SPRAWOZDANIE

Zajęcia: Eksploracja i wizualizacja danych
Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 1
30.10.2021
Temat: "Wstęp do Python. Biblioteka Pandas"

Marek Bubiak
Informatyka II stopień,
niestacjonarne (zaoczne),
III semestr,
https://github.com/JeremyCoco/eiwd

1. Ładowanie biblioteki Pandas

```
In [1]: # ładowanie biblioteki Pandas
        import pandas as pd
```

2. Tworzenie ramki danych ze słownika

```
In [2]: # tworzenie ramki danych ze słownika

        dane = {
            "Numer" : [1,2,3,4,5,6,7],
            "Dzien" : ["Poniedziałek","Wtorek","Środa","Czwartek","Piątek","Sobota","Niedziela"]
        }

        df = pd.DataFrame(dane)
        df
```

Out[2]:

	Numer	Dzien
0	1	Poniedziałek
1	2	Wtorek
2	3	Środa
3	4	Czwartek
4	5	Piątek
5	6	Sobota
6	7	Niedziela

3. Zachowanie ramki danych pobranych z pliku w formacie csv (xlsx)

```
In [4]: # zachowanie ramki danych na komputerze w formacie csv
        path = r"C:\Users\Dzikus\Downloads\daneZeSlownika.csv"
        df.to_csv(path, encoding="utf-8")
```

4. Tworzenie ramki danych z listy list

```
In [5]: # tworzenie ramki danych z listy list
        week_days = [
            [1,2,3,4,5,6,7],
            ["Poniedziałek","Wtorek","Środa","Czwartek","Piątek","Sobota","Niedziela"]
        ]

        pd.DataFrame(week_days)
```

Out[5]:

	0	1	2	3	4	5	6
0	1	2	3	4	5	6	7
1	Poniedziałek	Wtorek	Środa	Czwartek	Piątek	Sobota	Niedziela

5. Transponowanie (wymieniamy kolumny a wierszy)

```
In [6]: # transponowanie (wymieniamy kolumny a wierszy)
        pd.DataFrame(week_days).T
```

Out[6]:

	0	1
0	1	Poniedziałek
1	2	Wtorek
2	3	Środa
3	4	Czwartek
4	5	Piątek
5	6	Sobota
6	7	Niedziela

6. Wyświetlić pierwsze 10 wierszy ramki danych

```
In [15]: #wczytanie danych z pliku *.csv
path = r"C:\Users\Dzikus\Downloads\IHME_GDP_1960_2050_CSV_1\IHME_GDP_1960_2050_Y2021M09D22.CSV"

df = pd.read_csv(path, low_memory=False, )

# pierwsze 10 wierszy ramki danych
df.head(10)
```

Out[15]:

	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

7. Wyświetlić ostatnie 10 wierszy ramki danych

```
In [16]: # ostatnie 10 wierszy ramki danych
df.tail(10)
```

Out[16]:

	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

8. Wyświetlić informację o ramce danych

```
In [18]: # informacja o ramce danych
df.info()
```

```
<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
```

9. Wyświetlić, ile wierszy i kolumn znajduje się w ramce danych

```
In [19]: # pokazuje, ile wierszy i kolumn znajduje się w ramce danych
         df.shape

Out[19]: (19838, 11)
```

10. Wyświetlić informację statystyczną o kolumnach liczbowych (wartości niepowtarzalne, średnia, odchylenie standardowe, minimum, kwartyle, maksimum)

```
In [20]: # informacje statystyczne w kolumnach (wartości niepowtarzalne,
         # średnia, odchylenie standardowe, minimum, kwartyle, maksimum)
         df.describe()
```

Out[20]:

	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

11. Wyświetlić informację statystyczną o kolumnach kategoryzowanych (ile unikalnych wartości, top - jaka jest najpopularniejsza wartość, freq - jak często najpopularniejsza)

```
In [21]: #statystyki obejmują nie tylko kolumny liczbowe, ale także wiersze
         # (unique - ile unikalnych wartości, top - jaka jest najpopularniejsza wartość,
         # freq - jak często najpopularniejsza)
         df.describe(include = 'all')
```

Out[21]:

	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	AGO	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

12. Usunąć brakujące wartości w ramce danych

```
In [23]: # usuwanie brakujących wartości (NA)
         df.dropna(inplace=True)
         df.head()
```

Out[23]:

	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

13. Przedstawić wybór wierszy z ramki danych pod warunkiem odnośnie określonej wartości kolumny

```
In [24]: df[df["year"] == 1960]
```

Out[24]:

	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
182	6	China	CHN	Country	1960	7.567040e+02	3.366123e+02	1.259304e+03	2.523051e+02	2.287723e+02	2.773206e+02
273	7	Democratic People's Republic of Korea	PRK	Country	1960	3.464463e+03	2.905950e+03	3.942335e+03	2.260950e+03	2.166923e+03	2.390282e+03
364	8	Taiwan (Province of China)	TWN	Country	1960	2.791608e+03	2.227734e+03	3.645526e+03	1.693585e+03	1.674297e+03	1.711836e+03
455	10	Cambodia	KHM	Country	1960	1.577499e+03	1.019173e+03	2.219433e+03	5.764952e+02	3.563508e+02	7.586329e+02
...	...	...	...	...	...	...	...	...	...	...	...
19019	413	Tokelau	TKL	Country	1960	1.465968e+03	1.216908e+03	1.697964e+03	6.869542e+02	6.703647e+02	7.024712e+02
19110	416	Tuvalu	TUV	Country	1960	1.992716e+03	1.812297e+03	2.185372e+03	1.715644e+03	1.563543e+03	1.832102e+03
19201	422	United States Virgin Islands	VIR	Country	1960	1.140270e+04	1.063712e+04	1.207289e+04	1.129532e+04	1.012697e+04	1.242223e+04
19292	435	South Sudan	SSD	Country	1960	2.128791e+03	1.595640e+03	2.574858e+03	7.197482e+02	6.481443e+02	7.772999e+02
19383	522	Sudan	SDN	Country	1960	2.547179e+03	1.644073e+03	3.628642e+03	6.338828e+02	6.013645e+02	6.652940e+02

205 rows × 11 columns

14. Przedstawić wybór wierszy z ramki danych pod warunkiem spełnienia kilku warunków jednocześnie

```
In [29]: df[(df["level"] == "Country") & (df["year"] == 1961)]
```

Out[29]:

	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
183	6	China	CHN	Country	1961	643.349774	269.768498	1106.862096	203.703824	176.825230	236.687872
274	7	Democratic People's Republic of Korea	PRK	Country	1961	3450.020864	2934.947713	3914.786863	2244.690491	2152.405274	2373.724393
365	8	Taiwan (Province of China)	TWN	Country	1961	2872.660145	2311.020738	3705.664528	1734.474519	1711.173153	1757.078727
456	10	Cambodia	KHM	Country	1961	1525.145382	979.899529	2159.802183	557.449086	345.739851	736.307040
547	11	Indonesia	IDN	Country	1961	1623.539644	852.198938	2320.768157	597.860868	531.262847	672.842421
...	...	...	...	...	...	...	...	...	...	...	...
19020	413	Tokelau	TKL	Country	1961	1525.645285	1275.694453	1755.612875	714.790698	697.518393	731.851761
19111	416	Tuvalu	TUV	Country	1961	2025.825111	1840.634875	2220.899794	1744.927428	1592.756979	1865.579295
19202	422	United States Virgin Islands	VIR	Country	1961	11461.189150	10688.949084	12168.794958	11364.079294	10218.326201	12466.341606
19293	435	South Sudan	SSD	Country	1961	2135.201430	1605.047069	2571.515461	722.115635	649.983727	780.481136
19384	522	Sudan	SDN	Country	1961	2482.585119	1612.927210	3533.336963	617.083664	585.486006	645.946618

204 rows × 11 columns

15. Wybrać wiersze które zawierają w kolumnie kategoryzowanej określone słowo

```
In [30]: df[df["location_name"] == "Sudan"]
```

Out[30]:

	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
19383	522	Sudan	SDN	Country	1960	2547.179302	1644.073039	3628.641669	633.882757	601.364519	665.294016
19384	522	Sudan	SDN	Country	1961	2482.585119	1612.927210	3533.336963	617.083664	585.486006	645.946618
19385	522	Sudan	SDN	Country	1962	2574.844128	1695.153232	3627.690114	639.502180	607.460746	669.311605
19386	522	Sudan	SDN	Country	1963	2441.718632	1607.123912	3463.482637	605.459753	576.710823	630.304619
19387	522	Sudan	SDN	Country	1964	2355.692315	1566.218099	3351.024821	582.962548	556.338676	605.232759
...	...	...	...	...	...	...	...	...	...	...	...
19469	522	Sudan	SDN	Country	2046	6656.899075	3356.042298	11550.507119	1459.547314	980.168323	2269.566302
19470	522	Sudan	SDN	Country	2047	6729.026669	3374.504195	11712.060216	1475.378472	988.690170	2286.932826
19471	522	Sudan	SDN	Country	2048	6796.122627	3398.698859	11843.857084	1490.020809	993.524823	2322.389652
19472	522	Sudan	SDN	Country	2049	6866.342766	3417.443728	11962.042860	1505.368205	1002.889036	2362.591145
19473	522	Sudan	SDN	Country	2050	6935.554937	3429.197754	12081.785960	1520.563732	1002.953346	2408.108095

91 rows × 11 columns

16. Wybrać wiersze które nie zawierają w kolumnie kategoryzowanej określone słowo

```
In [35]: selection = df[df["location_name"] != "Asia"]
         selection.head()
```

Out[35]:

	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

17. Utwórz kolumnę na podstawie istniejącej

```
In [36]: location_name = df.location_name
         location_name
```

```
Out[36]: 0        Global
         1        Global
         2        Global
         3        Global
         4        Global
                   ...
         19469     Sudan
         19470     Sudan
         19471     Sudan
         19472     Sudan
         19473     Sudan
         Name: location_name, Length: 18655, dtype: object
```

18. Usuń kolumnę

```
In [38]: df_copy = df
         df_copy.drop(["location_name", "iso3"], axis=1, inplace=True)
         df_copy.head()
```

Out[38]:

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

19. Zmień nazwę kolumny

```
In [39]: df_copy.rename(columns={"year": "rok", "location_id": "id"}, inplace=True)
         df_copy.head()
```

Out[39]:

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

20. Zachowaj ramkę danych jako plik csv na komputerze

```
In [40]: path = r"C:\Users\Dzikus\Downloads\df_copy.csv"
         df.to_csv(path, encoding="utf-8")
```

21. Wyświetlić średnia (maksymalną, minimalną) wartość z jednej kolumny

```
In [42]: col = df["gdp_ppp_mean"]
         mean = col.mean()
         _max = col.max()
         _min = col.min()

         print(f"Średnia: {mean}\nMaksimum: {_max}\nMinimum: {_min}")
```

```
Średnia: 448950770593.6332
Maksimum: 182741391837932.0
Minimum: 144.806256438462
```

22. Wyświetlić liczbę wierszy

```
In [43]: df.rok.count()
Out[43]: 18655
```

23. Wyświetlić wartości unikatowe w kolumnie

```
In [47]: df["rok"].unique()
Out[47]: array([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)
```

24. Wyświetlić liczby rekordów odpowiadających do wartości

```
In [50]: df_copy[df_copy["id"] == 187].rok.count()
Out[50]: 91
```

25. Sortowanie wierszy ramki danych według wartości określonej kolumny (malejąco, rosnąco)

```
In [49]: df_copy.sort_values(["gdp_ppp_upper"], ascending=True).head()
```

Out[49]:

	id	level	rok	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
15258	187	Country	2021	144.806256	62.990256	262.109448	117.497898	106.885540	127.046786
15259	187	Country	2022	145.845802	63.336551	264.032901	118.340834	107.636300	128.366648
15260	187	Country	2023	147.061289	63.853934	266.073159	119.326124	108.370797	129.740750
15257	187	Country	2020	151.493017	70.883163	266.749076	123.193355	112.066586	133.205105
15261	187	Country	2024	148.359669	64.338452	268.348580	120.378255	108.942014	130.733717

26. Wyświetlić wierszy dla 10 największych (najmniejszych) wartości określonej kolumny

```
In [54]: df.nlargest(10, 'gdp_ppp_mean')[['rok', 'gdp_ppp_mean']]
```

Out[54]:

	rok	gdp_ppp_mean
90	2050	1.827414e+14
89	2049	1.811701e+14
88	2048	1.795422e+14
87	2047	1.778053e+14
86	2046	1.759560e+14
85	2045	1.740498e+14
84	2044	1.720934e+14
83	2043	1.701152e+14
82	2042	1.681175e+14
81	2041	1.661209e+14

27. Wyświetlić wierszy dla 10 największych wartości określonej kolumny pod warunkiem określonych wartości innej kolumny

```
In [57]: df[(df['id'].isin([187, 192])) & (df['rok'] == 2003)]
```

Out[57]:

	id	level	rok	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
15240	187	Country	2003	165.077376	75.241513	281.85689	134.639462	125.515319	143.91606

28. Grupowanie wierszy według wartości kolumny kategoryzowanej, potem - uśrednienie wartości wszystkich kolumn w grupie – MultiIndex

```
In [58]: df.groupby('rok').agg('mean')
```

Out[58]:

rok	id	gdp_ppp_mean	gdp_ppp_lower	gdp_ppp_upper	gdp_usd_mean	gdp_usd_lower	gdp_usd_upper
1960	135.639024	8.528513e+10	7.814218e+10	9.324812e+10	6.326159e+10	6.179953e+10	6.508179e+10
1961	135.639024	8.846523e+10	8.095304e+10	9.670697e+10	6.566329e+10	6.413496e+10	6.746446e+10
1962	135.639024	9.245503e+10	8.483118e+10	1.005599e+11	6.861346e+10	6.712486e+10	7.042663e+10
1963	135.639024	9.588596e+10	8.837590e+10	1.041460e+11	7.130884e+10	6.986011e+10	7.305819e+10
1964	135.639024	1.024671e+11	9.442264e+10	1.110630e+11	7.575543e+10	7.431699e+10	7.746333e+10
...	...	...	...	...	...	...	...
2046	135.639024	8.583220e+11	7.915827e+11	9.409579e+11	5.275673e+11	4.862689e+11	5.759145e+11
2047	135.639024	8.673428e+11	7.978932e+11	9.526097e+11	5.326453e+11	4.893155e+11	5.842021e+11
2048	135.639024	8.758158e+11	8.034297e+11	9.650481e+11	5.373930e+11	4.920505e+11	5.915018e+11
2049	135.639024	8.837564e+11	8.086222e+11	9.772106e+11	5.418284e+11	4.939856e+11	5.981921e+11
2050	135.639024	8.914214e+11	8.131744e+11	9.878353e+11	5.460821e+11	4.961880e+11	6.047354e+11

91 rows × 7 columns

29. Grupowanie wierszy według wartości kolumny kategoryzowanej, potem - uśrednienie wartości dla pewnych kolumn, liczba wartości i mediana dla pozostałych kolumn w grupach

```
In [70]: data = df.groupby('location_name').agg({'location_id': ['mean'], 'gdp_ppp_mean': ['mean'],
         'gdp_ppp_lower': ['mean'], 'year': ['count'], 'gdp_ppp_upper':
         ['median'], 'gdp_usd_mean': ['median']})
         data
```

Out[70]:

	location_id	gdp_ppp_mean	gdp_ppp_lower	year	gdp_ppp_upper	gdp_usd_mean
	mean	mean	mean	count	median	median
location_name						
Afghanistan	160.0	1941.160286	1236.392538	91	2776.309765	515.274036
Albania	43.0	9092.515182	7497.502508	91	9075.499017	3098.516205
Algeria	139.0	8820.271149	6354.741822	91	11218.304481	3163.885729
American Samoa	298.0	15340.365197	13676.178347	91	15350.704406	13620.772462
Andorra	74.0	25139.562251	19212.344640	91	34824.933478	38178.372791
...	...	...	...	...	...	...
Venezuela (Bolivarian Republic of)	133.0	10594.142490	6906.942146	91	16306.155638	5823.785745
Viet Nam	20.0	5737.873614	3963.905853	91	5395.236220	1437.919432
Yemen	157.0	2637.237249	1253.872401	91	4512.871947	828.806903
Zambia	191.0	3107.029470	2256.455986	91	4016.872139	1078.009951
Zimbabwe	198.0	2925.918096	2053.892011	91	3621.537763	1069.856772

216 rows × 6 columns

30. Wyświetlić nazwy kolumn indeksu złożonego

```
In [71]: data.index
```

```
Out[71]: Index(['Afghanistan', 'Albania', 'Algeria', 'American Samoa', 'Andorra',
                'Angola', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Australia',
                ...
                'United States of America', 'Upper-middle income', 'Uruguay',
                'Uzbekistan', 'Vanuatu', 'Venezuela (Bolivarian Republic of)',
                'Viet Nam', 'Yemen', 'Zambia', 'Zimbabwe'],
               dtype='object', name='location_name', length=216)
```

31. Sortować kolumnę indeksu złożonego

```
In [73]: data['gdp_ppp_lower']['mean'].sort_values(ascending=False)
```

```
Out[73]: location_name
         Global                                  8.770511e+13
         High income                             4.296017e+13
         High-income                             3.910369e+13
         Upper-middle income                     2.770854e+13
         Southeast Asia, East Asia, and Oceania  1.644275e+13
                                                     ...
         Niger                                   9.053953e+02
         Mozambique                              8.709698e+02
         Burundi                                 8.240844e+02
         Malawi                                  7.925530e+02
         Somalia                                 9.249822e+01
         Name: mean, Length: 216, dtype: float64
```

32. Stworzyć tabelę przystawną (pivot table) na podstawie ramki danych

```
In [75]: pivot = df.pivot_table(values='gdp_ppp_mean', index='location_name', columns='year',aggfunc='count', margins=False, dropna=True, fil
         pivot
```

Out[75]:

year	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	...	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050
location_name																					
Afghanistan	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Albania	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Algeria	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
American Samoa	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Andorra	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Venezuela (Bolivarian Republic of)	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Viet Nam	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Yemen	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Zambia	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Zimbabwe	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1

216 rows × 91 columns

33. Wyświetlić indeksy i kolumny tabeli przystawnej

```
In [76]: pivot.index
```

```
Out[76]: Index(['Afghanistan', 'Albania', 'Algeria', 'American Samoa', 'Andorra',
                'Angola', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Australia',
                ...
                'United States of America', 'Upper-middle income', 'Uruguay',
                'Uzbekistan', 'Vanuatu', 'Venezuela (Bolivarian Republic of)',
                'Viet Nam', 'Yemen', 'Zambia', 'Zimbabwe'],
               dtype='object', name='location_name', length=216)
```

34. Utwórz indeks złożony tabeli przystawnej i wyświetl go

```
In [78]: pivot2 = df.pivot_table(values='gdp_ppp_mean', index=['location_name', 'location_id'], columns='year',
                               aggfunc='count', margins=False, dropna=True, fill_value=None)
         pivot2
```

Out[78]:

year		1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	...	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050
location_name	location_id																					
Afghanistan	160	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Albania	43	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Algeria	139	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
American Samoa	298	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Andorra	74	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Venezuela (Bolivarian Republic of)	133	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Viet Nam	20	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Yemen	157	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Zambia	191	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
Zimbabwe	198	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1

218 rows × 91 columns

35. Zaimportuj moduł pyplot z biblioteki matplotlib

```
In [79]: import matplotlib.pyplot as plt

         %matplotlib inline
```

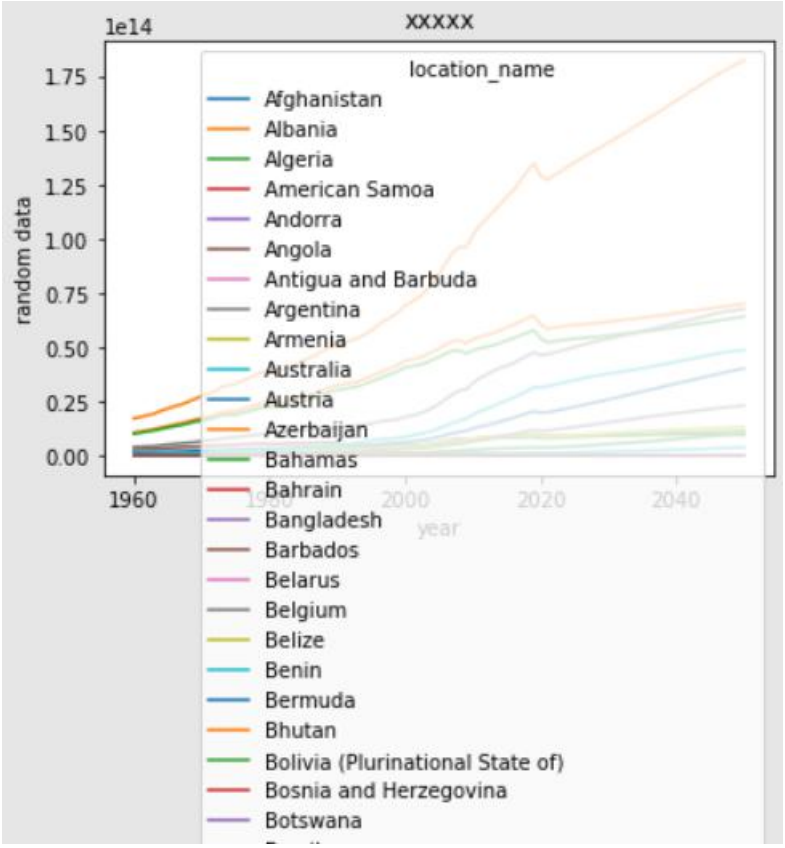
36. Wskazać, że wykresy należy rysować bezpośrednio w zeszycie, a nie w osobnej zakładce

```
In [79]: import matplotlib.pyplot as plt

         %matplotlib inline
```

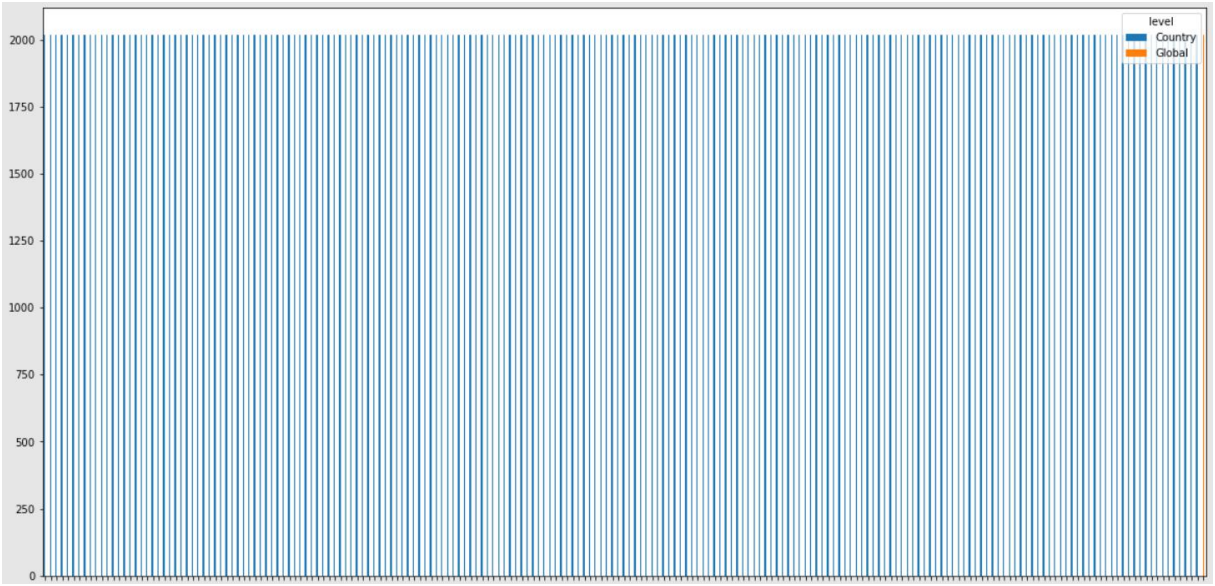
37. Wyświetlić wykres na podstawie tabeli przystawnej

```
In [83]: pivot3 = df.pivot_table(values='gdp_ppp_mean', index='year', columns='location_name', aggfunc='mean',
                                 margins=False, dropna=True, fill_value=None)
         fig = pivot3.plot(kind='line')
         plt.ylabel('random data')
         plt.title('xxxxx')
         plt.rcParams["figure.figsize"] = (20,10)
         #display(fig)
         #plt.show()
```



38. Narysować histogram na podstawie wartości kolumny

```
In [57]: df_bar = df[(df['level'].isin(['Global','Country'])) & (df['year'] == 2019)].pivot_table(values='year',
                 index='gdp_ppp_mean', columns='level', aggfunc='mean',
                       fill_value=None, margins=False, dropna=True)
         df_bar.plot(kind = 'bar')
         plt.ylabel('')
         plt.title('')
```



39. Pokazać dodawanie nowych kolumn za pomocą operacji matematycznych

```
In [64]: df['sum'] = df['gdp_ppp_mean'] + df['gdp_ppp_lower']
         df[['gdp_ppp_mean', 'gdp_ppp_lower', 'sum']].tail()
```

Out[64]:

	gdp_ppp_mean	gdp_ppp_lower	sum
19833	3.617310e+12	3.140835e+12	6.758144e+12
19834	3.724063e+12	3.225849e+12	6.949912e+12
19835	3.831942e+12	3.307609e+12	7.139551e+12
19836	3.941856e+12	3.398884e+12	7.340739e+12
19837	4.053883e+12	3.482933e+12	7.536816e+12

40. Przedstawić na przykładzie dodawanie nowych kolumn z pomocą funkcji lambda

In [65]:
```
df['years_ago'] = df['year'].apply(lambda y: 2051 - int(y))
df[['year', 'years_ago']]
```

Out[65]:

	year	years_ago
0	1960	91
1	1961	90
2	1962	89
3	1963	88
4	1964	87
...	...	...
19833	2046	5
19834	2047	4
19835	2048	3
19836	2049	2
19837	2050	1

19838 rows × 2 columns

41. Przedstawić możliwości pracy z dużymi plikami przy użyciu argumentu chunksize

In [68]:
```
path = r"C:\Users\Dzikus\Downloads\IHME_GDP_1960_2050_CSV_1\IHME_GDP_1960_2050_Y2021M09D22.CSV"

chunks = pd.read_csv(path, low_memory=False, chunksize=10_000)

for i, chunk in enumerate(chunks):
    print(f"\n\nChunk number {i+1}:")
    print(chunk.iloc[0:3,0:4])
```

```
Chunk number 1:
   location_id location_name iso3   level
0            1        Global    G  Global
1            1        Global    G  Global
2            1        Global    G  Global


Chunk number 2:
       location_id location_name iso3    level
10000          126    Costa Rica  CRI  Country
10001          126    Costa Rica  CRI  Country
10002          126    Costa Rica  CRI  Country
```