Osijek	62.0	obrtnik	m	5200.0	6200.0
17	Pula	69.0	obrtnik	m	4700.0	9200.0
18	Osijek	76.0	obrtnik	m	4900.0	8200.0
19	Zagreb	59.0	obrtnik	f	6300.0	25000.0
20	Zagreb	75.0	penzioner	m	8200.0	33000.0

```
[19]: penzici4 = penzici.copy()
```

Preporuka je brisati podatke samo ako imamo 1 - 2 posto podataka s missing vrijednostima. Ako želimo sačuvati vrijednosti, možemo opet koristiti metode imputacije

Na mjesto missing numeričkih vrijednosti, stavit ćemo prosjek

```
[20]: prosjek_godina = penzici4 ["starost"].mean()
```

```
[21]: prosjek_godina
```

```
[21]: np.float64(67.11111111111111)
```

```
[22]: penzici4 ["starost"] = penzici4 ["starost"].fillna (prosjek_godina)
```

Svako prazno polje (NaN) u stupcu Starost zamjenjujemo s prosjekom stupca

Kad smo radili s putlierima, računali smo prosjek stupca bez vrijednosti redaka u kojima je outlier. Nismo htjeli da outlieri utječu na prosjek

Kad radimo s prosjekom koji želimo iskoristiti za missing vrijednosti, polja NaN se automatski isključuju

```
[23]: penzici4
```

```
[23]:
```

	grad	starost	zanimanje	spol	primanja	stednja
0	Split	55.000000	obrtnik	m	2500.0	10000.0
1	Zagreb	77.000000	penzioner	m	1700.0	5000.0
2	Split	62.000000	penzioner	m	1800.0	2000.0
3	Zagreb	67.111111	obrtnik	f	3500.0	4500.0
4	Split	59.000000	NaN	NaN	NaN	NaN
5	Zagreb	71.000000	penzioner	m	1250.0	6500.0
6	Osijek	59.000000	obrtnik	m	4000.0	12000.0
7	Pula	67.111111	obrtnik	NaN	3000.0	10000.0
8	Zagreb	65.000000	NaN	m	2000.0	7000.0
9	Split	52.000000	obrtnik	m	5000.0	NaN
10	Zagreb	60.000000	penzioner	f	1700.0	2500.0
11	Pula	67.111111	obrtnik	m	4500.0	NaN
12	Osijek	58.000000	obrtnik	f	3000.0	11000.0
13	Zagreb	83.000000	penzioner	m	3500.0	14000.0
14	Split	74.000000	obrtnik	f	4900.0	16000.0
15	Zagreb	92.000000	penzioner	m	6000.0	20000.0
16	Osijek	62.000000	obrtnik	m	5200.0	6200.0
17	Pula	69.000000	obrtnik	m	4700.0	9200.0
18	Osijek	76.000000	obrtnik	m	4900.0	8200.0
19	Zagreb	59.000000	obrtnik	f	6300.0	25000.0
20	Zagreb	75.000000	penzioner	m	8200.0	33000.0

Prosjek možemo koristiti samo za numeričke stupce, ne pomaže nam kod kategoričkih

Na kategoričkim stupcima koristimo mjeru mod (najčešća vrijednost)

```
[24]: penzici4["spol"].mode()
```

```
[24]: 0      m
      Name: spol, dtype: object
```

```
[25]: penzici4 ["spol"] = penzici4 ["spol"].fillna(value = "m")
```

```
[26]: penzici4
```

```
[26]:
```

	grad	starost	zanimanje	spol	primanja	stednja
0	Split	55.000000	obrtnik	m	2500.0	10000.0
1	Zagreb	77.000000	penzioner	m	1700.0	5000.0
2	Split	62.000000	penzioner	m	1800.0	2000.0

3	Zagreb	67.111111	obrtnik	f	3500.0	4500.0
4	Split	59.000000	NaN	m	NaN	NaN
5	Zagreb	71.000000	penzioner	m	1250.0	6500.0
6	Osijek	59.000000	obrtnik	m	4000.0	12000.0
7	Pula	67.111111	obrtnik	m	3000.0	10000.0
8	Zagreb	65.000000	NaN	m	2000.0	7000.0
9	Split	52.000000	obrtnik	m	5000.0	NaN
10	Zagreb	60.000000	penzioner	f	1700.0	2500.0
11	Pula	67.111111	obrtnik	m	4500.0	NaN
12	Osijek	58.000000	obrtnik	f	3000.0	11000.0
13	Zagreb	83.000000	penzioner	m	3500.0	14000.0
14	Split	74.000000	obrtnik	f	4900.0	16000.0
15	Zagreb	92.000000	penzioner	m	6000.0	20000.0
16	Osijek	62.000000	obrtnik	m	5200.0	6200.0
17	Pula	69.000000	obrtnik	m	4700.0	9200.0
18	Osijek	76.000000	obrtnik	m	4900.0	8200.0
19	Zagreb	59.000000	obrtnik	f	6300.0	25000.0
20	Zagreb	75.000000	penzioner	m	8200.0	33000.0

2 Zadatak 1

U stupcima zanimanje, primanja i stednja imamo missing values. Na kopiji penzici 4 riješite te probleme.

[]: