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

In [ ]:
```
pd.options.display.float_format = '{:,.2f}'.format
```

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, 37.2MB/s]
```

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	Popul
223	133	URY	Uruguay	Montevideo	South America	3422794	3429086	3402818	335
213	18	TUR	Turkey	Ankara	Asia	85341241	84135428	79646178	7319
14	176	BHS	Bahamas	Nassau	North America	409984	406471	392697	37
225	181	VUT	Vanuatu	Port-Vila	Oceania	326740	311685	276438	24
33	73	KHM	Cambodia	Phnom Penh	Asia	16767842	16396860	15417523	1436
71	146	GAB	Gabon	Libreville	Africa	2388992	2292573	2028517	171
29	175	BRN	Brunei	Bandar Seri Begawan	Asia	449002	441725	421437	39
157	222	PLW	Palau	Ngerulmud	Oceania	18055	17972	17794	1
102	11	JPN	Japan	Tokyo	Asia	123951692	125244761	127250933	12810
221	3	USA	United States	Washington, D.C.	North America	338289857	335942003	324607776	31118

In [ ]:

Out[ ]:

```
df.describe(include='all')
```

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Po
count	234.00	234	234	234	234	234.00	234.00	
unique	NaN	234	234	234	6	NaN	NaN	
top	NaN	AFG	Afghanistan	Kabul	Africa	NaN	NaN	
freq	NaN	1	1	1	57	NaN	NaN	
mean	117.50	NaN	NaN	NaN	NaN	34,074,414.71	33,501,070.95	31,72
std	67.69	NaN	NaN	NaN	NaN	136,766,424.80	135,589,876.92	130,40
min	1.00	NaN	NaN	NaN	NaN	510.00	520.00	
25%	59.25	NaN	NaN	NaN	NaN	419,738.50	415,284.50	40
50%	117.50	NaN	NaN	NaN	NaN	5,559,944.50	5,493,074.50	5,30
75%	175.75	NaN	NaN	NaN	NaN	22,476,504.75	21,447,979.50	19,73
max	234.00	NaN	NaN	NaN	NaN	1,425,887,337.00	1,424,929,781.00	1,393,71

In [ ]:

```
df.isnull().sum(axis = 0)
```

Out[ ]:

```
Rank                            0
CCA3                            0
Country/Territory               0
Capital                         0
Continent                       0
2022 Population                 0
2020 Population                 0
2015 Population                 0
2010 Population                 0
2000 Population                 0
1990 Population                 0
1980 Population                 0
1970 Population                 0
Area (km²)                      0
Density (per km²)               0
Growth Rate                     0
World Population Percentage     0
dtype: int64
```

In [ ]:

]: Out[ ]:

DataFrame'i andmete lugemine ja redigeerimine

Andmete lugemine veergude/ridade kaupa

Andmete lugemisel saab määrata soovitud read ja veerud

Veerud - saab määrata veeru/veergude vajalikud nimed

```
df['Country/Territory'].head()
```

```
0           Afghanistan
1           Albania
2           Algeria
3           American Samoa
4           Andorra
Name: Country/Territory, dtype: object
```

In [ ]:
```
df[['Country/Territory','Continent','2022 Population']].head()
```

Out[ ]:

	Country/Territory	Continent	2022 Population
0	Afghanistan	Asia	41128771
1	Albania	Europe	2842321
2	Algeria	Africa	44903225
3	American Samoa	Oceania	44273
4	Andorra	Europe	79824

Read

Juurdepääs ridadele indeksi järgi on võimalik mitmel viisil:

`.loc` - kasutatakse juurdepääsuks reasildi järgi

`.iloc` - kasutatakse juurdepääsuks numbrilise väärtuse järgi (alates 0-st)

In [ ]:
```
df.iloc[99]
```

Out[ ]:
```
Rank                          25
CCA3                         ITA
Country/Territory          Italy
Capital                     Rome
Continent                 Europe
2022 Population         59037474
2020 Population         59500579
2015 Population         60232906
2010 Population         59822450
2000 Population         56966397
1990 Population         56756561
1980 Population         56329482
1970 Population         53324036
```

In [ ]:

]: Out[ ]:

```
Area (km²)                      301336
Density (per km²)               195.92
Growth Rate                       1.00
World Population Percentage       0.74
Name: 99, dtype: object
df.iloc[100:103] # lõigete (inglise slice) kasutamine
```

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	Pop
100	52	CIV	Ivory Coast	Yamoussoukro	Africa	28160542	26811790	23596741	21
101	139	JAM	Jamaica	Kingston	North America	2827377	2820436	2794445	2
102	11	JPN	Japan	Tokyo	Asia	123951692	125244761	127250933	128

In [ ]:
```
df.iloc[[1,54,67]]
```

Out[ ]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	2010 Population
1	138	ALB	Albania	Tirana	Europe	2842321	2866849	2882481	2913399
54	84	DOM	Dominican Republic	Santo Domingo	North America	11228821	10999664	10405832	9775755
67	118	FIN	Finland	Helsinki	Europe	5540745	5529468	5479461	5363271

Valimit saab teha indeksi ja veergude järgi

In [ ]:
```
df.loc[[61, 67, 111],['Country/Territory','2022 Population','2020 Population']]
```

Out[ ]:

	Country/Territory	2022 Population	2020 Population
61	Estonia	1326062	1329444
67	Finland	5540745	5529468
111	Latvia	1850651	1897052

Veergude/ridade eemaldamine, lisamine

Kui on vaja lisada veerg, siis piisab, kui kirjutada uue veeru nime ja omistada väärtuse.

Veeru eemaldamiseks kasutatakse funktsiooni *drop* või käsku *del.*

In [ ]:

```
df_temp = df[['Country/Territory','Continent','2022 Population','2020 Populatio n']]
pd.options.mode.chained_assignment = None # keelata sõnumite kuvamine
```

```
#df_temp = df_temp.copy() # sõnum pakub teha koopia või kasutada .l oc, ja siis saab luua uue veeru
df_temp['2022-2020'] = df_temp['2022 Population'] - df_temp['2020 Population'] # uue veeru lisamine'2022-2020' df_temp.sample(5)
```

Out[ ]:

	Country/Territory	Continent	2022 Population	2020 Population	2022-2020
217	Uganda	Africa	47249585	44404611	2844974
44	Cook Islands	Oceania	17011	17029	-18
19	Belgium	Europe	11655930	11561717	94213
3	American Samoa	Oceania	44273	46189	-1916
155	Oman	Asia	4576298	4543399	32899

In [ ]:

```
#df_temp = df_temp.drop(['2022-2020'], axis='columns')
#del df_temp['2022-2020']
df_temp.drop(['2022-2020'], axis='columns', inplace=True)
df_temp.tail()
```

Out[ ]:

	Country/Territory	Continent	2022 Population	2020 Population
229	Wallis and Futuna	Oceania	11572	11655
230	Western Sahara	Africa	575986	556048
231	Yemen	Asia	33696614	32284046
232	Zambia	Africa	20017675	18927715
233	Zimbabwe	Africa	16320537	15669666

inplace = True??? võidakse meetoditest ära võtta, seega on parem kasutada omistamist

Ridade lisamiseks võib kasutada funktsiooni df.loc(), et lisada rida DataFrame'i lõppu.

`df.loc[len(df.index)] = [value1, value2, value3, ...]`

Samuti saab kasutada funktsiooni `df.append()` mõne olemasoleva DataFrame rea lisamiseks teise DataFrame'i lõppu või uue rea lisamiseks. `df = df.append(df2, ignore_index = True)` или `df = df.append(new_row, ignore_index=True)`

In [ ]:

In [ ]:
```
df = df.append({"Country/Territory":"Test", "Capital":"Test"}, ignore_index=True)
```

```
df.tail(3)
```

Out[ ]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	
232	63.00	ZMB	Zambia	Lusaka	Africa	20,017,675.00	18,927,715.00	16,248,230.00	13
233	74.00	ZWE	Zimbabwe	Harare	Africa	16,320,537.00	15,669,666.00	14,154,937.00	12
234	NaN	NaN	Test	Test	NaN	NaN	NaN	NaN	

In [ ]:
```
df.loc[len(df.index)] = [100, 'TST', 'Test', 'Test', 'Test', 10000000, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0]
df.tail(3)
```

Out[ ]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	
233	74.00	ZWE	Zimbabwe	Harare	Africa	16,320,537.00	15,669,666.00	14,154,937.00	1
234	NaN	NaN	Test	Test	NaN	NaN	NaN	NaN	
235	100.00	TST	Test	Test	Test	10,000,000.00	0.00	0.00	

In [ ]:
```
df = df.drop(df.tail(1).index) # viimase rea kustutamine
df.tail()
```

Out[ ]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	
230	172.00	ESH	El Western Sahara Aaiún		Africa	575,986.00	556,048.00	491,824.00	
231	46.00	YEM	Yemen	Sanaa	Asia	33,696,614.00	32,284,046.00	28,516,545.00	24
232	63.00	ZMB	Zambia	Lusaka	Africa	20,017,675.00	18,927,715.00	16,248,230.00	1
233	74.00	ZWE	Zimbabwe	Harare	Africa	16,320,537.00	15,669,666.00	14,154,937.00	1

234	NaN	NaN	Test	Test	NaN	NaN	NaN	NaN

In [ ]:
```
df = df.drop(index=234) # kustutamine indeksi järgi
df.tail(3)
```

Out[ ]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	
231	46.00	YEM	Yemen	Sanaa	Asia	33,696,614.00	32,284,046.00	28,516,545.00	24
232	63.00	ZMB	Zambia	Lusaka	Africa	20,017,675.00	18,927,715.00	16,248,230.00	13
233	74.00	ZWE	Zimbabwe	Harare	Africa	16,320,537.00	15,669,666.00	14,154,937.00	12

Veeru/veergude ümbernimetamine

Saab konkreetseid veerge ümber nimetada.

In [ ]:
```
#df.rename(columns = {'Country/Territory':'country', '2022 Population':'population_2022'}, inplace = True )
df = df.rename(columns = {'Country/Territory':'country', '2022 Population':'population_2022'}) df.head()
```

Out[ ]:

	Rank	CCA3	country	Capital	Continent	population_2022	2020 Population	2015 Population	Pop
0	36.00	AFG	Afghanistan	Kabul	Asia	41,128,771.00	38,972,230.00	33,753,499.00	28,189
1	138.00	ALB	Albania	Tirana	Europe	2,842,321.00	2,866,849.00	2,882,481.00	2,913
2	34.00	DZA	Algeria	Algiers	Africa	44,903,225.00	43,451,666.00	39,543,154.00	35,856
3	213.00	ASM	American Samoa	Pago Pago	Oceania	44,273.00	46,189.00	51,368.00	54
4	203.00	AND	Andorra	Andorra la Vella	Europe	79,824.00	77,700.00	71,746.00	71

Ümber nimetada kõik veerud

In [ ]:
```
df_temp.columns
```

Out[ ]:
```
Index(['Country/Territory', 'Continent', '2022 Population', '2020 Population'], d
type='object')
```

In [ ]:

In [ ]:
```
df_temp.columns = ['country_territory', 'continent', '2022_population', '2020_population']
df_temp[:5]
```

Out[ ]:

	country_territory	continent	2022_population	2020_population
0	Afghanistan	Asia	41128771	38972230
1	Albania	Europe	2842321	2866849
2	Algeria	Africa	44903225	43451666
3	American Samoa	Oceania	44273	46189
4	Andorra	Europe	79824	77700

Reafunktsioonide str rakendamine, näiteks, lower(), replace()...

```
df.columns = df.columns.str.replace(' ', '_')
df.head()
```

Out[ ]:

	Rank	CCA3	country	Capital	Continent	population_2022	2020_Population	2015_Population
0	36.00	AFG	Afghanistan	Kabul	Asia	41,128,771.00	38,972,230.00	33,753,499.00
1	138.00	ALB	Albania	Tirana	Europe	2,842,321.00	2,866,849.00	2,882,481.00
2	34.00	DZA	Algeria	Algiers	Africa	44,903,225.00	43,451,666.00	39,543,154.00
3	213.00	ASM	American Samoa	Pago Pago	Oceania	44,273.00	46,189.00	51,368.00
4	203.00	AND	Andorra	Andorra la Vella	Europe	79,824.00	77,700.00	71,746.00

NaN väärtus

NaN - väärtuste puudumine (tühjad). Andmekogumites võivad puuduvad väärtused kuvada küsimärgi, NaN või lihtsalt tühja lahtrina.

Iga väärtuste puudumise olukord on erinev ja seda tuleb vastavalt ülesandele erinevalt hinnata, näiteks masinõppe mudelite loomiseks tuleb kasvõi read eemaldada, asendada teatud väärtustega, näiteks 'No info', asendada 0 või keskmise väärtusega…

Puudumistega näite jaoks kasutatakse andmestikku *Titanic* teegist *seaborn.*

In [ ]:
```
import seaborn as sns

sns.get_dataset_names()
```

Out[ ]:
```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
```

```
'car_crashes',
'diamonds',
'dots',
'dowjones',
'exercise',
'flights',
'fmri',
'geyser',
'glue',
'healthexp',
'iris',
'mpg',
'penguins',
'planets',
'seaice',
'taxis',
'tips',
'titanic']
```

In [ ]:

```
df_titanic = sns.load_dataset('titanic')
df_titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
  0    survived     891 non-null    int64
  1    pclass       891 non-null    int64
  2    sex          891 non-null    object
  3    age          714 non-null    float64
  4    sibsp        891 non-null    int64
  5    parch        891 non-null    int64
  6    fare         891 non-null    float64
  7    embarked     889 non-null    object
  8    class        891 non-null    category
  9    who          891 non-null    object
  10   adult_male   891 non-null    bool
  11   deck         203 non-null    category
  12   embark_town  889 non-null    object
  13   alive        891 non-null    object
  14   alone        891 non-null    bool    dtypes: bool(2), category(2),
 float64(2), int64(4), object(5) memory usage: 80.7+ KB
```

In [ ]:

```
df_titanic.head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	e
0	0	3	male	22.00	1	0	7.25	S	Third	man	True	NaN	
1	1	1	female	38.00	1	0	71.28	C	First	woman	False	C	
2	1	3	female	26.00	0	0	7.92	S	Third	woman	False	NaN	
3	1	1	female	35.00	1	0	53.10	S	First	woman	False	C	
4	0	3	male	35.00	0	0	8.05	S	Third	man	True	NaN	

In [ ]:

```
df_titanic.head().isna()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	emb
0	False	False	False	False	False	False	False	False	False	False	False	True	
1	False	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	True	
3	False	False	False	False	False	False	False	False	False	False	False	False	

In [ ]:

4	False	False	False	False	False	False	False	False	False	False	False	True

```
df_titanic.isna().mean()
```

Out[ ]:
```
survived      0.00
pclass        0.00
sex           0.00
age           0.20
sibsp         0.00
parch         0.00
fare          0.00
embarked      0.00
class         0.00
who           0.00
adult_male    0.00
deck          0.77
embark_town   0.00
alive         0.00
alone         0.00
dtype: float64
```

Tulemus - 20% age sisaldab tühje väärtusi (NaN), deck - 77%

Puuduvate andmete tuvastamiseks on veel võimalusi - isnull() и notnull().

Selliseid sümboleid nagu tühjad read '' või numpy.inf ei loeta NaN väärtusteks (et loetakse, tuleb seada pandas.options.mode.use_inf_as_na = True).

In [ ]:
```
df_titanic.head().isnull()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	emb
0	False	False	False	False	False	False	False	False	False	False	False	True	
1	False	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	True	
3	False	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	False	True	

```
df_titanic.isnull().sum()
```

Out[ ]:
```
survived      0
pclass        0
sex           0
age         177
sibsp         0
parch         0
```

In [ ]:

```
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

Puuduvad väärtused saab eemaldada meetodi `dropna()` abil. Kustutada saab nii veeru, millel on vähemalt üks puuduv väärtus, kui ka rea. `dropna()` – puuduvate väärtustega ridade eemaldamine.

`dropna(subset=['col1', 'col2'], inplace=True)` – ridade kustutamine, kus on teatud veergudes puuduvad väärtused (subset). `dropna(axis='col1', inplace=True)` – puuduvate väärtustega veergude kustutamine.

In [ ]:

```
df_titanic[df_titanic['age'].isnull() == True].head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
5	0	3	male	NaN	0	0	8.46	Q	Third	man	True	NaN
17	1	2	male	NaN	0	0	13.00	S	Second	man	True	NaN
19	1	3	female	NaN	0	0	7.22	C	Third	woman	False	NaN
26	0	3	male	NaN	0	0	7.22	C	Third	man	True	NaN
28	1	3	female	NaN	0	0	7.88	Q	Third	woman	False	NaN

```
df_titanic.dropna(subset=['age'], inplace=True)
df_titanic.isnull().sum()
```

Out[ ]:

```
survived         0
pclass           0
sex              0
age              0
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           530
embark_town      2
alive            0
```

In [ ]:
```
alone            0
dtype: int64
```

Andmete filtreerimine

Andmete valimist saab teha mitmel viisil: loogiliste avaldiste abil, meetodiga *query, loc, iloc.*

Loogilised avaldised

Kasutatakse DataFrame'I operaatorit [ ], millesse pannakse filtreerimise tingimus – loogiline avaldis.

In [ ]:
```
df[df['population_2022'] > 1_000_000_000] # rahvastik üle 1 mld
```

Out[ ]:

	Rank	CCA3	country	Capital	Continent	population_2022	2020_Population	2015_Population	201
41	1	CHN	China	Beijing	Asia	1425887337	1424929781	1393715448	
92	2	IND	India	New Delhi	Asia	1417173173	1396387127	1322866505	

Saab ühendada mitu tingimust, kasutades operaatorit | ja & (VÕI ja I), samal ajal kui loogilised tingimused tuleb määrata ()

In [ ]:
```
df['Continent'].unique() # unikaalsete väärtuste saamine
```

Out[ ]:
```
array(['Asia', 'Europe', 'Africa', 'Oceania', 'North America',
       'South America'], dtype=object)
```

```
df[((df['Continent']=='Asia') | (df['Continent']=='Africa')) & (df['population_2022'] > 100_000_000)]
```

Out[ ]:

	Rank	CCA3	country	Capital	Continent	population_2022	2020_Population	2015_Populatio
16	8	BGD	Bangladesh	Dhaka	Asia	171186372	167420951	15783000
41	1	CHN	China	Beijing	Asia	1425887337	1424929781	139371544
57	14	EGY	Egypt	Cairo	Africa	110990103	107465134	9772379
63	12	ETH	Ethiopia	Addis Ababa	Africa	123379924	117190911	10247189
92	2	IND	India	New Delhi	Asia	1417173173	1396387127	132286650

In [ ]:

93	4	IDN	Indonesia	Jakarta	Asia	275501339	271857970	25909197
102	11	JPN	Japan	Tokyo	Asia	123951692	125244761	12725093
149	6	NGA	Nigeria	Abuja	Africa	218541212	208327405	18399578
156	5	PAK	Pakistan	Islamabad	Asia	235824862	227196741	21096929
163	13	PHL	Philippines	Manila	Asia	115559009	112190977	10303136

In [ ]:
```
df[(df['Continent'].isin(['Asia','Africa'])) & (df['population_2022'] > 100_000_000)]
```

Out[ ]:

	Rank	CCA3	country	Capital	Continent	population_2022	2020_Population	2015_Populatio
16	8	BGD	Bangladesh	Dhaka	Asia	171186372	167420951	15783000
41	1	CHN	China	Beijing	Asia	1425887337	1424929781	139371544
57	14	EGY	Egypt	Cairo	Africa	110990103	107465134	9772379
63	12	ETH	Ethiopia	Addis Ababa	Africa	123379924	117190911	10247189
92	2	IND	India	New Delhi	Asia	1417173173	1396387127	132286650
93	4	IDN	Indonesia	Jakarta	Asia	275501339	271857970	25909197
102	11	JPN	Japan	Tokyo	Asia	123951692	125244761	12725093
149	6	NGA	Nigeria	Abuja	Africa	218541212	208327405	18399578
156	5	PAK	Pakistan	Islamabad	Asia	235824862	227196741	21096929
163	13	PHL	Philippines	Manila	Asia	115559009	112190977	10303136

Päringukeel query

Luuakse avaldis, mis on ühe või mitme tingimuse kombinatsioon.

Saab kasutada mitut tingimust ühes veerus, kasutades operaatoreid and, or, not.

Filtrite väärtusi saab salvestada muutujasse ja muutujat saab kasutada päringus, kasutades sümbolit @.

Võimalik on edastada ka argument inplace = True funktsioonile, et muuta algset DataFrame'i.

In [ ]:
```
df.query("Continent=='Africa' or Continent=='Asia'").sample(5, random_state=4) # r
andom_state - juhuslik valim
```

Out[ ]:

	Rank	CCA3	country	Capital	Continent	population_2022	2020_Population	2015_Population
135	134	MNG	Mongolia	Ulaanbaatar	Asia	3398366	3294335	2964749
38	117	CAF	Central African Republic	Bangui	Africa	5579144	5343020	4819333
63	12	ETH	Ethiopia	Addis Ababa	Africa	123379924	117190911	102471895
195	86	SSD	South Sudan	Juba	Africa	10913164	10606227	11194299
5	42	AGO	Angola	Luanda	Africa	35588987	33428485	28127721

In [ ]:
```
continent = 'Asia'

df.query("Continent==@continent").head()
```

Out[ ]:

	Rank	CCA3	country	Capital	Continent	population_2022	2020_Population	2015_Population
0	36	AFG	Afghanistan	Kabul	Asia	41128771	38972230	33753499
9	140	ARM	Armenia	Yerevan	Asia	2780469	2805608	2878595
13	91	AZE	Azerbaijan	Baku	Asia	10358074	10284951	9863480
15	154	BHR	Bahrain	Manama	Asia	1472233	1477469	1362142
16	8	BGD	Bangladesh	Dhaka	Asia	171186372	167420951	157830000

In [ ]:

Andmete rühmitamine ja sortimine

`groupby` meetod andmete rühmitamiseks ühe või mitme veeru alusel. `sort_values` andmete sortimine. Vaikimisi on sortimine kasvavas järjekorras (ascending=True). kahanevas järjekorras (ascending=False)

```
df.groupby('Continent')['country'].count()
```

Out[ ]:
```
Continent
Africa           57
Asia             50
Europe           50
North America    40
Oceania          23
South America    14
Name: country, dtype: int64
```

```
df.sort_values(by=['population_2022'], ascending=False).head(10)
```

In [ ]:

Out[ ]:

	Rank	CCA3	country	Capital	Continent	population_2022	2020_Population	2015_Populat
41	1	CHN	China	Beijing	Asia	1425887337	1424929781	1393715
92	2	IND	India	New Delhi	Asia	1417173173	1396387127	1322866
221	3	USA	United States	Washington, D.C.	North America	338289857	335942003	324607
93	4	IDN	Indonesia	Jakarta	Asia	275501339	271857970	259091
156	5	PAK	Pakistan	Islamabad	Asia	235824862	227196741	210969
149	6	NGA	Nigeria	Abuja	Africa	218541212	208327405	183995
27	7	BRA	Brazil	Brasilia	South America	215313498	213196304	205188
16	8	BGD	Bangladesh	Dhaka	Asia	171186372	167420951	157830
171	9	RUS	Russia	Moscow	Europe	144713314	145617329	144668
131	10	MEX	Mexico	Mexico City	North America	127504125	125998302	120149

Eraldi ülesannete klassi, mis ei saa ilma tsükliteta, nimetatakse **andmete agregeerimiseks.** Need min, max, sum, count, mean jne.

Funktsioon agg võimaldab rakendada mitut funktsiooni korraga.

In [ ]:

Näiteks peab leidma rahvastiku arvu, keskmise, minimaalse ja maksimaalse väärtuse 2022. aastal kontinentide lõikes.

```
df.groupby(['Continent'])['population_2022'].agg(['count','mean','max','min'])
```

Out[ ]:

Continent	count	mean	max	min
Africa	57	25,030,367.23	218541212	107118
Asia	50	94,427,665.48	1425887337	449002
Europe	50	14,862,950.76	144713314	510
North America	40	15,007,403.40	338289857	4390
Oceania	23	1,958,198.00	26177413	1871
South America	14	31,201,186.29	215313498	3780

In [ ]:

```
(df.groupby(['Continent'])
[['population_2022','2020_Population']]
 .agg({'population_2022':['count','mean','median'],
       '2020_Population':['mean','median']}))
```

Out[ ]:

	population_2022			2020_Population	
	count	mean	median	mean	median
Continent					
Africa	57	25,030,367.23	13,352,864.00	23,871,435.26	12,643,123.00
Asia	50	94,427,665.48	18,082,920.00	93,261,730.70	17,688,051.50
Europe	50	14,862,950.76	5,228,714.00	14,915,843.92	5,162,979.00
North America	40	15,007,403.40	236,399.00	14,855,914.82	234,990.50
Oceania	23	1,958,198.00	114,164.00	1,910,148.96	112,106.00
South America	14	31,201,186.29	15,112,555.00	30,823,574.50	14,762,378.50

Andmete tükeldamine intervallideks (bins)

.Pidevate arvandmetega töötamisel on sageli kasulik jagada/tükeldada (to bin) andmed mitmeks segmendiks, et neid edasi analüüsida - andmete diskrediteerimine.

Pandas saab seda teha funktsioonidega `cut` ja `qcut` .

Funktsiooni cut abil saab jagada/tükeldada ühesuurusteks gruppideks või edastada loendi väärtustest, mis esindavad iga rühma väärtuste laiust.

`qcut` jagab väärtused gruppideks nii, et igas grupis on sama arv elemente.

```
df['population_bins'] = pd.cut(df['population_2022'], 5)
df[['country','population_2022','population_bins']].sample(10)
```

Out[ ]:

In [ ]:

	country	population_2022	population_bins
198	Sudan	46874204	(-1425376.827, 285177875.4]
226	Vatican City	510	(-1425376.827, 285177875.4]
91	Iceland		(-1425376.827, 285177875.4]
190	Slovenia	372899	(-1425376.827, 285177875.4]
135	Mongolia		(-1425376.827, 285177875.4]
101	Jamaica	2119844	(-1425376.827, 285177875.4]
70	French Polynesia	3398366	(-1425376.827, 285177875.4]
7	Antigua and Barbuda	2827377	(-1425376.827, 285177875.4]
4	Andorra	306279	(-1425376.827, 285177875.4]
17	Barbados	93763	(-1425376.827, 285177875.4]
		79824	
		281635	

```
df.groupby('population_bins')['population_2022'].agg(['count','mean'])
```

In [ ]:
Out[ ]:

population_bins	count	mean
(-1425376.827, 285177875.4]	231	20,744,860.06
(285177875.4, 570355240.8]	1	338,289,857.00
(570355240.8, 855532606.2]	0	NaN
(855532606.2, 1140709971.6]	0	NaN
(1140709971.6, 1425887337.0]	2	1,421,530,255.00

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

Out[ ]:
```
count                              234
unique                               3
top       (-1425376.827, 285177875.4]
freq                               231
Name: population_bins, dtype: object
```

In [ ]:
```
df['population_bins'].cat.categories
```

Out[ ]:
```
IntervalIndex([(-1425376.827, 285177875.4], (285177875.4, 570355240.8], (57035524
0.8, 855532606.2], (855532606.2, 1140709971.6], (1140709971.6, 1425887337.0]], dt
ype='interval[float64, right]')
first_bin = df['population_bins'].cat.categories[0] # muuta esimese intervalli
algus 0-ks - allikas stackoverflow new_bin = pd.Interval(0, first_bin.right)
```

In [ ]:
```
df['population_bins'] = df['population_bins'].cat.rename_categories({first_bin: new_bin}) df['population_bins'].cat.categories
```

Out[ ]:
```
IntervalIndex([(0.0, 285177875.4], (285177875.4, 570355240.8], (570355240.8, 855532606.2], (855532606.2, 1140709971.6], (1140709971.6, 1425887337.0]], dtype='interval[float64, right]')
```

In [ ]:
```
df.groupby('population_bins')['population_2022'].agg(['count','mean'])
```

Out[ ]:

population_bins	count	mean
(0.0, 285177875.4]	231	20,744,860.06
(285177875.4, 570355240.8]	1	338,289,857.00
(570355240.8, 855532606.2]	0	NaN
(855532606.2, 1140709971.6]	0	NaN
(1140709971.6, 1425887337.0]	2	1,421,530,255.00

Intervallid saab määrata väärtuste loendina

In [ ]:
```
bins = [0, 100_000, 1_000_000, 10_000_000, 100_000_000, 1_000_000_000, 10_000_000_000] # np.inf
df['population_bins'] = pd.cut(df['population_2022'], bins) # saab luua ka binide nimede loendit labels
df[['country','population_2022','population_bins']].sample(10)
```

Out[ ]:

	country	population_2022	population_bins
165	Portugal	10270865	(10000000, 100000000]
134	Monaco	36469	(0, 100000]
145	New Caledonia	289950	(100000, 1000000]
96	Ireland	5023109	(1000000, 10000000]
156	Pakistan	235824862	(100000000, 1000000000]
190	Slovenia	2119844	(1000000, 10000000]
230	Western Sahara	575986	(100000, 1000000]
137	Montserrat	4390	(0, 100000]
118	Luxembourg	647599	(100000, 1000000]
229	Wallis and Futuna	11572	(0, 100000]

```
df.groupby('population_bins')['population_2022'].agg(['count','mean'])
```

Out[ ]:

population_bins	count	mean

In [ ]:

(0, 100000]	35	37,999.23
(100000, 1000000]	39	410,368.15
(1000000, 10000000]	67	4,722,366.58
(10000000, 100000000]	79	32,859,042.19
(100000000, 1000000000]	12	183,396,275.58
(1000000000, 10000000000]	2	1,421,530,255.00

qcut´i kasutamine

In [ ]:
```
df['population_bins_2'] = pd.qcut(df['population_2022'], 5)
df[['country','population_2022','population_bins_2']].sample(10)
```

Out[ ]:

	country	population_2022	population_bins_2
153	Northern Mariana Islands	49551	(509.999, 225380.8]
91	Iceland	372899	(225380.8, 2756137.8]
90	Hungary	9967308	(9964403.8, 33172058.2]
149	Nigeria	218541212	(33172058.2, 1425887337.0]
57	Egypt	110990103	(33172058.2, 1425887337.0]
19	Belgium	11655930	(9964403.8, 33172058.2]
205	Tanzania	65497748	(33172058.2, 1425887337.0]
81	Guam	171774	(509.999, 225380.8]
118	Luxembourg	647599	(225380.8, 2756137.8]
87	Haiti	11584996	(9964403.8, 33172058.2]

In [ ]:
```
df.groupby('population_bins_2')['population_2022'].agg(['count','mean'])
```

Out[ ]:

population_bins_2	count	mean
(509.999, 225380.8]	47	63,853.83
(225380.8, 2756137.8]	47	1,122,663.28
(2756137.8, 9964403.8]	46	5,826,071.30
(9964403.8, 33172058.2]	47	17,792,272.38
(33172058.2, 1425887337.0]	47	144,966,184.17

In [ ]:

```
# Andmete vormindamine
(df.groupby(['Continent'])['population_2022']
.agg(['count', 'mean'])
.sort_values('count',ascending=False)
.style
.applymap(
     lambda x: 'color : green; font-weight:bold' if x>50 else '',
     subset=['count']))
```

Out[ ]:

Continent	count	mean
Africa	**57**	25030367.228070
Asia	50	94427665.480000
Europe	50	14862950.760000
North America	40	15007403.400000
Oceania	23	1958198.000000
South America	14	31201186.285714

Pivot-tabel

Pivot-tabel võtab sisendil vastu üksikute veergude andmed ja rühmitab need, moodustades kahemõõtmelise tabeli, mis rakendab andmete mitmemõõtmelist üldistamist.

DataFrame'i objekti meetodi `pivot_table` täielik signatuur:

`DataFrame.pivot_table(values=None, index=None, columns=None, aggfunc='mean', fill_value=None, margins=False, dropna=True)` Parameeter `index` : näitab rühmitust ridade tasandil.

`columns` : näitab rühmitust veergude tasandil.

`values` : need on arvväärtused, mida tuleb üldistada, näiteks arvutada kogus, summa jne.

Parameeter `aggfunc` määrab agregaadi tüübi. Vaikimisi võrdub selle väärtus '`mean`'-iga. Nagu `groupby` puhul, saab ka agregaadi tüübi määrata kas etteantud rea või loendi abil (nt 'mean', 'count' jt.) või agregatsiooni rakendava funktsiooni kaudu (nt np.sum() või sum() jt). Parameetrid `fill_value` ja `dropna` määravad, kuidas puuduvaid andmeid töödeldakse.

`margins` - kokkuvõtete lisamine veergudesse ja ridadesse. Kokkuvõttele saab määrata nime, kasutades `margins_name` .

```
df.pivot_table(index='Continent', values=['population_2022']) # vaikimisi mean
```

Out[ ]:

Continent	population_2022

In [ ]:

Africa	25,030,367.23
Asia	94,427,665.48
Europe	14,862,950.76
North America	15,007,403.40
Oceania	1,958,198.00
South America	31,201,186.29

In [ ]:
```
df.pivot_table(index='Continent',
               values=['population_2022','2020_Population'],
 aggfunc=['count','mean'],       margins=True)
```

Out[ ]:

	count		mean	
	2020_Population	**population_2022**	**2020_Population**	**population_2022**
Continent				
Africa	57	57	23,871,435.26	25,030,367.23
Asia	50	50	93,261,730.70	94,427,665.48
Europe	50	50	14,915,843.92	14,862,950.76
North America	40	40	14,855,914.82	15,007,403.40
Oceania	23	23	1,910,148.96	1,958,198.00
South America	14	14	30,823,574.50	31,201,186.29
All	234	234	33,501,070.95	34,074,414.71

In [ ]:
```
p_table = df.pivot_table(index='Continent',
 values=['population_2022','2020_Population'],
 aggfunc={'population_2022':['count','mean','median'], '2020_Populat
 ion':['mean','median']})
```

```
p_table
```

Out[ ]:

	2020_Population		population_2022		
	mean	**median**	**count**	**mean**	**median**
Continent					
Africa	23,871,435.26	12,643,123.00	57.00	25,030,367.23	13,352,864.00
Asia	93,261,730.70	17,688,051.50	50.00	94,427,665.48	18,082,920.00
Europe	14,915,843.92	5,162,979.00	50.00	14,862,950.76	5,228,714.00
North America	14,855,914.82	234,990.50	40.00	15,007,403.40	236,399.00
Oceania	1,910,148.96	112,106.00	23.00	1,958,198.00	114,164.00
South America	30,823,574.50	14,762,378.50	14.00	31,201,186.29	15,112,555.00

In [ ]:
```
p_table.columns = ['_'.join(col).strip() for col in p_table.columns.values] #
 veergude nimede vormindamine - ühendamine üheks nimeks p_table
```

In [ ]:

Out[ ]:

Continent	2020_Population_mean	2020_Population_median	population_2022_count	population_2022_me
Africa	23,871,435.26	12,643,123.00	57.00	25,030,367
Asia	93,261,730.70	17,688,051.50	50.00	94,427,665
Europe	14,915,843.92	5,162,979.00	50.00	14,862,950
North America	14,855,914.82	234,990.50	40.00	15,007,403
Oceania	1,910,148.96	112,106.00	23.00	1,958,198
South America	30,823,574.50	14,762,378.50	14.00	31,201,186

```
column_order = ['population_2022_count', '2020_Population_mean', '2020_Population_
median', 'population_2022_mean', 'population_2022_median'] p_table =
p_table.reindex(column_order, axis=1)
p_table = p_table.rename(columns = {'population_2022_count': 'count_countries'})
p_table
```

Out[ ]:

Continent	count_countries	2020_Population_mean	2020_Population_median	population_2022_mean	po
Africa	57.00	23,871,435.26	12,643,123.00	25,030,367.23	
Asia	50.00	93,261,730.70	17,688,051.50	94,427,665.48	
Europe	50.00	14,915,843.92	5,162,979.00	14,862,950.76	
North America	40.00	14,855,914.82	234,990.50	15,007,403.40	
Oceania	23.00	1,910,148.96	112,106.00	1,958,198.00	
South America	14.00	30,823,574.50	14,762,378.50	31,201,186.29	

Ülesanne

Iga ülesanne – erinev jupyter fail.

Igas ülesandes lisage andmestiku kirjeldus, allikaviited, järeldused saadud info põhjal, näiteks describe kirjelda näidatut või andmete rühmitamine koos suuruse, keskmise ja mediaani arvutamisega näitas sõltuvust, seda madalam väärtus ... seda rohkem ....

Ülesanne German Credit Risk

Analüüsige andmestikku German Credit Risk Allikas 1 (https://www.kaggle.com/datasets/kabure/german-creditdata-with-risk) (on veerg Risk) või Allikas 2 (https://www.kaggle.com/datasets/uciml/german-credit) andmestiku kirjeldus.

1. Määrake andmestiku suurus (veergude ja ridade arv).Tüüpide määramiseks kasutage dtypes'i.
   Printige veergude nimed.
2. Veerg Risk - vahetage väärtus 1/0.
3. Kirjeldav statistika describe abil, info. Tehke järeldused.
4. Otsustage, kas kõik veerud on analüüsimiseks vajalikud. Ümbernimetage ka veerud, et kõik nimetused on alumises registris ja tühik on alakriipsuga _.
5. Leidke, kas andmestikus on tühjad väärtused.
6. Määrake teatud arvuliste väljade jaoks mediaan ja võrrelge keskmise väärtusega, määrake ka protsentide abil, kuidas on andmed jaotatud, kas ühtlaselt või mitte. Saab lisada graafikuid (pandas plot, hist)
7. Andmete rühmitamise abil leidke, millised laenusummad kõige sagedamini kliendid said (kasutage *bine* - intervalle).
8. Analüüsige, millest sõltuvad laenuvõlad (vanus, sugu, laenusumma jne või kohe mitmest tunnusest, näiteks vanus ja laenusumma. Kirjutage oma järeldused. Kus on vaja, kasutage andmeintervalle (bine).
9. Andmestiku andmete analüüsimiseks looge mitu liigendtabelit/pivot-tabelit.
10. Kirjutage kokkuvõte - lõpparuanne andmete analüüsi kohta.

Ülesanne Data Science Job Posting on Glassdoor

On olemas andmestik Kaggle Link (https://www.kaggle.com/datasets/rashikrahmanpritom/data-science-job-posting-onglassdoor)- Data Science´i spetsialistide töötasu.

Analüüsge andmeid, tehke järeldusi, millest sõltub Data Science'i spetsialistide palk.