Pandas pakett: põhiline info

Pythoni keeles tarkvarateek andmete töötlemiseks ja analüüsimiseks..

Pandas on statistilise andmetöötluse pakett, mis on funktsionaalsuselt sarnane SQL-i ja R-ga. Sisaldab andmebaaside ja Exceli tabelitega töötamise funktsionaalsust.

Pandas'i töö andmetega on üles ehitatud NumPy teegile, mis on madalama taseme tööriist.

In [ ]:
```
import pandas as pd
import numpy as np
```

Andmetüüp `Series`

Seriesi struktuur/objekt kujutab endast objekti, mis sarnaneb ühemõõtmelise massiiviga (Pythoni loend (list)), kuid iga loendi elemendile lisatakse indeksid - assotsiatiivne massiiv või sõnastik Pythonis.

Indeks on vasakul ja element ise on paremal. Kui indeks ei ole selgelt määratud, loob pandas automaatselt RangeIndexi 0 kuni N-1, kus N on elementide koguarv. Series on ka salvestatud elementide tüüp (näites on dtype: float64).

Objektil Series on atribuudid, mille kaudu saab elementide loendi ja indeksid, need on `values` ja `index` .

Puuduvad andmed kirjutatakse kujul `np.nan`. Keskmise ja muude tehete arvutamisel ei arvesta vastavad funktsioonid puuduvaid väärtusi.

In [ ]:
```
arr = [1, 3, 4, np.nan, 6, 7]
my_series = pd.Series(arr)
my_series
```

Out[ ]:
```
0    1.0
1    3.0
2    4.0
3    NaN
4    6.0 5    7.0 dtype: float64
```

In [ ]:
```
my_series.index
```

Out[ ]:
```
RangeIndex(start=0, stop=6, step=1)
```

In [ ]:
```
my_series.values
```

Out[ ]:
```
array([ 1.,  3.,  4., nan,  6.,  7.])
```

Tühjade väärtuste leidmiseks on spetsiaalne meetod `.isna()` .

In [ ]:
```
my_series.isna()
```

Out[ ]:
```
0    False
1    False
2     False
3     True
```

```
4     False5    False dtype: bool
```

Andmekogumi kohta põhiline info on meetod `describe`: kirjete arv, keskmine, standardhälve, miinimum, kvartiilid, maksimum ja ka andmetüüp.

In [ ]:
```
my_series.describe()
```

Out[ ]:
```
count    5.000000
mean     4.200000
std      2.387467
min      1.000000
25%      3.000000
50%      4.000000
75%      6.000000
max      7.000000
dtype: float64
```

In [ ]:
```
my_series.mean()
```

Out[ ]:
```
4.2
```

Keskmine veergude järgi

In [ ]:
```
my_series.mean()
```

Out[ ]:
```
4.2
```

Indekseerimine

Selles näites on tavaline indekseerimine.

In [ ]:
```
my_series[2]
```

Out[ ]:
```
4.0
```

Indekseid saab määrata ilmselt:

In [ ]:
```
my_series= pd.Series([1, 3, 4, 5, 6, 7], index=['a', 'b', 'c', 'd', 'e', 'f'])
my_series['f']
```

Out[ ]:
```
7
```

Valimit saab teha mitme indeksi järgi

In [ ]:
```
my_series[['a', 'c', 'd']]
```

```
Out[ ]: a    1
        c    4
        d    5
        dtype: int64
```

```
In [ ]: my_series[['a', 'c', 'd']] = 10
```

```
In [ ]: my_series
```

```
Out[ ]: a    10
        b     3
        c    10 d
        10


        e     6


        f     7


        dtype: int64
```

Series objekti võib pidada sõnaraamatuks, kus võtmeks on indeks ja väärtuseks on element ise.

```
In [ ]: my_series = pd.Series({'a': 1, 'b': 2, 'c': 3, 'd': 4})
        my_series
```

```
Out[ ]: a    1 b
        2 c    3 d
        4


dtype: int64
```

Objektil Series ja selle indeksil on atribuut `name`, mis annab objektile ja indeksile vastavalt nime.

In [ ]:
```
my_series.name = 'numbers'
my_series.index.name = 'letters'
my_series
```

Out[ ]:
```
letters
a    1


b    2c    3 d    4


Name: numbers, dtype: int64
```

Fitreerimine

In [ ]:
```
my_series[my_series >= 7]
```

Out[ ]:
```
Series([], Name: numbers, dtype: int64)
```

In [ ]:
```
my_series[(my_series > 3) & (my_series <= 7)]
```

Out[ ]:
```
letters
d    4
Name: numbers, dtype: int64
```

In [ ]:
```
my_series[(my_series == 3) | (my_series == 7)]
```

Out[ ]:
```
letters
c    3
Name: numbers, dtype: int64
```

In [ ]:
```
my_series[my_series.isin([1, 3, 5, 7])]
```

Out[ ]:
```
letters
a    1 c    3 Name:
numbers, dtype: int64
```

DataFrame

DataFrame'i objekt on tabelina esitatud andmestruktuur. Igas tabelis on alati read ja veerud. DataFrame'i veerud on Series'I objektid, mille read on nende vahetud elemendid.

Näide. DataFrame'i loomine sõnastiku abil.

In [ ]:

```
df = pd.DataFrame({
    'country': ['Estonia', 'Latvia', 'Lithuania', 'Finland', 'Sweden', 'Norway'],
    'capital': ['Tallinn', 'Riga', 'Vilnius', 'Helsinki', 'Stockholm', 'Oslo'],
    'population (million)': [1.4, 1.9, 2.8, 5.5, 10.4, 5.4]
})

df
```

Out[ ]:

	country	capital	population (million)
0	Estonia	Tallinn	1.4
1	Latvia	Riga	1.9
2	Lithuania	Vilnius	2.8
3	Finland	Helsinki	5.5
4	Sweden	Stockholm	10.4
5	Norway	Oslo	5.4

Teine võimalus DataFrame'i loomiseks on pesastatud loendite abil

In [ ]:

```
df = pd.DataFrame([['Estonia', 'Tallinn', 1.4],
                   ['Latvia', 'Riga', 1.9],
                   ['Lithuania', 'Vilnius', 2.8],
 ['Finland', 'Helsinki', 5.5],
                   ['Sweden', 'Stockholm', 10.4],
 ['Norway', 'Oslo', 5.4]],
                  columns=['country', 'capital', 'population (million)'])
df
```

Out[ ]:

	country	capital	population (million)
0	Estonia	Tallinn	1.4
1	Latvia	Riga	1.9
2	Lithuania	Vilnius	2.8
3	Finland	Helsinki	5.5
4	Sweden	Stockholm	10.4
5	Norway	Oslo	5.4

Info saamine Dataframe'I kohta

Meetod info ja saab teavet, nagu pealkiri, väärtuste arv, veergude andmetüübid, kas veerud sisaldavad nullväärtusi.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 3 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
  0    country               6 non-null      object
  1    capital               6 non-null      object
```

In [ ]:
```
df.info()
```

```
 2    population (million)  6 non-null      float64 dtypes: float64(1),
object(2) memory usage: 272.0+ bytes
```

Ridade ja veergude arv - meetod - meetod `shape`

In [ ]:
```
df.shape
```

Out[ ]: (6, 3)

In [ ]:
```
print(df.shape[0]) #horisontaalne suurus - ridade arv print(df.shape[1])
#horisontaalne suurus – vergude arv
```

```
6
3
```

Ridade koguarvu saab ka funktsiooniga `len`

In [ ]:
```
len(df)
```

Out[ ]: 6

Meetod `describe` - saab statistilist teavet veergude väärtuste kohta. Näiteks arvväärtuste jaoks saab teada veeru keskmise, minimaalse ja maksimaalse väärtuse.

In [ ]:
```
df.describe()
```

Out[ ]:

	population (million)
count	6.000000
mean	4.566667
std	3.342255
min	1.400000
25%	2.125000
50%	4.100000
75%	5.475000
max	10.400000

std - standardhälve on arvandmete kogumi hajumise mõõt (kui kaugel andmed aritmeetilisest keskmisest (*mean*) on) - matemaatiline ootus.

Standardhälbe leidmise valem

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x_{cp})^2}{n-1}}$$

25%, 50%, 75% - protsentiilid - jagavad kogu valimi teatud osadeks. Näiteks 25% *Populationi* andmetest on väärtusega alla 2,125 ja 75% on üle 2,125.

Võite leida täieliku dokumentatsiooni `describe()` kohta [siit](https://pandas.pydata.org/pandasdocs/stable/reference/api/pandas.DataFrame.describe.html)

```
In [ ]: df.describe(percentiles=[0.5,0.75,0.9], include='all') # paremeetrite kasutamine
```

Out[ ]:

	country	capital	population (million)
count	6	6	6.000000
unique	6	6	NaN
top	Estonia	Tallinn	NaN
freq	1	1	NaN
mean	NaN	NaN	4.566667
std	NaN	NaN	3.342255
min	NaN	NaN	1.400000
50%	NaN	NaN	4.100000
75%	NaN	NaN	5.475000
90%	NaN	NaN	7.950000
max	NaN	NaN	10.400000

```
In [ ]: df['population (million)'].describe() # saada kirjeldavat statistikat konkreetse
        veeru jaoks
```

```
Out[ ]: count     6.000000
        mean      4.566667
        std       3.342255
        min       1.400000
        25%       2.125000
        50%       4.100000 75%       5.475000 max
        10.400000 Name: population (million),
        dtype: float64
```

Statistilisi funktsioone saab rakendada ka eraldi DataFrame'i või konkreetsete veergude jaoks.

```
In [ ]: df['population (million)'].mean()
```

```
Out[ ]: 4.566666666666666
```

```
In [ ]: df['population (million)'].max()
```

```
Out[ ]: 10.4
```

median - mediaan on arv, mis on täpselt andmekogumi arvude loendi keskel, kui need sorditakse esmalt kasvavas järjekorras. mode on kõige sagedamini esinev väärtus.

```
In [ ]: df['population (million)'].median()
```

```
Out[ ]: 4.1
```

Andmete laadimine

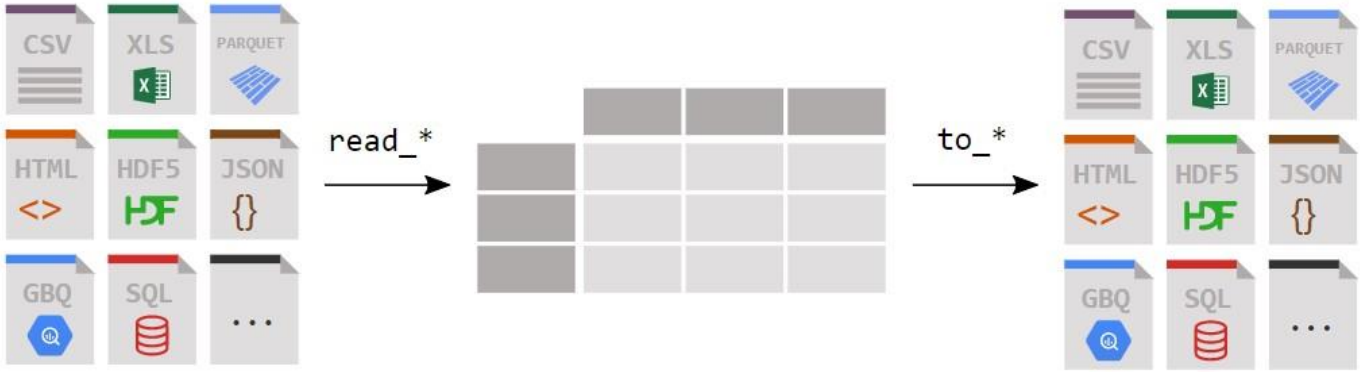
Andmeid saab lugeda erinevatest allikatest: genereerida andmeid, lugeda failidest, andmebaasidest, pilvetehnoloogiatest, näiteks Google Drive, Google Sheets jne.

csv faili allalaadimiseks pandas'is kasutatakse funktsiooni read_csv(). Pandas toetab palju erinevaid failiformaate või andmeallikaid (csv, excel, sql, json...), mida saab lugeda või salvestada.

Andmete lugemiseks kasutatakse funktsioone eesliitega read_*.

* - andmevorming, mida kasutatakse lugemisel/kirjutamisel.

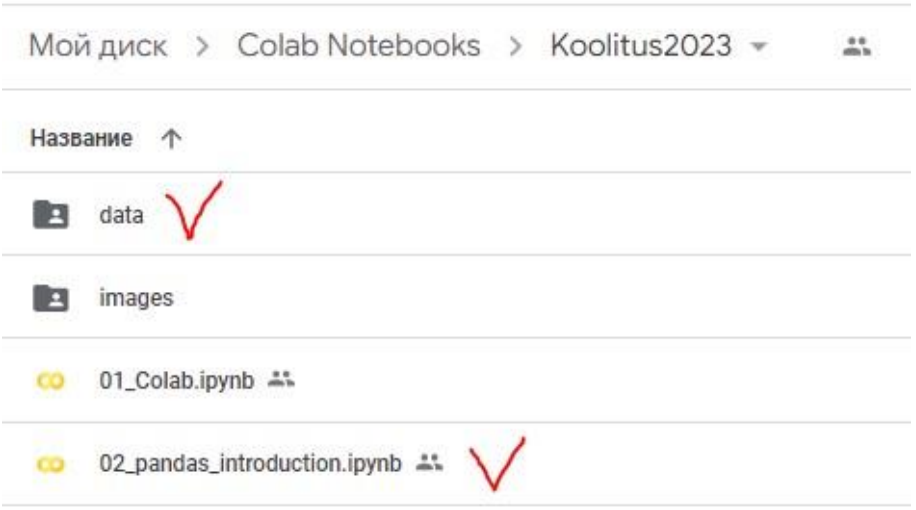
Andmete kirjutamiseks DataFrame'ist faili, andmebaasi kasutatakse funktsioone to_*

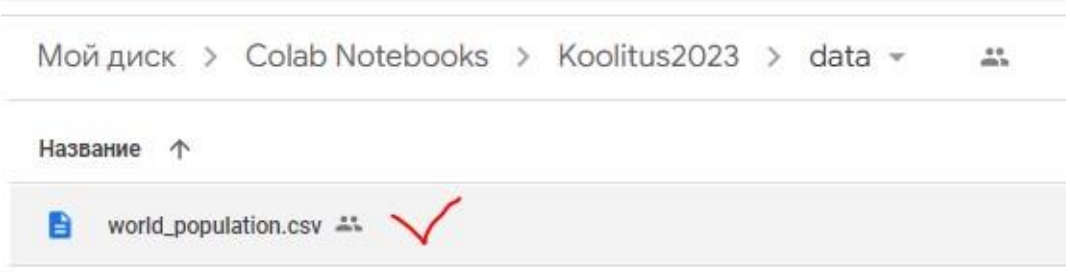


Allikas (http://dfedorov.spb.ru/pandas/02.%20%D0%9A%D0%B0%D0%BA%20%D0%BC%D0%BD%D0%B5%20%D1%87%D

Andmete lugemisel Colabis on mitu võimalust.

1. **Andmete lugemine, seal samas, kus on notebook** või kaustas näiteks data world_population.csv.

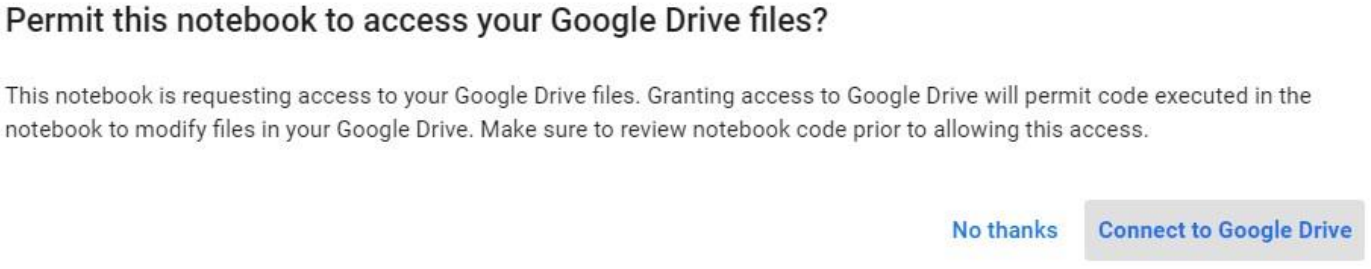





```
In [ ]: from google.colab import drive
        drive.mount('/content/drive', force_remount=True)
```

```
Mounted at /content/drive
```

Google Drive kettale pöördumisel tuleb `drive.mount'il` lubada juurdepääsu.



Seejärel valige kasutaja account ja klõpsake nuppu Luba.

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	2010 Population
0	36	AFG	Afghanistan	Kabul	Asia	41128771	38972230	33753499	28189672
1	138	ALB	Albania	Tirana	Europe	2842321	2866849	2882481	2913399
2	34	DZA	Algeria	Algiers	Africa	44903225	43451666	39543154	35856344

3	213	ASM	American Samoa	Pago Pago	Oceania	44273	46189	51368	54849
4	203	AND	Andorra	Andorra la Vella	Europe	79824	77700	71746	71519

head(), tail() - saab vaadata DataFrame'i 5 esimest või viimast rida. Sulgudes saab määrata soovitud ridade (kirjete) arvu. Näiteks,

In [ ]:
```
df.tail(10)
```

Out[ ]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	201 Populatio
224	43	UZB	Uzbekistan	Tashkent	Asia	34627652	33526656	30949417	2861422
225	181	VUT	Vanuatu	Port-Vila	Oceania	326740	311685	276438	24545
226	234	VAT	Vatican City	Vatican City	Europe	510	520	564	59
227	51	VEN	Venezuela	Caracas	South America	28301696	28490453	30529716	2871502
228	16	VNM	Vietnam	Hanoi	Asia	98186856	96648685	92191398	8741101
229	226	WLF	Wallis and Futuna	Mata-Utu	Oceania	11572	11655	12182	1314
230	172	ESH	Western Sahara	El Aaiún	Africa	575986	556048	491824	41329
231	46	YEM	Yemen	Sanaa	Asia	33696614	32284046	28516545	2474394
232	63	ZMB	Zambia	Lusaka	Africa	20017675	18927715	16248230	1379208
233	74	ZWE	Zimbabwe	Harare	Africa	16320537	15669666	14154937	1283977

sample() - 1 vaikimisi või määratud arv ridu valitud juhuslikult.

In [ ]:
```
df.sample()
```

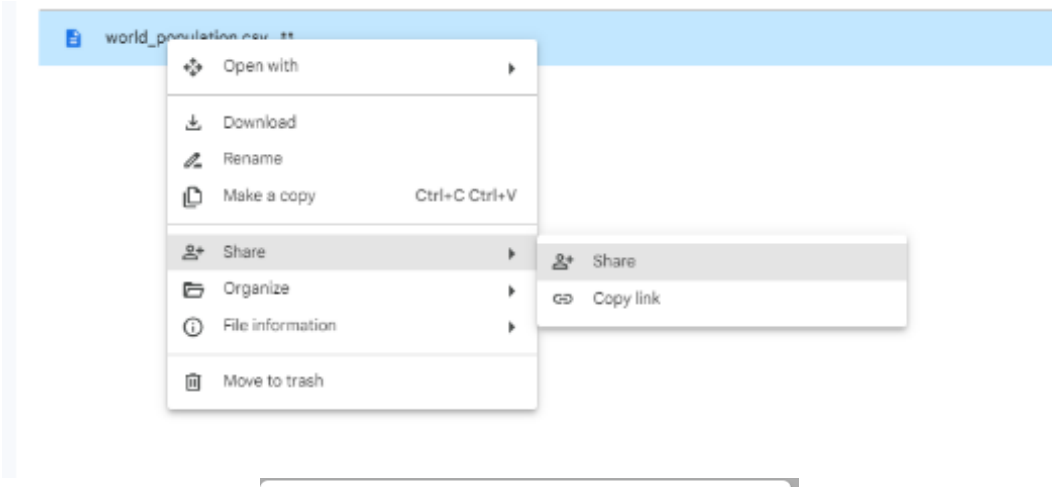
Out[ ]:

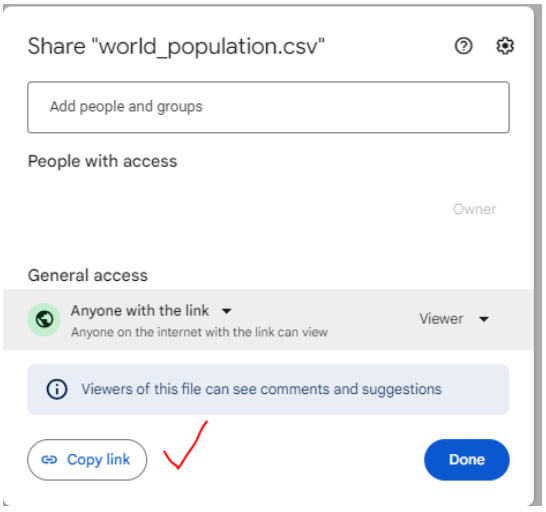
	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	201 Populatio

91	179	ISL	Iceland	Reykjavík	Europe	372899	366669	331060	31833

Teine võimalus on määrata id-viited failile. Näiteks siis, kui see notebook on jagamiseks saadaval ja andmefailile kettal juurdepääsu ei ole.

Ilmuvas aknas peab valima Kopeeri link (ärge unustage juurdepääsu lubada).





Näiteks on see link failile
`https://drive.google.com/file/d/11C4WPvrWuep1M4EolWfNGv3cAyk7tapW/view?usp=share_link`

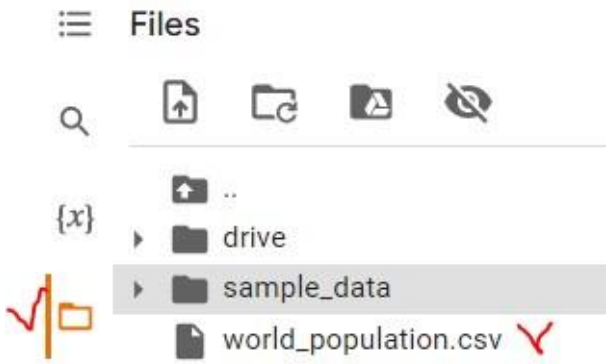
Sellest peab kopeerima ainult id -> `11C4WPvrWuep1M4EolWfNGv3cAyk7tapW` ja käivitada gdown

```
In [ ]: !gdown --id 11C4WPvrWuep1M4EolWfNGv3cAyk7tapW
```

```
/usr/local/lib/python3.8/dist-packages/gdown/cli.py:127: FutureWarning: Option `-
id` was deprecated in version 4.3.1 and will be removed in 5.0. You don't need t o
pass it anymore to use a file ID.
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=11C4WPvrWuep1M4EolWfNGv3cAyk7tapW
To: /content/world_population.csv
```

```
100% 29.2k/29.2k [00:00<00:00, 27.5MB/s]
```

Fail on alla laaditud. Klõpsake vasakpoolsel kaustaikoonil - see fail on seal (kui mitte, siis klõpsake nuppu Refresh/Värskenda). Seejärel paremklõpsake sellel ja seal Copy path.



In [ ]:
```
df = pd.read_csv('/content/world_population.csv')
df.sample(10)
```

Out[ ]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	20 Populatio
107	192	KIR	Kiribati	Tarawa	Oceania	131232	126463	116707	1079
93	4	IDN	Indonesia	Jakarta	Asia	275501339	271857970	259091970	2440161
72	142	GMB	Gambia	Banjul	Africa	2705992	2573995	2253133	19372
196	30	ESP	Spain	Madrid	Europe	47558630	47363807	46431342	465727
83	207	GGY	Guernsey	Saint Peter Port	Europe	63301	62794	61629	607
94	17	IRN	Iran	Tehran	Asia	88550570	87290193	81790841	753738
149	6	NGA	Nigeria	Abuja	Africa	218541212	208327405	183995785	1609528
102	11	JPN	Japan	Tokyo	Asia	123951692	125244761	127250933	1281054
110	103	LAO	Laos	Vientiane	Asia	7529475	7319399	6787419	63234
142	225	NRU	Nauru	Yaren	Oceania	12668	12315	11185	102

Fail world_population - Allikas (https://www.kaggle.com/datasets/iamsouravbanerjee/world-population-dataset?resource=download) - kaggle.com

Ülesanne

1. Looge Colab’is uus fail ipynb.
2. Hankige ja kirjeldage võimalikult üksikasjalikult teavet world_population.csv allika kohta (kasutage: info, describe, columns, value_counts...)
3. Kontrollige, kas on tühjad (puuduvad) väärtused, NaN. Näiteks, df.isnull().sum()