SPRAWOZDANIE

Zajęcia: Nauka o danych I

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 1 (27.09.2005) Data: 5.10.2025 Temat: Ustalenia platformy Jupyter. Użycie biblioteki pandas Wariant 2	Marzena Swakoń Informatyka II stopień, zaoczne, 1 semestr, gr B

1. Polecenie:

Zadanie dotyczy pobrania danych z pliku, tworzenia ramki danych, wykonania poszczególnych zadań na podstawie odpowiedniego zbioru danych:

Wariant 2. Gross Domestic Product Per Capita 1960-2050

http://ghdx.healthdata.org/record/ihme-data/global-gdp-per-capita-1960-2050

2. Opis programu opracowanego (kody źródłowe, rzuty ekranu)

[298]:
```
import pandas as pd
# ładowanie biblioteki Pandas

df = pd.read_csv("IHME_GDP_1960_2050_Y2021M09D22.csv",
                 delimiter = ',', encoding = 'utf-8')
# ładowanie csv do dataframe (Gross Domestic Product Per Capita 1960-2050 http://ghdx.healthdata.org/record/ihme-data/global-gdp-per-capita-1960-2050)
```

[299]:
```
dict_GDP_per_person = {"Location" : df.location_name, "GDP_per_person" : df.gdp_ppp_mean}
df = pd.DataFrame(dict_GDP_per_person)
# tworzenie ramki danych ze słownika
df
```

[299]:

	Location	GDP_per_person
0	Global	1.748345e+13
1	Global	1.813537e+13
2	Global	1.895328e+13
3	Global	1.965662e+13
4	Global	2.100575e+13
...	...	...
19833	Low income	3.617310e+12
19834	Low income	3.724063e+12
19835	Low income	3.831942e+12
19836	Low income	3.941856e+12
19837	Low income	4.053883e+12

19838 rows × 2 columns

[300]:
```
df.to_csv("GDP_per_person.csv")
# zachowanie ramki danych pobranych z pliku w formacie csv (xlsx)
```

[301]:
```
df = pd.read_csv("IHME_GDP_1960_2050_Y2021M09D22.csv", delimiter = ',', encoding = 'utf-8')

lists_GDP_per_person = [[df.location_name], [df.gdp_ppp_mean]]
pd.DataFrame(lists_GDP_per_person)
# tworzenie ramki danych z listy list
```

[301]:

	0
0	0 Global 1 Global 2 ...
1	0 1.748345e+13 1 1.813537e+13 2 ...

[302]:
```
pd.DataFrame(lists_GDP_per_person).T
# transponowanie (wymieniamy kolumny a wierszy)
```

[302]:

	0	1
0	0 Global 1 Global 2 ...	0 1.748345e+13 1 1.813537e+13 2 ...

[303]:
```
df.head(10)
# pierwsze 10 wierszy ramki danych
```

[303]:

	location_id	location_name	iso3	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
0	1	Global	G	Global	1960	1.748345e+13	1.601915e+13	1.911586e+13	1.296863e+13	1.266890e+13	1.334177e+13
1	1	Global	G	Global	1961	1.813537e+13	1.659537e+13	1.982493e+13	1.346097e+13	1.314767e+13	1.383021e+13
2	1	Global	G	Global	1962	1.895328e+13	1.739039e+13	2.061477e+13	1.406576e+13	1.376060e+13	1.443746e+13
3	1	Global	G	Global	1963	1.965662e+13	1.811706e+13	2.134993e+13	1.461831e+13	1.432132e+13	1.497693e+13
4	1	Global	G	Global	1964	2.100575e+13	1.935664e+13	2.276791e+13	1.552986e+13	1.523498e+13	1.587998e+13
5	1	Global	G	Global	1965	2.202459e+13	2.034585e+13	2.382275e+13	1.628972e+13	1.598727e+13	1.663310e+13
6	1	Global	G	Global	1966	2.306193e+13	2.136085e+13	2.489782e+13	1.708885e+13	1.678223e+13	1.742396e+13
7	1	Global	G	Global	1967	2.391268e+13	2.217842e+13	2.577837e+13	1.770884e+13	1.740660e+13	1.804193e+13
8	1	Global	G	Global	1968	2.516723e+13	2.340479e+13	2.698215e+13	1.865379e+13	1.833216e+13	1.898399e+13
9	1	Global	G	Global	1969	2.642403e+13	2.464521e+13	2.831984e+13	1.955395e+13	1.921164e+13	1.987990e+13

[304]:
```
df.tail(10)
# ostatnie 10 wierszy ramki danych
```

[304]:

	location_id	location_name	iso3	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
19828	44578	Low income	NaN	World Bank Income Group	2041	3.120963e+12	2.724077e+12	3.582807e+12	9.752426e+11	8.875033e+11	1.068693e+12
19829	44578	Low income	NaN	World Bank Income Group	2042	3.216988e+12	2.801335e+12	3.686394e+12	1.008813e+12	9.169149e+11	1.107239e+12
19830	44578	Low income	NaN	World Bank Income Group	2043	3.314031e+12	2.886768e+12	3.815672e+12	1.042881e+12	9.461940e+11	1.147550e+12
19831	44578	Low income	NaN	World Bank Income Group	2044	3.413020e+12	2.968361e+12	3.933135e+12	1.077714e+12	9.735487e+11	1.188093e+12
19832	44578	Low income	NaN	World Bank Income Group	2045	3.514244e+12	3.055623e+12	4.049325e+12	1.113207e+12	1.003241e+12	1.228145e+12
19833	44578	Low income	NaN	World Bank Income Group	2046	3.617310e+12	3.140835e+12	4.166469e+12	1.149318e+12	1.031500e+12	1.271992e+12
19834	44578	Low income	NaN	World Bank Income Group	2047	3.724063e+12	3.225849e+12	4.292403e+12	1.186597e+12	1.061313e+12	1.318836e+12
19835	44578	Low income	NaN	World Bank Income Group	2048	3.831942e+12	3.307609e+12	4.424674e+12	1.224062e+12	1.092874e+12	1.365610e+12
19836	44578	Low income	NaN	World Bank Income Group	2049	3.941856e+12	3.398884e+12	4.560961e+12	1.262129e+12	1.122895e+12	1.413991e+12
19837	44578	Low income	NaN	World Bank Income Group	2050	4.053883e+12	3.482933e+12	4.713596e+12	1.300764e+12	1.151548e+12	1.457362e+12

[305]:
```
df.info()
# informacja o ramce danych
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19838 entries, 0 to 19837
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   location_id    19838 non-null  int64
 1   location_name  19838 non-null  object
 2   iso3           18655 non-null  object
 3   level          19838 non-null  object
 4   year           19838 non-null  int64
 5   gdp_ppp_mean   19838 non-null  float64
 6   gdp_ppp_lower  19838 non-null  float64
 7   gdp_ppp_upper  19838 non-null  float64
 8   gdp_usd_mean   19838 non-null  float64
 9   gdp_usd_lower  19838 non-null  float64
 10  gdp_usd_upper  19838 non-null  float64
dtypes: float64(6), int64(2), object(3)
memory usage: 1.7+ MB
```

[306]:
```
df.shape
# ile wierszy i kolumn znajduje się w ramce danych
```

[306]:
```
(19838, 11)
```

[307]:
```
df.describe()
# informacje statystyczne w kolumnach (wartości niepowtarzalne, średnia, odchylenie standardowe, minimum, kwartyle, maksimum)
```

[307]:

	location_id	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
count	19838.000000	19838.000000	1.983800e+04	1.983800e+04	1.983800e+04	1.983800e+04	1.983800e+04	1.983800e+04
mean	949.871560	2005.000000	1.334543e+12	1.235788e+12	1.444079e+12	8.554096e+11	8.197528e+11	8.967612e+11
std	5965.433243	26.268513	9.148287e+12	8.610030e+12	9.789327e+12	6.286364e+12	6.041288e+12	6.585419e+12
min	1.000000	1960.000000	1.448063e+02	6.299026e+01	2.621094e+02	1.174979e+02	8.318772e+01	1.270468e+02
25%	63.000000	1982.000000	3.678736e+03	2.639116e+03	4.829886e+03	1.624411e+03	1.395430e+03	1.828575e+03
50%	125.500000	2005.000000	1.103640e+04	8.105541e+03	1.346178e+04	4.863298e+03	4.279291e+03	5.465731e+03
75%	183.000000	2028.000000	2.949281e+04	2.308992e+04	3.562660e+04	1.997525e+04	1.795003e+04	2.223434e+04
max	44578.000000	2050.000000	1.827414e+14	1.667007e+14	2.025062e+14	1.119468e+14	1.017185e+14	1.239708e+14

[308]:
```
df.describe(include = 'all')
# statystyki obejmują nie tylko kolumny liczbowe, ale także wiersze
# (unique - ile unikalnych wartości, top - jaka jest najpopularniejsza wartość, freq - jak często najpopularniejsza)
```

[308]:

	location_id	location_name	iso3	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
count	19838.000000	19838	18655	19838	19838.000000	1.983800e+04	1.983800e+04	1.983800e+04	1.983800e+04	1.983800e+04	1.983800e+04
unique	NaN	216	205	4	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	South Asia	G	Country	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	182	91	18564	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	949.871560	NaN	NaN	NaN	2005.000000	1.334543e+12	1.235788e+12	1.444079e+12	8.554096e+11	8.197528e+11	8.967612e+11
std	5965.433243	NaN	NaN	NaN	26.268513	9.148287e+12	8.610030e+12	9.789327e+12	6.286364e+12	6.041288e+12	6.585419e+12
min	1.000000	NaN	NaN	NaN	1960.000000	1.448063e+02	6.299026e+01	2.621094e+02	1.174979e+02	8.318772e+01	1.270468e+02
25%	63.000000	NaN	NaN	NaN	1982.000000	3.678736e+03	2.639116e+03	4.829886e+03	1.624411e+03	1.395430e+03	1.828575e+03
50%	125.500000	NaN	NaN	NaN	2005.000000	1.103640e+04	8.105541e+03	1.346178e+04	4.863298e+03	4.279291e+03	5.465731e+03
75%	183.000000	NaN	NaN	NaN	2028.000000	2.949281e+04	2.308992e+04	3.562660e+04	1.997525e+04	1.795003e+04	2.223434e+04
max	44578.000000	NaN	NaN	NaN	2050.000000	1.827414e+14	1.667007e+14	2.025062e+14	1.119468e+14	1.017185e+14	1.239708e+14

[309]:
```
df.dropna(inplace=True)
# usuwanie brakujących wartości (NA)
```

[310]:
```
df["level"]
# wybór kolumny, metoda 1
```

[310]:
```
0         Global
1         Global
2         Global
3         Global
4         Global
          ...
19469    Country
19470    Country
19471    Country
19472    Country
19473    Country
Name: level, Length: 18655, dtype: object
```

[311]:
```
df.level
# wybór kolumny, metoda 2
```

[311]:
```
0         Global
1         Global
2         Global
3         Global
4         Global
          ...
19469    Country
19470    Country
19471    Country
19472    Country
19473    Country
Name: level, Length: 18655, dtype: object
```

[312]:
```
df[["location_name","level","year"]]
# wybór kilku kolumn jednocześnie
```

[312]:

	location_name	level	year
0	Global	Global	1960
1	Global	Global	1961
2	Global	Global	1962
3	Global	Global	1963
4	Global	Global	1964
...	...	...	...
19469	Sudan	Country	2046
19470	Sudan	Country	2047
19471	Sudan	Country	2048
19472	Sudan	Country	2049
19473	Sudan	Country	2050

18655 rows × 3 columns

[313]:
```
df.loc[:, "gdp_ppp_mean":"gdp_ppp_upper"]
# wybierz wszystkie wiersze i kolumny od „gdp_ppp_mean” do „gdp_ppp_upper”
```

[313]:

	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper
0	1.748345e+13	1.601915e+13	1.911586e+13
1	1.813537e+13	1.659537e+13	1.982493e+13
2	1.895328e+13	1.739039e+13	2.061477e+13
3	1.965662e+13	1.811706e+13	2.134993e+13
4	2.100575e+13	1.935664e+13	2.276791e+13
...	...	...	...
19469	6.656899e+03	3.356042e+03	1.155051e+04
19470	6.729027e+03	3.374504e+03	1.171206e+04
19471	6.796123e+03	3.398699e+03	1.184386e+04
19472	6.866343e+03	3.417444e+03	1.196204e+04
19473	6.935555e+03	3.429198e+03	1.208179e+04

18655 rows × 3 columns

[314]:
```
df.loc[10:15, "level":"year"]
# wybierz wiersze od 10 do 15 (prawa krawędź, czyli liczba 15 jest uwzględniona w metodzie iloc),
# kolumny od „level” do „year”
```

[314]:

	level	year
10	Global	1970
11	Global	1971
12	Global	1972
13	Global	1973
14	Global	1974
15	Global	1975

[315]:
```
df.iloc[10:15, 0:3]
# wybierz wiersze od 10 do 15 (prawa krawędź, czyli liczba 15 nie jest uwzględniona w metodzie iloc),
# kolumny od 0 do 3
```

[315]:

	location_id	location_name	iso3
10	1	Global	G
11	1	Global	G
12	1	Global	G
13	1	Global	G
14	1	Global	G

[316]:
```
df[df["location_name"] == "Poland"]
# wybierz według warunku tylko te wiersze w kolumnie "location_name", w których wskazany jest kraj ("Poland")
```

[316]:

	location_id	location_name	iso3	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
3913	51	Poland	POL	Country	1960	6477.852541	3974.353115	8867.341510	3340.866044	3013.073267	3767.183362
3914	51	Poland	POL	Country	1961	6854.160388	4209.334194	9266.617081	3528.521854	3181.314168	3955.470420
3915	51	Poland	POL	Country	1962	6696.268564	4137.676353	9056.936779	3447.774153	3104.590497	3867.083569
3916	51	Poland	POL	Country	1963	6981.908383	4381.353870	9390.144726	3589.554120	3242.049335	4016.402456
3917	51	Poland	POL	Country	1964	7192.051449	4537.064677	9677.839801	3693.222182	3344.989514	4126.100385
...	...	...	...	...	...	...	...	...	...	...	...
3999	51	Poland	POL	Country	2046	38341.602497	28096.663520	51449.445517	17392.703365	12862.490537	23265.109294
4000	51	Poland	POL	Country	2047	38361.930630	27796.090370	52186.079877	17401.902620	12665.251631	23581.749549
4001	51	Poland	POL	Country	2048	38342.127372	27466.932992	52654.694631	17393.027956	12498.169215	23783.922827
4002	51	Poland	POL	Country	2049	38291.223003	27105.252343	53056.023696	17369.859621	12305.183049	24051.240960
4003	51	Poland	POL	Country	2050	38239.561147	26728.543779	53156.930801	17346.349035	12139.994778	24206.609198

91 rows × 11 columns

[317]:
```
df[df["level"] != "Country"]
# wybierz według warunku tylko te wiersze w kolumnie "level", w których nie jest wskazany kraj ("Country")
```

[317]:

	location_id	location_name	iso3	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
0	1	Global	G	Global	1960	1.748345e+13	1.601915e+13	1.911586e+13	1.296863e+13	1.266890e+13	1.334177e+13
1	1	Global	G	Global	1961	1.813537e+13	1.659537e+13	1.982493e+13	1.346097e+13	1.314767e+13	1.383021e+13
2	1	Global	G	Global	1962	1.895328e+13	1.739039e+13	2.061477e+13	1.406576e+13	1.376060e+13	1.443746e+13
3	1	Global	G	Global	1963	1.965662e+13	1.811706e+13	2.134993e+13	1.461831e+13	1.432132e+13	1.497693e+13
4	1	Global	G	Global	1964	2.100575e+13	1.935664e+13	2.276791e+13	1.552986e+13	1.523498e+13	1.587998e+13
...	...	...	...	...	...	...	...	...	...	...	...
86	1	Global	G	Global	2046	1.759560e+14	1.622744e+14	1.928964e+14	1.081513e+14	9.968511e+13	1.180625e+14
87	1	Global	G	Global	2047	1.778053e+14	1.635681e+14	1.952850e+14	1.091923e+14	1.003097e+14	1.197614e+14
88	1	Global	G	Global	2048	1.795422e+14	1.647031e+14	1.978349e+14	1.101656e+14	1.008704e+14	1.212579e+14
89	1	Global	G	Global	2049	1.811701e+14	1.657675e+14	2.003282e+14	1.110748e+14	1.012670e+14	1.226294e+14
90	1	Global	G	Global	2050	1.827414e+14	1.667007e+14	2.025062e+14	1.119468e+14	1.017185e+14	1.239708e+14

91 rows × 11 columns

[318]:
```
df[(df["location_name"] == "Poland") & (df["year"] >= 2000) & (df["year"] <= 2025)]
# kilka warunków na raz
```

[318]:

	location_id	location_name	iso3	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
3953	51	Poland	POL	Country	2000	16216.972354	14426.328563	17768.181829	7791.408953	7725.943403	7843.419625
3954	51	Poland	POL	Country	2001	16487.904946	14773.967146	17959.688784	7891.308758	7826.822228	7939.053706
3955	51	Poland	POL	Country	2002	16874.678183	15227.960046	18281.048016	8041.167697	7996.515127	8084.019154
3956	51	Poland	POL	Country	2003	17417.134853	15921.831120	18940.584365	8331.178693	8289.326227	8372.462204
3957	51	Poland	POL	Country	2004	18373.512432	16959.679703	19870.262632	8757.579659	8713.417331	8794.725970
3958	51	Poland	POL	Country	2005	19041.479311	17607.425075	20574.972690	9069.367143	9022.357802	9106.483639
3959	51	Poland	POL	Country	2006	20270.233606	18738.497355	21805.889040	9631.532462	9582.075276	9670.234609
3960	51	Poland	POL	Country	2007	21873.173630	20532.505755	23295.350478	10315.658078	10257.854072	10359.655937
3961	51	Poland	POL	Country	2008	22927.562639	21698.362309	24244.603222	10750.728983	10688.326402	10797.362341
3962	51	Poland	POL	Country	2009	23742.894202	22605.496876	24851.808333	11050.767254	10992.979795	11096.216843
3963	51	Poland	POL	Country	2010	24973.613215	23668.401544	25806.054124	11483.772332	11408.487175	11541.593953
3964	51	Poland	POL	Country	2011	26349.401036	25100.119954	27135.040741	12038.176112	11963.263559	12092.772107
3965	51	Poland	POL	Country	2012	26886.844683	25539.914861	28069.504151	12215.260691	12141.940413	12283.011267
3966	51	Poland	POL	Country	2013	27238.317883	25914.711448	28420.330330	12373.703619	12304.802636	12451.725249
3967	51	Poland	POL	Country	2014	28027.205917	26752.703925	28890.302427	12804.947700	12747.007553	12874.608912
3968	51	Poland	POL	Country	2015	29376.535311	27786.930870	30893.909749	13341.351456	13314.091883	13367.561843
3969	51	Poland	POL	Country	2016	30225.896887	28529.628460	31725.951218	13769.062646	13751.401158	13787.328914
3970	51	Poland	POL	Country	2017	31769.387501	29885.932286	33617.691730	14442.337586	14420.942773	14467.380536
3971	51	Poland	POL	Country	2018	33487.015221	31456.895715	35517.952761	15219.035737	15189.922175	15243.420231
3972	51	Poland	POL	Country	2019	34989.122460	32845.754009	37258.505218	15872.008849	15795.936919	15954.012310
3973	51	Poland	POL	Country	2020	33815.098332	31743.648412	36008.334279	15339.439866	15265.920452	15418.691785
3974	51	Poland	POL	Country	2021	32326.585411	30303.458800	34412.227000	14664.072938	14593.790447	14739.835543
3975	51	Poland	POL	Country	2022	32414.949511	30305.912649	34598.854634	14704.166798	14479.421379	14940.272545
3976	51	Poland	POL	Country	2023	32507.956803	30265.865868	34811.455280	14746.400178	14330.677311	15167.031872
3977	51	Poland	POL	Country	2024	32622.320619	30247.509184	35144.558305	14798.274235	14204.305762	15406.695352

[319]:
```
df["gdp_usd_round"] = df["gdp_usd_upper"].round(decimals = 1)
# utwórz nową kolumnę, która będzie miała zaokrąglone wartości z kolumny gdp_usd_upper
df.head()
```

[319]:

	location_id	location_name	iso3	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper	gdp_usd_round
0	1	Global	G	Global	1960	1.748345e+13	1.601915e+13	1.911586e+13	1.296863e+13	1.266890e+13	1.334177e+13	1.334177e+13
1	1	Global	G	Global	1961	1.813537e+13	1.659537e+13	1.982493e+13	1.346097e+13	1.314767e+13	1.383021e+13	1.383021e+13
2	1	Global	G	Global	1962	1.895328e+13	1.739039e+13	2.061477e+13	1.406576e+13	1.376060e+13	1.443746e+13	1.443746e+13
3	1	Global	G	Global	1963	1.965662e+13	1.811706e+13	2.134993e+13	1.461831e+13	1.432132e+13	1.497693e+13	1.497693e+13
4	1	Global	G	Global	1964	2.100575e+13	1.935664e+13	2.276791e+13	1.552986e+13	1.523498e+13	1.587998e+13	1.587998e+13

[320]:
```
df.drop("gdp_usd_round", axis=1, inplace = True)
# usuń kolumnę gdp_usd_round
```

[321]:
```
df.rename(columns = {"iso3": "location_code"}, inplace = True)
# zmień nazwę kolumny
df
```

[321]:

	location_id	location_name	location_code	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
0	1	Global	G	Global	1960	1.748345e+13	1.601915e+13	1.911586e+13	1.296863e+13	1.266890e+13	1.334177e+13
1	1	Global	G	Global	1961	1.813537e+13	1.659537e+13	1.982493e+13	1.346097e+13	1.314767e+13	1.383021e+13
2	1	Global	G	Global	1962	1.895328e+13	1.739039e+13	2.061477e+13	1.406576e+13	1.376060e+13	1.443746e+13
3	1	Global	G	Global	1963	1.965662e+13	1.811706e+13	2.134993e+13	1.461831e+13	1.432132e+13	1.497693e+13
4	1	Global	G	Global	1964	2.100575e+13	1.935664e+13	2.276791e+13	1.552986e+13	1.523498e+13	1.587998e+13
...	...	...	...	...	...	...	...	...	...	...	...
19469	522	Sudan	SDN	Country	2046	6.656899e+03	3.356042e+03	1.155051e+04	1.459547e+03	9.801683e+02	2.269566e+03
19470	522	Sudan	SDN	Country	2047	6.729027e+03	3.374504e+03	1.171206e+04	1.475378e+03	9.886902e+02	2.286933e+03
19471	522	Sudan	SDN	Country	2048	6.796123e+03	3.398699e+03	1.184386e+04	1.490021e+03	9.935248e+02	2.322390e+03
19472	522	Sudan	SDN	Country	2049	6.866343e+03	3.417444e+03	1.196204e+04	1.505368e+03	1.002889e+03	2.362591e+03
19473	522	Sudan	SDN	Country	2050	6.935555e+03	3.429198e+03	1.208179e+04	1.520564e+03	1.002953e+03	2.408108e+03

18655 rows × 11 columns

[322]:
```
df.to_csv("GDP_per_person_2.csv")
# zapisywanie dataframe do csv na komputerze
```

[323]:
```
df["gdp_ppp_mean"].mean()
# średnia z jednej kolumny
```

[323]:
```
np.float64(448950770593.6321)
```

[324]:
```
df['gdp_ppp_mean'].max()
# maksymalna wartość jedna kolumna na raz
```

[324]:
```
182741391837932.0
```

[325]:
```
df['gdp_ppp_mean'].min()
# minimalna wartość jedna kolumna na raz
```

[325]:
```
144.806256438462
```

[326]:
```
df['gdp_ppp_mean'].count()
# liczba rekordów
```

[326]:
```
np.int64(18655)
```

[327]:
```
df['gdp_ppp_mean'].unique()
# wartości unikatowe
```

[327]:
```
array([1.74834498e+13, 1.81353706e+13, 1.89532786e+13, ...,
       6.79612263e+03, 6.86634277e+03, 6.93555494e+03])
```

[328]:
```
df['gdp_ppp_mean'].value_counts()
# liczba rekordów pasujących do unikalnych wartości
```

[328]:
```
gdp_ppp_mean
6.935555e+03    1
1.748345e+13    1
1.813537e+13    1
1.895328e+13    1
1.965662e+13    1
               ..
3.884179e+13    1
3.758670e+13    1
3.613642e+13    1
3.484729e+13    1
3.325339e+13    1
Name: count, Length: 18655, dtype: int64
```

[329]:
```
df.sort_values(['year'], ascending = False)
# sortowanie malejąco
```

[329]:

	location_id	location_name	location_code	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
19473	522	Sudan	SDN	Country	2050	6.935555e+03	3.429198e+03	1.208179e+04	1.520564e+03	1.002953e+03	2.408108e+03
1091	16	Philippines	PHL	Country	2050	1.228046e+04	8.806731e+03	1.682555e+04	4.790026e+03	3.434351e+03	6.553565e+03
18381	367	Monaco	MCO	Country	2050	1.902880e+05	1.218498e+05	2.920071e+05	2.385452e+05	1.614090e+05	3.361224e+05
1182	17	Sri Lanka	LKA	Country	2050	1.962130e+04	1.386344e+04	2.731434e+04	5.451715e+03	3.878599e+03	7.461878e+03
909	14	Maldives	MDV	Country	2050	2.793245e+04	1.615196e+04	4.623992e+04	1.469534e+04	9.979461e+03	2.180008e+04
...	...	...	...	...	...	...	...	...	...	...	...
18018	320	Cook Islands	COK	Country	1960	2.053001e+03	1.740441e+03	2.340449e+03	3.302019e+03	2.582369e+03	3.919157e+03
14287	177	Djibouti	DJI	Country	1960	6.199516e+03	3.757012e+03	9.788527e+03	4.145554e+03	2.571869e+03	5.986814e+03
3276	44	Bosnia and Herzegovina	BIH	Country	1960	1.789672e+03	5.884096e+02	3.972539e+03	5.292547e+02	3.743372e+02	7.662351e+02
12922	160	Afghanistan	AFG	Country	1960	2.221336e+03	1.353293e+03	3.082416e+03	6.909928e+02	5.165132e+02	9.642339e+02
0	1	Global	G	Global	1960	1.748345e+13	1.601915e+13	1.911586e+13	1.296863e+13	1.266890e+13	1.334177e+13

18655 rows × 11 columns

[330]:
```
df.sort_values(['year'], ascending = True)
# sortowanie rosnąco
```

[330]:

	location_id	location_name	location_code	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
6461	83	Iceland	ISL	Country	1960	1.454238e+04	1.288146e+04	1.643550e+04	1.514908e+04	1.360613e+04	1.726783e+04
17290	213	Niger	NER	Country	1960	1.461266e+03	1.157773e+03	1.770731e+03	7.689299e+02	6.856744e+02	8.352938e+02
182	6	China	CHN	Country	1960	7.567040e+02	3.366123e+02	1.259304e+03	2.523051e+02	2.287723e+02	2.773206e+02
14105	175	Burundi	BDI	Country	1960	8.990588e+02	7.530268e+02	1.035991e+03	2.782921e+02	2.577184e+02	2.930667e+02
10101	128	Guatemala	GTM	Country	1960	3.763207e+03	2.446894e+03	4.720979e+03	2.099218e+03	1.952079e+03	2.298903e+03
...	...	...	...	...	...	...	...	...	...	...	...
90	1	Global	G	Global	2050	1.827414e+14	1.667007e+14	2.025062e+14	1.119468e+14	1.017185e+14	1.239708e+14
15105	185	Rwanda	RWA	Country	2050	3.410382e+03	2.378887e+03	4.830309e+03	1.149385e+03	8.091485e+02	1.603905e+03
12102	150	Oman	OMN	Country	2050	2.498143e+04	1.442400e+04	3.989478e+04	1.299342e+04	9.151710e+03	1.805737e+04
19473	522	Sudan	SDN	Country	2050	6.935555e+03	3.429198e+03	1.208179e+04	1.520564e+03	1.002953e+03	2.408108e+03
12466	154	Tunisia	TUN	Country	2050	1.308847e+04	9.199205e+03	1.800064e+04	4.105923e+03	2.891934e+03	5.637392e+03

18655 rows × 11 columns

[331]:
```
df.nlargest(10,'gdp_ppp_mean')
# 10 najwyższych wartości dla kolumny 'gdp_ppp_mean'
```

[331]:

	location_id	location_name	location_code	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
90	1	Global	G	Global	2050	1.827414e+14	1.667007e+14	2.025062e+14	1.119468e+14	1.017185e+14	1.239708e+14
89	1	Global	G	Global	2049	1.811701e+14	1.657675e+14	2.003282e+14	1.110748e+14	1.012670e+14	1.226294e+14
88	1	Global	G	Global	2048	1.795422e+14	1.647031e+14	1.978349e+14	1.101656e+14	1.008704e+14	1.212579e+14
87	1	Global	G	Global	2047	1.778053e+14	1.635681e+14	1.952850e+14	1.091923e+14	1.003097e+14	1.197614e+14
86	1	Global	G	Global	2046	1.759560e+14	1.622744e+14	1.928964e+14	1.081513e+14	9.968511e+13	1.180625e+14
85	1	Global	G	Global	2045	1.740498e+14	1.608327e+14	1.903320e+14	1.070717e+14	9.903290e+13	1.164315e+14
84	1	Global	G	Global	2044	1.720934e+14	1.594056e+14	1.874514e+14	1.059643e+14	9.831993e+13	1.147651e+14
83	1	Global	G	Global	2043	1.701152e+14	1.579438e+14	1.847172e+14	1.048522e+14	9.771637e+13	1.129875e+14
82	1	Global	G	Global	2042	1.681175e+14	1.566207e+14	1.817886e+14	1.037319e+14	9.697921e+13	1.115238e+14
81	1	Global	G	Global	2041	1.661209e+14	1.552230e+14	1.792966e+14	1.026157e+14	9.630379e+13	1.098151e+14

[332]:
```
df.nsmallest(10, 'gdp_ppp_mean')
# 10 minimalnych wartości 'gdp_ppp_mean'
```

[332]:

	location_id	location_name	location_code	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
15258	187	Somalia	SOM	Country	2021	144.806256	62.990256	262.109448	117.497898	106.885540	127.046786
15259	187	Somalia	SOM	Country	2022	145.845802	63.336551	264.032901	118.340834	107.636300	128.366648
15260	187	Somalia	SOM	Country	2023	147.061289	63.853934	266.073159	119.326124	108.370797	129.740750
15261	187	Somalia	SOM	Country	2024	148.359669	64.338452	268.348580	120.378255	108.942014	130.733717
15262	187	Somalia	SOM	Country	2025	149.840150	64.891911	271.234282	121.577851	109.871213	132.262378
15263	187	Somalia	SOM	Country	2026	151.357570	65.532653	275.428581	122.805601	110.436079	134.347813
15257	187	Somalia	SOM	Country	2020	151.493017	70.883163	266.749076	123.193355	112.066586	133.205105
15264	187	Somalia	SOM	Country	2027	152.720336	66.036475	277.528972	123.908711	111.144789	135.953024
15265	187	Somalia	SOM	Country	2028	154.008563	66.448878	279.879941	124.953754	111.824965	137.535891
15266	187	Somalia	SOM	Country	2029	155.240324	66.833274	282.488240	125.950266	112.307498	139.212056

[333]:
```
df[df['location_name'] == "Poland"].nlargest(10,'gdp_ppp_mean')
# 10 najwyższych wartości dla kolumny 'gdp_ppp_mean' w Polsce
```

[333]:

	location_id	location_name	location_code	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
4000	51	Poland	POL	Country	2047	38361.930630	27796.090370	52186.079877	17401.902620	12665.251631	23581.749549
4001	51	Poland	POL	Country	2048	38342.127372	27466.932992	52654.694631	17393.027956	12498.169215	23783.922827
3999	51	Poland	POL	Country	2046	38341.602497	28096.663520	51449.445517	17392.703365	12862.490537	23265.109294
3998	51	Poland	POL	Country	2045	38310.991457	28210.802867	50856.938961	17378.830577	12966.356828	22982.886727
4002	51	Poland	POL	Country	2049	38291.223003	27105.252343	53056.023696	17369.859621	12305.183049	24051.240960
3997	51	Poland	POL	Country	2044	38255.709502	28612.932167	50457.779813	17353.713535	13095.302179	22773.395829
4003	51	Poland	POL	Country	2050	38239.561147	26728.543779	53156.930801	17346.349035	12139.994778	24206.609198
3996	51	Poland	POL	Country	2043	38167.733724	28876.433219	49729.333605	17313.833385	13152.190918	22513.534843
3995	51	Poland	POL	Country	2042	38038.938282	29205.795211	49230.732911	17255.501259	13298.015373	22242.787940
3994	51	Poland	POL	Country	2041	37879.536633	29534.567189	48331.668321	17183.191423	13383.267219	21827.744208

[334]:
```
df.groupby('location_name').agg({'gdp_usd_mean' : 'mean'})
# grupowanie wierszy wedlug wartości kolumny kategoryzowanej, potem - uśrednienie wartości wszystkich kolumn w grupie - MultiIndex
```

[334]:

location_name	gdp_usd_mean
Afghanistan	507.846605
Albania	3433.776450
Algeria	2933.148414
American Samoa	15143.469699
Andorra	37802.770687
...	...
Venezuela (Bolivarian Republic of)	4885.044952
Viet Nam	1846.199971
Yemen	911.177936
Zambia	1034.843085
Zimbabwe	1045.095636

205 rows × 1 columns

[335]:
```
df_location_code = df.groupby('location_code').agg({'year': ['count'],
                                                    'gdp_usd_mean':['mean', 'median']})
# grupowanie wierszy według wartości kolumny kategoryzowanej, potem - uśrednienie wartości dla pewnych kolumn, liczba wartości i mediana dla pozostałych
df_location_code
```

[335]:

	year	gdp_usd_mean	
	count	mean	median
location_code			
AFG	91	507.846605	515.274036
AGO	91	2117.070792	2142.178837
ALB	91	3433.776450	3098.516205
AND	91	37802.770687	38178.372791
ARE	91	55966.449396	50856.888341
...	...	...	...
WSM	91	3516.104132	4024.794747
YEM	91	911.177936	828.806903
ZAF	91	4716.059123	4739.079636
ZMB	91	1034.843085	1078.009951
ZWE	91	1045.095636	1069.856772

205 rows × 3 columns

[336]:
```
df_location_code.index
# indeksy - lokalizacja skrót
```

[336]:
```
Index(['AFG', 'AGO', 'ALB', 'AND', 'ARE', 'ARG', 'ARM', 'ASM', 'ATG', 'AUS',
       ...
       'VCT', 'VEN', 'VIR', 'VNM', 'VUT', 'WSM', 'YEM', 'ZAF', 'ZMB', 'ZWE'],
      dtype='object', name='location_code', length=205)
```

[337]:
```
df_location_code.columns
# nazwy kolumn indeksu złożonego
```

[337]:
```
MultiIndex([(        'year',  'count'),
            ('gdp_usd_mean',   'mean'),
            ('gdp_usd_mean', 'median')],
           )
```

[338]:
```
df_location_code['gdp_usd_mean']['mean'].sort_values(ascending = False)
# sortowanie kolumny indeksu złożonego
```

[338]:
```
location_code
G      6.044914e+13
MCO    1.524926e+05
BMU    9.423244e+04
LUX    8.342488e+04
CHE    7.179691e+04
           ...
AFG    5.078466e+02
MWI    3.841131e+02
MOZ    3.833063e+02
BDI    3.166798e+02
SOM    1.807701e+02
Name: mean, Length: 205, dtype: float64
```

[339]:
```
df_pivot = df.pivot_table(values='gdp_usd_mean', index='location_code', columns='year', aggfunc='mean',
                          margins=False, dropna=True, fill_value=None)
# tabela przystawna (pivot table) na podstawie ramki danych
df_pivot
```

[339]:

year	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969
location_code										
AFG	690.992776	682.493782	678.584622	676.055466	672.157496	671.010257	662.090834	663.698124	668.950180	662.796770
AGO	1851.308305	2034.061930	1954.007569	2010.137252	2173.525060	2279.814704	2357.792342	2439.146375	2368.198756	2389.373947
ALB	1339.524075	1347.791424	1385.345352	1425.181598	1466.269618	1511.575360	1559.553181	1610.164222	1658.971736	1706.494834
AND	34118.166882	34489.345300	35025.128689	35443.391878	36579.598750	37130.928858	37769.938692	38178.372791	38935.426398	39758.240318
ARE	100082.068567	102905.168802	106962.945063	109853.261116	112251.670358	114012.841650	115119.031927	115491.243566	114477.080650	111005.598265
...	...	...	...	...	...	...	...	...	...	...
WSM	1920.092054	1950.023097	1989.575091	2022.721857	2097.445682	2139.011262	2186.012829	2219.950910	2274.596828	2333.577411
YEM	490.774951	492.519402	494.536840	497.938392	499.662633	502.489814	506.690052	511.426590	515.044691	519.748767
ZAF	3371.664637	3416.735294	3510.837899	3652.194627	3806.245512	3915.683211	3980.512033	4135.998491	4196.496647	4296.585406
ZMB	1026.198558	1008.297363	968.352629	973.514686	1055.455460	1166.911267	1121.420252	1170.681447	1160.436805	1147.665367
ZWE	800.772293	819.593227	805.962961	812.735502	796.908332	814.530841	798.895556	828.064000	812.589338	883.292514

205 rows × 91 columns

[340]:
```
df_pivot.index
# indeksy tabeli przystawnej
```

[340]:
```
Index(['AFG', 'AGO', 'ALB', 'AND', 'ARE', 'ARG', 'ARM', 'ASM', 'ATG', 'AUS',
       ...
       'VCT', 'VEN', 'VIR', 'VNM', 'VUT', 'WSM', 'YEM', 'ZAF', 'ZMB', 'ZWE'],
      dtype='object', name='location_code', length=205)
```

[341]:
```
df_pivot.columns
# kolumny tabeli przystawnej
```

[341]:
```
Index([1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971,
       1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983,
       1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995,
       1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007,
       2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019,
       2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031,
       2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043,
       2044, 2045, 2046, 2047, 2048, 2049, 2050],
      dtype='int64', name='year')
```

[342]:
```
df_pivot = df.pivot_table(values='gdp_usd_mean', index=['location_name','location_code'], columns='year', aggfunc='mean',
                          margins=False, dropna=True, fill_value=None)
# indeks złożony tabeli przystawnej
df_pivot
```

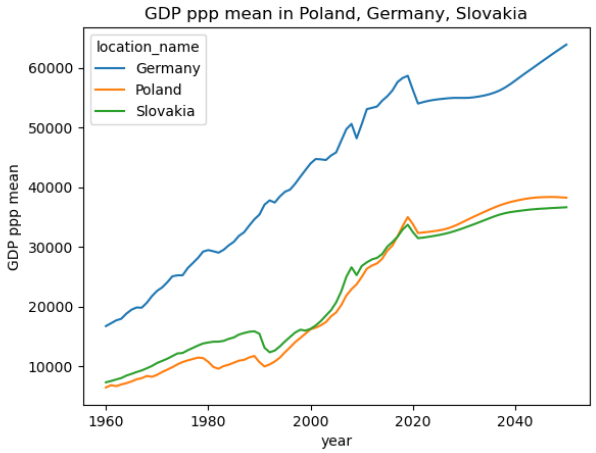
[342]:

	year	1960	1961	1962	1963	1964	1965	1966	1967	1968	19
location_name	location_code										
Afghanistan	AFG	690.992776	682.493782	678.584622	676.055466	672.157496	671.010257	662.090834	663.698124	668.950180	662.796
Albania	ALB	1339.524075	1347.791424	1385.345352	1425.181598	1466.269618	1511.575360	1559.553181	1610.164222	1658.971736	1706.494
Algeria	DZA	1939.599872	1700.333158	1393.253366	1709.272499	1753.611028	1811.619022	1683.353516	1784.656980	1899.555829	1993.504
American Samoa	ASM	21250.113556	21113.027633	21073.724259	20959.852345	21262.209427	21212.934410	21208.484996	21070.323415	21120.323220	21197.576
Andorra	AND	34118.166882	34489.345300	35025.128689	35443.391878	36579.598750	37130.928858	37769.938692	38178.372791	38935.426398	39758.240
...	...	...	...	...	...	...	...	...	...	...	...
Venezuela (Bolivarian Republic of)	VEN	5768.817697	5764.350615	5983.617969	6056.316181	6419.148302	6473.561248	6379.015524	6375.092819	6591.024839	6533.033
Viet Nam	VNM	408.871382	414.056177	446.221444	444.369557	449.710258	441.227988	432.742488	375.281778	360.466962	378.157
Yemen	YEM	490.774951	492.519402	494.536840	497.938392	499.662633	502.489814	506.690052	511.426590	515.044691	519.748
Zambia	ZMB	1026.198558	1008.297363	968.352629	973.514686	1055.455460	1166.911267	1121.420252	1170.681447	1160.436805	1147.665
Zimbabwe	ZWE	800.772293	819.593227	805.962961	812.735502	796.908332	814.530841	798.895556	828.064000	812.589338	883.292

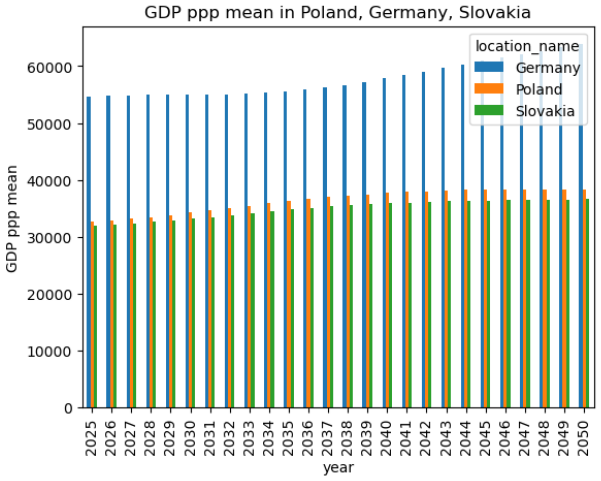
205 rows × 91 columns

[343]:
```
import matplotlib.pyplot as plt
# zaimportuj moduł pyplot z biblioteki matplotlib

%matplotlib inline
# wskazanie, że wykresy należy rysować bezpośrednio w zeszycie, a nie w osobnej zakładce
```

[344]:
```
df[(df['location_name'] == 'Poland') | (df['location_name'] == 'Germany') | (df['location_name'] == 'Slovakia')].pivot_table(values='gdp_ppp_mean', index
                                    fill_value=None, margins=False, dropna=True).plot(kind = 'line')

plt.ylabel('GDP ppp mean') # etykieta osi y
plt.title('GDP ppp mean in Poland, Germany, Slovakia') # tytuł wykresu
plt.show()
# wykres na podstawie tabeli przystawnej
```



[345]:
```
df_bar = df[(df['location_name'].isin(['Poland','Germany','Slovakia'])) & (df['year'] >= 2025)].pivot_table(values='gdp_ppp_mean',
                index='year', columns='location_name', aggfunc='mean',
                    fill_value=None, margins=False, dropna=True)
df_bar.plot(kind = 'bar')
plt.ylabel('GDP ppp mean')
plt.title('GDP ppp mean in Poland, Germany, Slovakia')
plt.show()
# histogram na podstawie wartości kolumny
```



[346]:
```
# sposoby łączenia ramek danych za pomocą metod merge i concat
```

[347]:
```
df1 = pd.read_csv('IHME_GDP_1960_2050_Y2021M09D22_TAB1.csv')
df1.head(10)
# wyświetlamy pierwszy 10 wierszy
```

[347]:

	location_id	location_name	iso3	level	year	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
0	1	Global	G	Global	1960	1.748345e+13	1.601915e+13	1.911586e+13	1.296863e+13	1.266890e+13	1.334177e+13
1	1	Global	G	Global	1961	1.813537e+13	1.659537e+13	1.982493e+13	1.346097e+13	1.314767e+13	1.383021e+13
2	1	Global	G	Global	1962	1.895328e+13	1.739039e+13	2.061477e+13	1.406576e+13	1.376060e+13	1.443746e+13
3	1	Global	G	Global	1963	1.965662e+13	1.811706e+13	2.134993e+13	1.461831e+13	1.432132e+13	1.497693e+13
4	1	Global	G	Global	1964	2.100575e+13	1.935664e+13	2.276791e+13	1.552986e+13	1.523498e+13	1.587998e+13
5	1	Global	G	Global	1965	2.202459e+13	2.034585e+13	2.382275e+13	1.628972e+13	1.598727e+13	1.663310e+13
6	1	Global	G	Global	1966	2.306193e+13	2.136085e+13	2.489782e+13	1.708885e+13	1.678223e+13	1.742396e+13
7	1	Global	G	Global	1967	2.391268e+13	2.217842e+13	2.577837e+13	1.770884e+13	1.740660e+13	1.804193e+13
8	1	Global	G	Global	1968	2.516723e+13	2.340479e+13	2.698215e+13	1.865379e+13	1.833216e+13	1.898399e+13
9	1	Global	G	Global	1969	2.642403e+13	2.464521e+13	2.831984e+13	1.955395e+13	1.921164e+13	1.987990e+13

[348]:
```
df2 = pd.read_csv('IHME_GDP_1960_2050_Y2021M09D22_TAB2.csv')
df2.head(10)
# wyświetlamy pierwszy 10 wierszy
```

[348]:

	location_id_2	location_name_2	iso3_2	level_2	year_2	gdp_ppp_mean_2	gdp_ppp_lower_2	gdp_ppp_upper_2	gdp_usd_mean_2	gdp_usd_lower_2	gdp_usd_upper_2
0	1	Global	G	Global	1960	1.748340e+13	1.601910e+13	1.911590e+13	1.296860e+13	1.266890e+13	1.334180e+13
1	1	Global	G	Global	1961	1.813540e+13	1.659540e+13	1.982490e+13	1.346100e+13	1.314770e+13	1.383020e+13
2	1	Global	G	Global	1962	1.895330e+13	1.739040e+13	2.061480e+13	1.406580e+13	1.376060e+13	1.443750e+13
3	1	Global	G	Global	1963	1.965660e+13	1.811710e+13	2.134990e+13	1.461830e+13	1.432130e+13	1.497690e+13
4	1	Global	G	Global	1964	2.100570e+13	1.935660e+13	2.276790e+13	1.552990e+13	1.523500e+13	1.588000e+13
5	1	Global	G	Global	1965	2.202460e+13	2.034580e+13	2.382280e+13	1.628970e+13	1.598730e+13	1.663310e+13
6	1	Global	G	Global	1966	2.306190e+13	2.136090e+13	2.489780e+13	1.708890e+13	1.678220e+13	1.742400e+13
7	1	Global	G	Global	1967	2.391270e+13	2.217840e+13	2.577840e+13	1.770880e+13	1.740660e+13	1.804190e+13
8	1	Global	G	Global	1968	2.516720e+13	2.340480e+13	2.698220e+13	1.865380e+13	1.833220e+13	1.898400e+13
9	1	Global	G	Global	1969	2.642400e+13	2.464520e+13	2.831980e+13	1.955390e+13	1.921160e+13	1.987990e+13

[349]:
```
df1.rename(columns = {'location_name': 'lokalizacja', 'year':'rok'}, inplace = True)
df2.rename(columns = {'location_name_2': 'lokalizacja_2', 'year_2': 'rok_2', 'gdp_ppp_mean_2': 'gdp_ppp_mean', 'gdp_usd_mean_2': 'gdp_usd_mean'}, inplace
# zmień nazwy kolumn zgodnie ze wskaźnikami: liczba utraconych lat produkcyjnych dla 1 ramki danych (DALY)
# i śmiertelność dla drugiej ramki danych ()
```

[350]:
```
df2.head(10)
```

[350]:

	location_id_2	lokalizacja_2	iso3_2	level_2	rok_2	gdp_ppp_mean	gdp_ppp_lower_2	gdp_ppp_upper_2	gdp_usd_mean	gdp_usd_lower_2	gdp_usd_upper_2
0	1	Global	G	Global	1960	1.748340e+13	1.601910e+13	1.911590e+13	1.296860e+13	1.266890e+13	1.334180e+13
1	1	Global	G	Global	1961	1.813540e+13	1.659540e+13	1.982490e+13	1.346100e+13	1.314770e+13	1.383020e+13
2	1	Global	G	Global	1962	1.895330e+13	1.739040e+13	2.061480e+13	1.406580e+13	1.376060e+13	1.443750e+13
3	1	Global	G	Global	1963	1.965660e+13	1.811710e+13	2.134990e+13	1.461830e+13	1.432130e+13	1.497690e+13
4	1	Global	G	Global	1964	2.100570e+13	1.935660e+13	2.276790e+13	1.552990e+13	1.523500e+13	1.588000e+13
5	1	Global	G	Global	1965	2.202460e+13	2.034580e+13	2.382280e+13	1.628970e+13	1.598730e+13	1.663310e+13
6	1	Global	G	Global	1966	2.306190e+13	2.136090e+13	2.489780e+13	1.708890e+13	1.678220e+13	1.742400e+13
7	1	Global	G	Global	1967	2.391270e+13	2.217840e+13	2.577840e+13	1.770880e+13	1.740660e+13	1.804190e+13
8	1	Global	G	Global	1968	2.516720e+13	2.340480e+13	2.698220e+13	1.865380e+13	1.833220e+13	1.898400e+13
9	1	Global	G	Global	1969	2.642400e+13	2.464520e+13	2.831980e+13	1.955390e+13	1.921160e+13	1.987990e+13

[351]:
```
df_all = pd.merge(df1, df2, on = ['gdp_ppp_mean', 'gdp_usd_mean'], how = 'outer')

# wszystkie wiersze ze wszystkich ramek danych, nie ma znaczenia, czy pasują, czy nie
# określiliśmy nazwy kolumn, według których łączymy dataframe'y, ale można też łączyć według indeksu -
# liczba porządkowa, która jest pozostawiona pionowo z [df1.index] bez cudzysłowów
```

[352]:
```
df_all.head()
```

[352]:

	location_id	lokalizacja	iso3	level	rok	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper	location_id_2	lokaliz
0	NaN	NaN	NaN	NaN	NaN	144.806256	NaN	NaN	117.497898	NaN	NaN	187.0	Sc
1	187.0	Somalia	SOM	Country	2021.0	144.806256	62.990256	262.109448	117.497898	106.88554	127.046786	NaN	
2	187.0	Somalia	SOM	Country	2022.0	145.845802	63.336551	264.032901	118.340834	107.63630	128.366648	NaN	
3	NaN	NaN	NaN	NaN	NaN	145.845802	NaN	NaN	118.340834	NaN	NaN	187.0	Sc
4	NaN	NaN	NaN	NaN	NaN	147.061289	NaN	NaN	119.326124	NaN	NaN	187.0	Sc

[353]:
```
df_all.shape
# ile wierszy i kolumn znajduje się w ramce danych
```

[353]:
```
(39676, 20)
```

[354]:
```
# podziel ramkę danych na dwie ramki danych: pierwsza z wierszami od pierwszego do 50 000, druga od 50 000 do ostatniego
df_all_1 = df_all.iloc[:50000,:]
df_all_2 = df_all.iloc[50000:,:]
```

[355]:
```
df_all_new = pd.concat([df_all_1, df_all_2], axis = 0) # połącz ramki danych: jeśli axis = 0, to po wierszach, jeśli
# axis = 1, potem według kolumn
df_all_new.shape # nowa dataframe ma taką samą liczbę wierszy i kolumn jak przed podziałem
```

[355]:
```
(39676, 20)
```

[356]:
```
# dodawanie nowych kolumn za pomocą operacji matematycznych
df_all["gdp_usd_mean"] = df_all["gdp_usd_mean"].round(decimals = 1)
```

[357]:
```
df_all
```

[357]:

	location_id	lokalizacja	iso3	level	rok	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper	location_id_2	lo
0	NaN	NaN	NaN	NaN	NaN	1.448063e+02	NaN	NaN	1.175000e+02	NaN	NaN	187.0	
1	187.0	Somalia	SOM	Country	2021.0	1.448063e+02	6.299026e+01	2.621094e+02	1.175000e+02	1.068855e+02	1.270468e+02	NaN	
2	187.0	Somalia	SOM	Country	2022.0	1.458458e+02	6.333655e+01	2.640329e+02	1.183000e+02	1.076363e+02	1.283666e+02	NaN	
3	NaN	NaN	NaN	NaN	NaN	1.458458e+02	NaN	NaN	1.183000e+02	NaN	NaN	187.0	
4	NaN	NaN	NaN	NaN	NaN	1.470613e+02	NaN	NaN	1.193000e+02	NaN	NaN	187.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
39671	1.0	Global	G	Global	2048.0	1.795422e+14	1.647031e+14	1.978349e+14	1.101656e+14	1.008704e+14	1.212579e+14	NaN	
39672	NaN	NaN	NaN	NaN	NaN	1.811700e+14	NaN	NaN	1.110750e+14	NaN	NaN	1.0	
39673	1.0	Global	G	Global	2049.0	1.811701e+14	1.657675e+14	2.003282e+14	1.110748e+14	1.012670e+14	1.226294e+14	NaN	
39674	NaN	NaN	NaN	NaN	NaN	1.827410e+14	NaN	NaN	1.119470e+14	NaN	NaN	1.0	
39675	1.0	Global	G	Global	2050.0	1.827414e+14	1.667007e+14	2.025062e+14	1.119468e+14	1.017185e+14	1.239708e+14	NaN	

39676 rows × 20 columns

[358]:
```
# dodanie nowej kolumny poprzez sumowanie wartości kilku innych
df_all["gdp_lower_total"] = df_all["gdp_ppp_lower"] + df_all["gdp_usd_lower"]
df_all
```

[358]:

	location_id	lokalizacja	iso3	level	rok	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	...	location_id_2	lokalizacja_2	i
0	NaN	NaN	NaN	NaN	NaN	1.448063e+02	NaN	NaN	1.175000e+02	NaN	...	187.0	Somalia	
1	187.0	Somalia	SOM	Country	2021.0	1.448063e+02	6.299026e+01	2.621094e+02	1.175000e+02	1.068855e+02	...	NaN	NaN	
2	187.0	Somalia	SOM	Country	2022.0	1.458458e+02	6.333655e+01	2.640329e+02	1.183000e+02	1.076363e+02	...	NaN	NaN	
3	NaN	NaN	NaN	NaN	NaN	1.458458e+02	NaN	NaN	1.183000e+02	NaN	...	187.0	Somalia	
4	NaN	NaN	NaN	NaN	NaN	1.470613e+02	NaN	NaN	1.193000e+02	NaN	...	187.0	Somalia	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
39671	1.0	Global	G	Global	2048.0	1.795422e+14	1.647031e+14	1.978349e+14	1.101656e+14	1.008704e+14	...	NaN	NaN	
39672	NaN	NaN	NaN	NaN	NaN	1.811700e+14	NaN	NaN	1.110750e+14	NaN	...	1.0	Global	
39673	1.0	Global	G	Global	2049.0	1.811701e+14	1.657675e+14	2.003282e+14	1.110748e+14	1.012670e+14	...	NaN	NaN	
39674	NaN	NaN	NaN	NaN	NaN	1.827410e+14	NaN	NaN	1.119470e+14	NaN	...	1.0	Global	
39675	1.0	Global	G	Global	2050.0	1.827414e+14	1.667007e+14	2.025062e+14	1.119468e+14	1.017185e+14	...	NaN	NaN	

39676 rows × 21 columns

[359]:
```
# dodawanie nowych kolumn z pomocą funkcji lambda

# tworzymy listę krajów
Countries = ['Poland', 'Germany', 'Slovakia', 'Denmark']

# za pomocą funkcji lambda określamy, że jeśli kraj („lokalizacja") jest zawarty w liście, to nowa kolumna będzie miała postać „Countries"
# będzie true, jeśli nie, to false
df_all['Countries'] = df_all['lokalizacja'].apply(lambda x: True if x in Countries else False )
df_all[df_all['Countries'] == True]
```

[359]:

	location_id	lokalizacja	iso3	level	rok	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	...	lokalizacja_2	iso3_2	level_2
14932	51.0	Poland	POL	Country	1960.0	6477.852541	3974.353115	8867.341510	3340.9	3013.073267	...	NaN	NaN	NaN
15242	51.0	Poland	POL	Country	1962.0	6696.268564	4137.676353	9056.936779	3447.8	3104.590497	...	NaN	NaN	NaN
15444	51.0	Poland	POL	Country	1961.0	6854.160388	4209.334194	9266.617081	3528.5	3181.314168	...	NaN	NaN	NaN
15619	51.0	Poland	POL	Country	1963.0	6981.908383	4381.353870	9390.144726	3589.6	3242.049335	...	NaN	NaN	NaN
15869	51.0	Poland	POL	Country	1964.0	7192.051449	4537.064677	9677.839801	3693.2	3344.989514	...	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
35890	81.0	Germany	DEU	Country	2050.0	63857.718915	46435.398410	88362.369420	52579.7	38567.414686	...	NaN	NaN	NaN
35909	78.0	Denmark	DNK	Country	2047.0	64212.626171	50281.471214	80411.199024	65065.9	51753.234814	...	NaN	NaN	NaN
35961	78.0	Denmark	DNK	Country	2048.0	64978.292714	50630.235130	82257.714049	65841.7	51817.244221	...	NaN	NaN	NaN
36035	78.0	Denmark	DNK	Country	2049.0	65789.718622	50459.797698	83832.916845	66664.1	51800.959712	...	NaN	NaN	NaN
36079	78.0	Denmark	DNK	Country	2050.0	66606.206732	50386.004245	85691.294417	67491.4	52050.141871	...	NaN	NaN	NaN

364 rows × 22 columns

[360]:
```
# wybierz wiersze, które pasują do nowego warunku i sprawdź, czy zostały pomyślnie dodane
df_all[df_all['Countries'] == True]
```

[360]:

	location_id	lokalizacja	iso3	level	rok	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	...	lokalizacja_2	iso3_2	level_2
14932	51.0	Poland	POL	Country	1960.0	6477.852541	3974.353115	8867.341510	3340.9	3013.073267	...	NaN	NaN	NaN
15242	51.0	Poland	POL	Country	1962.0	6696.268564	4137.676353	9056.936779	3447.8	3104.590497	...	NaN	NaN	NaN
15444	51.0	Poland	POL	Country	1961.0	6854.160388	4209.334194	9266.617081	3528.5	3181.314168	...	NaN	NaN	NaN
15619	51.0	Poland	POL	Country	1963.0	6981.908383	4381.353870	9390.144726	3589.6	3242.049335	...	NaN	NaN	NaN
15869	51.0	Poland	POL	Country	1964.0	7192.051449	4537.064677	9677.839801	3693.2	3344.989514	...	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
35890	81.0	Germany	DEU	Country	2050.0	63857.718915	46435.398410	88362.369420	52579.7	38567.414686	...	NaN	NaN	NaN
35909	78.0	Denmark	DNK	Country	2047.0	64212.626171	50281.471214	80411.199024	65065.9	51753.234814	...	NaN	NaN	NaN
35961	78.0	Denmark	DNK	Country	2048.0	64978.292714	50630.235130	82257.714049	65841.7	51817.244221	...	NaN	NaN	NaN
36035	78.0	Denmark	DNK	Country	2049.0	65789.718622	50459.797698	83832.916845	66664.1	51800.959712	...	NaN	NaN	NaN
36079	78.0	Denmark	DNK	Country	2050.0	66606.206732	50386.004245	85691.294417	67491.4	52050.141871	...	NaN	NaN	NaN

364 rows × 22 columns

[361]:
```
# zapisanie złożonego warunku w osobnej funkcji, a następnie zastosowanie go do żądanej kolumny
def if_cis(x):
    if x in Countries:
        return True
    else:
        False
```

[362]:
```
df_all['Countries'] = df_all['lokalizacja'].apply(if_cis)
```

[363]:
```
df_all[df_all['Countries'] == True]
```

[363]:

	location_id	lokalizacja	iso3	level	rok	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	...	lokalizacja_2	iso3_2	level_2
14932	51.0	Poland	POL	Country	1960.0	6477.852541	3974.353115	8867.341510	3340.9	3013.073267	...	NaN	NaN	NaN
15242	51.0	Poland	POL	Country	1962.0	6696.268564	4137.676353	9056.936779	3447.8	3104.590497	...	NaN	NaN	NaN
15444	51.0	Poland	POL	Country	1961.0	6854.160388	4209.334194	9266.617081	3528.5	3181.314168	...	NaN	NaN	NaN
15619	51.0	Poland	POL	Country	1963.0	6981.908383	4381.353870	9390.144726	3589.6	3242.049335	...	NaN	NaN	NaN
15869	51.0	Poland	POL	Country	1964.0	7192.051449	4537.064677	9677.839801	3693.2	3344.989514	...	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
35890	81.0	Germany	DEU	Country	2050.0	63857.718915	46435.398410	88362.369420	52579.7	38567.414686	...	NaN	NaN	NaN
35909	78.0	Denmark	DNK	Country	2047.0	64212.626171	50281.471214	80411.199024	65065.9	51753.234814	...	NaN	NaN	NaN
35961	78.0	Denmark	DNK	Country	2048.0	64978.292714	50630.235130	82257.714049	65841.7	51817.244221	...	NaN	NaN	NaN
36035	78.0	Denmark	DNK	Country	2049.0	65789.718622	50459.797698	83832.916845	66664.1	51800.959712	...	NaN	NaN	NaN
36079	78.0	Denmark	DNK	Country	2050.0	66606.206732	50386.004245	85691.294417	67491.4	52050.141871	...	NaN	NaN	NaN

364 rows × 22 columns

[364]:
```
# zapisz ramkę danych do csv
df_all.to_csv('df_all.csv')
```

[365]:
```
# możliwości pracy z dużymi plikami przy użyciu argumentu chunksize
```

[366]:
```
# podziel duży plik na części za pomocą argumentu chunksize, który określa, ile wierszy powinno znajdować się w każdej części
# wydrukuj 10 wierszy każdej wynikowej części

for chunk_df in pd.read_csv('df_all.csv',
                            chunksize = 50000):
    print("CHUNK DF")
    print(chunk_df.head(10))
```

```
CHUNK DF
   Unnamed: 0  location_id lokalizacja iso3    level     rok  gdp_ppp_mean  \
0           0          NaN         NaN  NaN      NaN     NaN    144.806256
1           1        187.0     Somalia  SOM  Country  2021.0    144.806256
2           2        187.0     Somalia  SOM  Country  2022.0    145.845802
3           3          NaN         NaN  NaN      NaN     NaN    145.845802
4           4          NaN         NaN  NaN      NaN     NaN    147.061289
5           5        187.0     Somalia  SOM  Country  2023.0    147.061289
6           6        187.0     Somalia  SOM  Country  2024.0    148.359669
7           7          NaN         NaN  NaN      NaN     NaN    148.359669
8           8          NaN         NaN  NaN      NaN     NaN    149.840150
9           9        187.0     Somalia  SOM  Country  2025.0    149.840150

   gdp_ppp_lower  gdp_ppp_upper  gdp_usd_mean  ...  lokalizacja_2  iso3_2  \
0            NaN            NaN         117.5  ...        Somalia     SOM
1      62.990256     262.109448         117.5  ...            NaN     NaN
2      63.336551     264.032901         118.3  ...            NaN     NaN
3            NaN            NaN         118.3  ...        Somalia     SOM
4            NaN            NaN         119.3  ...        Somalia     SOM
5      63.853934     266.073159         119.3  ...            NaN     NaN
6      64.338452     268.348580         120.4  ...            NaN     NaN
7            NaN            NaN         120.4  ...        Somalia     SOM
8            NaN            NaN         121.6  ...        Somalia     SOM
9      64.891911     271.234282         121.6  ...            NaN     NaN


   level_2   rok_2 gdp_ppp_lower_2 gdp_ppp_upper_2  gdp_usd_lower_2  \
0  Country  2021.0       62.990256       262.109448       106.885540
1      NaN     NaN             NaN             NaN              NaN
2      NaN     NaN             NaN             NaN              NaN
3  Country  2022.0       63.336551       264.032901       107.636300
4  Country  2023.0       63.853934       266.073159       108.370797
5      NaN     NaN             NaN             NaN              NaN
6      NaN     NaN             NaN             NaN              NaN
7  Country  2024.0       64.338452       268.348580       108.942014
8  Country  2025.0       64.891911       271.234282       109.871213
9      NaN     NaN             NaN             NaN              NaN

   gdp_usd_upper_2  gdp_lower_total  Countries
0       127.046786              NaN        NaN
1              NaN       169.875796        NaN
2              NaN       170.972852        NaN
3       128.366648              NaN        NaN
4       129.740750              NaN        NaN
5              NaN       172.224732        NaN
6              NaN       173.280467        NaN
7       130.733717              NaN        NaN
8       132.262378              NaN        NaN
9              NaN       174.763125        NaN

[10 rows x 23 columns]
```

[367]:
```
# zastosuj metodę groupby oddzielnie do każdej części, a następnie połącz wynik w nową ramkę danych
new_df = pd.DataFrame() # pusta ramka danych
for chunk_df in pd.read_csv('df_all.csv',
                            chunksize = 50000):
    result = chunk_df.groupby(['lokalizacja', 'rok']).agg({'gdp_usd_mean': 'mean',
                                                           'gdp_ppp_mean': 'mean'})
    new_df = pd.concat([new_df,result])
```

[368]:
```
new_df
```

[368]:

lokalizacja	rok	gdp_usd_mean	gdp_ppp_mean
Afghanistan	**1960.0**	691.0	2221.335586
	1961.0	682.5	2192.653614
	1962.0	678.6	2178.869688
	1963.0	676.1	2169.572283
	1964.0	672.2	2155.861769
...	...	...	...
Zimbabwe	**2046.0**	1283.1	3086.223490
	2047.0	1295.5	3116.294363
	2048.0	1307.8	3145.941464
	2049.0	1320.0	3175.312716
	2050.0	1332.2	3204.717710

19656 rows × 2 columns

[369]:
```
# zapisz wynik do pliku
new_df.to_csv('gdp_mean_in_countries.csv', encoding = 'utf-8')
```

3. Wnioski

Biblioteka Pandas pozwala w prosty i czytelny sposób przetwarzać i analizować duży zbiór danych. Dzięki zastosowaniu funkcji (groupby), połączeń (merge) można w szybki sposób uzyskać informacje w analizie danych ekonomicznych, tak jak w tym przypadku, rozkład PKB w poszczególnych krajach i latach. Pandas pozwala wygenerować wykresy i histogramy, pomaga w przejrzysty sposób zwizualizować dane z różnych perspektyw, co przyśpiesza pracę analityczną.