

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

```
In [2]: # 2. Tworzenie ramki danych ze słownika
dane_slownik = {
    "Kraj": ["USA", "Kanada", "Meksyk"],
    "Ludność (mln)": [331, 38, 128],
    "PKB (bln USD)": [23, 2, 1.3]
}

ramka_danych_slownik = pd.DataFrame(dane_slownik)

print("Ramka danych utworzona ze słownika:")
print(ramka_danych_slownik)
```

Ramka danych utworzona ze słownika:

	Kraj	Ludność (mln)	PKB (bln USD)
0	USA	331	23.0
1	Kanada	38	2.0
2	Meksyk	128	1.3

```
In [5]: # 3. Wczytanie danych z pliku CSV
plik_csv = "IHME_USA_RISK_SPENDING_2016_Y2020M09D29.csv"
ramka_danych_csv = pd.read_csv(plik_csv)

# Wyświetlenie pierwszych wierszy ramki danych
print("Ramka danych wczytana z pliku CSV:")
print(ramka_danych_csv.head())
```

Ramka danych wczytana z pliku CSV:

	location_id	location_name	year	sex_id	sex	age_group_id	\
0	102	United States of America	2016	1	Male	158	
1	102	United States of America	2016	1	Male	170	
2	102	United States of America	2016	1	Male	171	
3	102	United States of America	2016	1	Male	154	
4	102	United States of America	2016	1	Male	22	

	age_group_name	acause	cause_name	risk_id	\
0	<20 years	_all	All causes	82	
1	20 to 44	_all	All causes	82	
2	45 to 64	_all	All causes	82	
3	65 plus	_all	All causes	82	
4	All Ages	_all	All causes	82	

	risk_name	metric	mean	\
0	Unsafe water, sanitation, and handwashing	2016 US Dollar	52.810750	
1	Unsafe water, sanitation, and handwashing	2016 US Dollar	48.000244	
2	Unsafe water, sanitation, and handwashing	2016 US Dollar	69.765933	
3	Unsafe water, sanitation, and handwashing	2016 US Dollar	88.960404	
4	Unsafe water, sanitation, and handwashing	2016 US Dollar	259.537331	

	lower	upper
0	34.776353	76.466271
1	31.644490	70.893323
2	45.764417	101.797946
3	59.168652	128.091201
4	171.226033	375.015989

```
In [6]: # 4. Tworzenie ramki danych z listy list
dane_lista = [
    ["Nowy Jork", 8419600, "USA"],
    ["Toronto", 2930000, "Kanada"],
    ["Meksyk", 9209944, "Meksyk"]
]

ramka_danych_lista = pd.DataFrame(dane_lista, columns=["Miasto", "Ludność", "Kraj"])

print("\nRamka danych utworzona z listy list:")
print(ramka_danych_lista)
```

Ramka danych utworzona z listy list:

	Miasto	Ludność	Kraj
0	Nowy Jork	8419600	USA
1	Toronto	2930000	Kanada
2	Meksyk	9209944	Meksyk

```
In [7]: # 5. Transponowanie (zamiana kolumn z wierszami)
ramka_danych_lista_transponowana = ramka_danych_lista.T

print("\nRamka danych po transponowaniu:")
print(ramka_danych_lista_transponowana)
```

Ramka danych po transponowaniu:

	0	1	2
Miasto	Nowy Jork	Toronto	Meksyk
Ludność	8419600	2930000	9209944
Kraj	USA	Kanada	Meksyk

```
In [8]: # 6. Wyświetlenie pierwszych 10 wierszy ramki danych
print("\nPierwsze 10 wierszy ramki danych:")
print(ramka_danych_csv.head(10))
```

Pierwsze 10 wierszy ramki danych:

	location_id	location_name	year	sex_id	sex	age_group_id	\
0	102	United States of America	2016	1	Male	158	
1	102	United States of America	2016	1	Male	170	
2	102	United States of America	2016	1	Male	171	
3	102	United States of America	2016	1	Male	154	
4	102	United States of America	2016	1	Male	22	
5	102	United States of America	2016	2	Female	158	
6	102	United States of America	2016	2	Female	170	
7	102	United States of America	2016	2	Female	171	
8	102	United States of America	2016	2	Female	154	
9	102	United States of America	2016	2	Female	22	

	age_group_name	acause	cause_name	risk_id	\
0	<20 years	_all	All causes	82	
1	20 to 44	_all	All causes	82	
2	45 to 64	_all	All causes	82	
3	65 plus	_all	All causes	82	
4	All Ages	_all	All causes	82	
5	<20 years	_all	All causes	82	
6	20 to 44	_all	All causes	82	
7	45 to 64	_all	All causes	82	
8	65 plus	_all	All causes	82	
9	All Ages	_all	All causes	82	

	risk_name	metric	mean	\
0	Unsafe water, sanitation, and handwashing	2016 US Dollar	52.810750	
1	Unsafe water, sanitation, and handwashing	2016 US Dollar	48.000244	
2	Unsafe water, sanitation, and handwashing	2016 US Dollar	69.765933	
3	Unsafe water, sanitation, and handwashing	2016 US Dollar	88.960404	
4	Unsafe water, sanitation, and handwashing	2016 US Dollar	259.537331	
5	Unsafe water, sanitation, and handwashing	2016 US Dollar	50.183598	
6	Unsafe water, sanitation, and handwashing	2016 US Dollar	68.050031	
7	Unsafe water, sanitation, and handwashing	2016 US Dollar	101.383728	
8	Unsafe water, sanitation, and handwashing	2016 US Dollar	143.750803	
9	Unsafe water, sanitation, and handwashing	2016 US Dollar	363.368161	

	lower	upper
0	34.776353	76.466271
1	31.644490	70.893323
2	45.764417	101.797946
3	59.168652	128.091201
4	171.226033	375.015989
5	33.315353	74.363904
6	44.128452	100.984476
7	66.731697	150.924915
8	94.584101	213.790479
9	240.982485	533.239066

```
In [9]: # 7. Wyświetlenie ostatnich 10 wierszy ramki danych
print("\nOstatnie 10 wierszy ramki danych:")
print(ramka_danych_csv.tail(10))
```

Ostatnie 10 wierszy ramki danych:

	location_id	location_name	year	sex_id	sex	\
1231	102	United States of America	2016	2	Female	
1232	102	United States of America	2016	2	Female	
1233	102	United States of America	2016	2	Female	
1234	102	United States of America	2016	2	Female	
1235	102	United States of America	2016	2	Female	
1236	102	United States of America	2016	3	Both	
1237	102	United States of America	2016	3	Both	
1238	102	United States of America	2016	3	Both	
1239	102	United States of America	2016	3	Both	
1240	102	United States of America	2016	3	Both	
	age_group_id	age_group_name	cause		cause_name	\
1231	158	<20 years	rf		Expenditure on risk factors	
1232	170	20 to 44	rf		Expenditure on risk factors	
1233	171	45 to 64	rf		Expenditure on risk factors	
1234	154	65 plus	rf		Expenditure on risk factors	
1235	22	All Ages	rf		Expenditure on risk factors	
1236	158	<20 years	rf		Expenditure on risk factors	
1237	170	20 to 44	rf		Expenditure on risk factors	
1238	171	45 to 64	rf		Expenditure on risk factors	
1239	154	65 plus	rf		Expenditure on risk factors	
1240	22	All Ages	rf		Expenditure on risk factors	
	risk_id	risk_name	metric	mean	lower	\
1231	108	High body-mass index	2016 US Dollar	101.248712	80.521641	
1232	108	High body-mass index	2016 US Dollar	2867.831645	2015.340713	
1233	108	High body-mass index	2016 US Dollar	3036.836793	2104.770152	
1234	108	High body-mass index	2016 US Dollar	1025.951678	697.331792	
1235	108	High body-mass index	2016 US Dollar	7031.868829	5216.248151	
1236	108	High body-mass index	2016 US Dollar	186.043387	142.143526	
1237	108	High body-mass index	2016 US Dollar	3697.662630	2626.047083	
1238	108	High body-mass index	2016 US Dollar	4403.916422	3083.734761	
1239	108	High body-mass index	2016 US Dollar	1457.018966	1104.520225	
1240	108	High body-mass index	2016 US Dollar	9744.641404	7222.418431	
		upper				
1231		126.900237				
1232		3745.282634				
1233		3961.429497				
1234		1909.052706				
1235		8948.681094				
1236		232.865620				
1237		4824.208117				
1238		5662.947080				
1239		2292.252672				
1240		12418.772000				

```
In [10]: # 8. Wyświetlenie informacji o ramce danych
print("\nInformacje o ramce danych:")
print(ramka_danych_csv.info())
```

Informacje o ramce danych:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1241 entries, 0 to 1240
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   location_id      1241 non-null   int64  
 1   location_name    1241 non-null   object  
 2   year              1241 non-null   int64  
 3   sex_id            1241 non-null   int64  
 4   sex                1241 non-null   object  
 5   age_group_id     1241 non-null   int64  
 6   age_group_name   1241 non-null   object  
 7   acause             1241 non-null   object  
 8   cause_name        1241 non-null   object  
 9   risk_id            1241 non-null   int64  
 10  risk_name         1241 non-null   object  
 11  metric              1241 non-null   object  
 12  mean               1241 non-null   float64 
 13  lower              1241 non-null   float64 
 14  upper              1241 non-null   float64 
dtypes: float64(3), int64(5), object(7)
memory usage: 145.6+ KB
None
```

```
In [12]: # 9. Wyświetlenie liczby wierszy i kolumn
print("\nLiczba wierszy i kolumn w ramce danych:")
print(ramka_danych_csv.shape) # Zwraca (liczba_wierszy, liczba_kolumn)
```

Liczba wierszy i kolumn w ramce danych:  
(1241, 15)

```
In [13]: # 10. Wyświetlenie informacji statystycznej o kolumnach liczbowych
print("\nStatystyki dla kolumn liczbowych")
print(ramka_danych_csv.describe())
```

Statystyki dla kolumn liczbowych:

	location_id	year	sex_id	age_group_id	risk_id	\
count	1241.0	1241.0	1241.000000	1241.000000	1241.000000	
mean	102.0	2016.0	2.020145	132.887188	129.024174	
std	0.0	0.0	0.812452	59.017044	73.481341	
min	102.0	2016.0	1.000000	22.000000	82.000000	
25%	102.0	2016.0	1.000000	154.000000	98.000000	
50%	102.0	2016.0	2.000000	158.000000	105.000000	
75%	102.0	2016.0	3.000000	170.000000	125.000000	
max	102.0	2016.0	3.000000	171.000000	381.000000	

	mean	lower	upper
count	1241.000000	1241.000000	1241.000000
mean	7523.642696	5972.186569	9227.098980
std	18443.451439	15681.094146	21316.739752
min	0.058442	-4401.524592	0.105215
25%	275.378150	129.072794	421.882542
50%	1372.476758	821.243571	2097.737090
75%	5763.893904	4033.397743	7515.218944
max	238503.405500	178217.189400	291573.753200

```
In [14]: # 11. Wyświetlenie informacji statystycznej o kolumnach kategorycznych
print("\nStatystyki dla kolumn kategorycznych")
print(ramka_danych_csv.describe(include=['object']))
```

Statystyki dla kolumn kategorycznych:

	location_name	sex	age_group_name	acause	cause_name	\
count	1241	1241		1241	1241	1241
unique		1	3		5	13
top	United States of America	Female		20 to 44	_all	All causes
freq		1241	422		271	268

	risk_name	metric
count	1241	1241
unique	19	1
top	Alcohol use	2016 US Dollar
freq	150	1241

In [15]: # 12. Usunięcie brakujących wartości w ramce danych

```
ramka_danych_csv = ramka_danych_csv.dropna()
```

```
print("\nBrakujące wartości zostały usunięte. Aktualny kształt ramki danych:")
print(ramka_danych_csv.shape) # Sprawdzenie liczby wierszy i kolumn po usunięciu
```

Brakujące wartości zostały usunięte. Aktualny kształt ramki danych:  
(1241, 15)

In [16]: # 13. Wybór wierszy i kolumn po nazwach oraz indeksach

```
# Sprawdzenie dostępnych kolumn
print("\nDostępne kolumny w ramce danych:")
print(ramka_danych_csv.columns)

# Wybór kolumn po nazwach (przykładowe kolumny)
kolumny_do_wyboru = ["location_name", "mean"]
kolumny_do_wyboru = [col for col in kolumny_do_wyboru if col in ramka_danych_csv]

print("\nWybór kolumn po nazwach:")
print(ramka_danych_csv.loc[0:5, kolumny_do_wyboru]) # Pierwsze 6 wierszy

# Wybór kolumn po indeksach
print("\nWybór kolumn po indeksach:")
print(ramka_danych_csv.iloc[0:5, 1:3]) #
```

Dostępne kolumny w ramce danych:

```
Index(['location_id', 'location_name', 'year', 'sex_id', 'sex', 'age_group_id',
       'age_group_name', 'acause', 'cause_name', 'risk_id', 'risk_name',
       'metric', 'mean', 'lower', 'upper'],
      dtype='object')
```

Wybór kolumn po nazwach:

	location_name	mean
0	United States of America	52.810750
1	United States of America	48.000244
2	United States of America	69.765933
3	United States of America	88.960404
4	United States of America	259.537331
5	United States of America	50.183598

Wybór kolumn po indeksach:

	location_name	year
0	United States of America	2016
1	United States of America	2016
2	United States of America	2016
3	United States of America	2016
4	United States of America	2016

In [17]: # 14. Wybór wierszy na podstawie określonej wartości kolumny

```
# Sprawdzenie, czy kolumna "risk_name" istnieje w danych
kolumna_filtrowana = "risk_name"
wartosc_filtrowana = "Alcohol use" # Możesz zmienić na inną wartość występującą

if kolumna_filtrowana in ramka_danych_csv.columns:
    print(f"\nWiersze, gdzie {kolumna_filtrowana} to '{wartosc_filtrowana}':")
    print(ramka_danych_csv[ramka_danych_csv[kolumna_filtrowana] == wartosc_filtrowana])
else:
    print(f"\nBłąd: Kolumna '{kolumna_filtrowana}' nie istnieje w ramce danych.")
```

Wiersze, gdzie risk\_name to 'Alcohol use':

	location_id	location_name	year	sex_id	sex	age_group_id	\
75	102	United States of America	2016	1	Male	158	
76	102	United States of America	2016	1	Male	170	
77	102	United States of America	2016	1	Male	171	
78	102	United States of America	2016	1	Male	154	
79	102	United States of America	2016	1	Male	22	
...	...	...	...	...	...	...	
1089	102	United States of America	2016	3	Both	158	
1090	102	United States of America	2016	3	Both	170	
1091	102	United States of America	2016	3	Both	171	
1092	102	United States of America	2016	3	Both	154	
1093	102	United States of America	2016	3	Both	22	

	age_group_name	acause	cause_name	risk_id	risk_name	\
75	<20 years	_all	All causes	102	Alcohol use	
76	20 to 44	_all	All causes	102	Alcohol use	
77	45 to 64	_all	All causes	102	Alcohol use	
78	65 plus	_all	All causes	102	Alcohol use	
79	All Ages	_all	All causes	102	Alcohol use	
...	...	...	...	...	...	
1089	<20 years	digest	Digestive diseases	102	Alcohol use	
1090	20 to 44	digest	Digestive diseases	102	Alcohol use	
1091	45 to 64	digest	Digestive diseases	102	Alcohol use	
1092	65 plus	digest	Digestive diseases	102	Alcohol use	
1093	All Ages	digest	Digestive diseases	102	Alcohol use	

	metric	mean	lower	upper
75	2016 US Dollar	809.896112	564.217748	1136.318912
76	2016 US Dollar	9805.336756	7776.821020	12054.894010
77	2016 US Dollar	12964.862750	8391.647387	18270.423670
78	2016 US Dollar	5421.192741	1730.902323	9576.866836
79	2016 US Dollar	29001.288360	19289.247540	39797.832630
...	...	...	...	...
1089	2016 US Dollar	22.821660	10.088865	41.226727
1090	2016 US Dollar	905.703730	522.197991	1308.987930
1091	2016 US Dollar	916.635307	392.092905	1478.297508
1092	2016 US Dollar	277.444986	31.625331	540.776131
1093	2016 US Dollar	2122.605684	987.873947	3282.622546

[150 rows x 15 columns]

In [19]: # 15. Wybór wierszy spełniających kilka warunków jednocześnie

```
# Sprawdzamy, czy kolumny istnieją
kolumna1 = "risk_name"
wartosc1 = "Alcohol use"
kolumna2 = "year"
wartosc2 = 2016

if kolumna1 in ramka_danych_csv.columns and kolumna2 in ramka_danych_csv.columns:
    print(f"\nWiersze, gdzie {kolumna1} to '{wartosc1}' i {kolumna2} to {wartosc2}")
    print(ramka_danych_csv[(ramka_danych_csv[kolumna1] == wartosc1) & (ramka_danych_csv[kolumna2] == wartosc2)])
else:
    print(f"\nBłąd: Jedna z kolumn '{kolumna1}' lub '{kolumna2}' nie istnieje w ramce danych")
```

Wiersze, gdzie risk\_name to 'Alcohol use' i year to 2016:

	location_id	location_name	year	sex_id	sex	age_group_id	\
75	102	United States of America	2016	1	Male	158	
76	102	United States of America	2016	1	Male	170	
77	102	United States of America	2016	1	Male	171	
78	102	United States of America	2016	1	Male	154	
79	102	United States of America	2016	1	Male	22	
...	...	...	...	...	...	...	
1089	102	United States of America	2016	3	Both	158	
1090	102	United States of America	2016	3	Both	170	
1091	102	United States of America	2016	3	Both	171	
1092	102	United States of America	2016	3	Both	154	
1093	102	United States of America	2016	3	Both	22	
	age_group_name	acause	cause_name	risk_id	risk_name	\	
75	<20 years	_all	All causes	102	Alcohol use		
76	20 to 44	_all	All causes	102	Alcohol use		
77	45 to 64	_all	All causes	102	Alcohol use		
78	65 plus	_all	All causes	102	Alcohol use		
79	All Ages	_all	All causes	102	Alcohol use		
...	...	...	...	...	...	...	
1089	<20 years	digest	Digestive diseases	102	Alcohol use		
1090	20 to 44	digest	Digestive diseases	102	Alcohol use		
1091	45 to 64	digest	Digestive diseases	102	Alcohol use		
1092	65 plus	digest	Digestive diseases	102	Alcohol use		
1093	All Ages	digest	Digestive diseases	102	Alcohol use		
	metric	mean	lower	upper			
75	2016 US Dollar	809.896112	564.217748	1136.318912			
76	2016 US Dollar	9805.336756	7776.821020	12054.894010			
77	2016 US Dollar	12964.862750	8391.647387	18270.423670			
78	2016 US Dollar	5421.192741	1730.902323	9576.866836			
79	2016 US Dollar	29001.288360	19289.247540	39797.832630			
...	...	...	...	...	...	...	
1089	2016 US Dollar	22.821660	10.088865	41.226727			
1090	2016 US Dollar	905.703730	522.197991	1308.987930			
1091	2016 US Dollar	916.635307	392.092905	1478.297508			
1092	2016 US Dollar	277.444986	31.625331	540.776131			
1093	2016 US Dollar	2122.605684	987.873947	3282.622546			

[150 rows x 15 columns]

In [20]: # 16. Wybór wierszy, gdzie kolumna kategoryczna zawiera określone słowo

```
# Kolumna do filtrowania i słowo kluczowe
kolumna_filtrowana = "risk_name"
słowo_kluczowe = "Tobacco" # Możesz zmienić na inne słowo

# Sprawdzenie, czy kolumna istnieje
if kolumna_filtrowana in ramka_danych_csv.columns:
    print(f"\nWiersze, gdzie kolumna '{kolumna_filtrowana}' zawiera słowo '{słowo_kluczowe}'")
    print(ramka_danych_csv[ramka_danych_csv[kolumna_filtrowana].str.contains(słowo_kluczowe)])
else:
    print(f"\nBłąd: Kolumna '{kolumna_filtrowana}' nie istnieje w ramce danych.")
```

Wiersze, gdzie kolumna 'risk\_name' zawiera słowo 'Tobacco':

	location_id	location_name	year	sex_id	sex	\
60	102	United States of America	2016	1	Male	
61	102	United States of America	2016	1	Male	
62	102	United States of America	2016	1	Male	
63	102	United States of America	2016	1	Male	
64	102	United States of America	2016	1	Male	
...	...	...	...	...	...	
1176	102	United States of America	2016	2	Female	
1177	102	United States of America	2016	3	Both	
1178	102	United States of America	2016	3	Both	
1179	102	United States of America	2016	3	Both	
1180	102	United States of America	2016	3	Both	
	age_group_id	age_group_name	acause		cause_name	\
60	158	<20 years	_all		All causes	
61	170	20 to 44	_all		All causes	
62	171	45 to 64	_all		All causes	
63	154	65 plus	_all		All causes	
64	22	All Ages	_all		All causes	
...	...	...	...		...	
1176	22	All Ages	resp	Chronic respiratory diseases		
1177	170	20 to 44	resp	Chronic respiratory diseases		
1178	171	45 to 64	resp	Chronic respiratory diseases		
1179	154	65 plus	resp	Chronic respiratory diseases		
1180	22	All Ages	resp	Chronic respiratory diseases		
	risk_id	risk_name	metric	mean	lower	\
60	98	Tobacco	2016 US Dollar	517.466504	343.789930	
61	98	Tobacco	2016 US Dollar	5858.348322	4965.560873	
62	98	Tobacco	2016 US Dollar	33276.545480	29740.534840	
63	98	Tobacco	2016 US Dollar	29654.174070	26291.391010	
64	98	Tobacco	2016 US Dollar	69306.534370	62651.505670	
...	...	...	...	...	...	
1176	98	Tobacco	2016 US Dollar	8222.271063	6747.414164	
1177	98	Tobacco	2016 US Dollar	692.063133	407.813810	
1178	98	Tobacco	2016 US Dollar	5600.259499	4531.227772	
1179	98	Tobacco	2016 US Dollar	8758.646080	7274.289789	
1180	98	Tobacco	2016 US Dollar	15050.968710	12542.772790	
		upper				
60		712.649285				
61		6851.740572				
62		36737.606660				
63		33371.325550				
64		76505.313690				
...		...				
1176		9796.790450				
1177		951.909694				
1178		6718.759946				
1179		10505.380740				
1180		17889.599790				

[126 rows x 15 columns]

In [21]: # 17. Wybór wierszy, gdzie kolumna kategoryczna NIE zawiera określonego słowa

```
# Kolumna do filtrowania i słowo kluczowe
kolumna_filtrowana = "risk_name"
słowo_kluczowe = "Tobacco" # Możesz zmienić na inne słowo
```

```
# Sprawdzenie, czy kolumna istnieje
if kolumna_filtrowana in ramka_danych_csv.columns:
    print(f"\nWiersze, gdzie kolumna '{kolumna_filtrowana}' NIE zawiera słowa '{")
    print(ramka_danych_csv[~ramka_danych_csv[kolumna_filtrowana].str.contains(sl
else:
    print(f"\nBłąd: Kolumna '{kolumna_filtrowana}' nie istnieje w ramce danych."
```

Wiersze, gdzie kolumna 'risk\_name' NIE zawiera słowa 'Tobacco':

	location_id	location_name	year	sex_id	sex	age_group_id	\
0	102	United States of America	2016	1	Male	158	
1	102	United States of America	2016	1	Male	170	
2	102	United States of America	2016	1	Male	171	
3	102	United States of America	2016	1	Male	154	
4	102	United States of America	2016	1	Male	22	
...	...	...	...	...	...	...	\
1236	102	United States of America	2016	3	Both	158	
1237	102	United States of America	2016	3	Both	170	
1238	102	United States of America	2016	3	Both	171	
1239	102	United States of America	2016	3	Both	154	
1240	102	United States of America	2016	3	Both	22	

	age_group_name	acause	cause_name	risk_id	\
0	<20 years	_all	All causes	82	
1	20 to 44	_all	All causes	82	
2	45 to 64	_all	All causes	82	
3	65 plus	_all	All causes	82	
4	All Ages	_all	All causes	82	
...	...	...	...	...	\
1236	<20 years	rf	Expenditure on risk factors	108	
1237	20 to 44	rf	Expenditure on risk factors	108	
1238	45 to 64	rf	Expenditure on risk factors	108	
1239	65 plus	rf	Expenditure on risk factors	108	
1240	All Ages	rf	Expenditure on risk factors	108	

	risk_name	metric	mean	\
0	Unsafe water, sanitation, and handwashing	2016 US Dollar	52.810750	
1	Unsafe water, sanitation, and handwashing	2016 US Dollar	48.000244	
2	Unsafe water, sanitation, and handwashing	2016 US Dollar	69.765933	
3	Unsafe water, sanitation, and handwashing	2016 US Dollar	88.960404	
4	Unsafe water, sanitation, and handwashing	2016 US Dollar	259.537331	
...	...	...	...	\
1236	High body-mass index	2016 US Dollar	186.043387	
1237	High body-mass index	2016 US Dollar	3697.662630	
1238	High body-mass index	2016 US Dollar	4403.916422	
1239	High body-mass index	2016 US Dollar	1457.018966	
1240	High body-mass index	2016 US Dollar	9744.641404	

	lower	upper
0	34.776353	76.466271
1	31.644490	70.893323
2	45.764417	101.797946
3	59.168652	128.091201
4	171.226033	375.015989
...	...	...
1236	142.143526	232.865620
1237	2626.047083	4824.208117
1238	3083.734761	5662.947080
1239	1104.520225	2292.252672
1240	7222.418431	12418.772000

[1115 rows x 15 columns]

In [27]: # 18. Tworzenie nowej kolumny na podstawie istniejących

```
# Sprawdzenie dostępnych kolumn w pliku
print("\nDostępne kolumny w ramce danych:")
print(ramka_danych_csv.columns)
```

```
# Wybór odpowiednich kolumn do obliczeń
kolumna_istniejaca1 = "spending_mean" # Przykładowa kolumna z wydatkami
kolumna_istniejaca2 = "population" # Przykładowa kolumna z populacją
nowa_kolumna = "spending_per_capita" # Nazwa nowej kolumny

# Sprawdzenie, czy wymagane kolumny istnieją w danych
if kolumna_istniejaca1 in ramka_danych_csv.columns and kolumna_istniejaca2 in ramka_danych_csv.columns:
    # Tworzenie nowej kolumny jako iloraz dwóch istniejących wartości
    ramka_danych_csv[nowa_kolumna] = ramka_danych_csv[kolumna_istniejaca1] / ramka_danych_csv[kolumna_istniejaca2]

print(f"\nNowa kolumna '{nowa_kolumna}' została utworzona na podstawie '{kolumna_istniejaca1} i {kolumna_istniejaca2}'")
print(ramka_danych_csv[[kolumna_istniejaca1, kolumna_istniejaca2, nowa_kolumna]])
```

Dostępne kolumny w ramce danych:

```
Index(['location_id', 'location_name', 'year', 'sex_id', 'sex', 'age_group_id',
       'age_group_name', 'acause', 'cause_name', 'risk_id', 'risk_name',
       'metric', 'mean', 'lower', 'upper'],
      dtype='object')
```

In [29]: # 21. Usunięcie kolumny z ramki danych

```
kolumna_do_usuniecia = "spending_per_capita" # Podaj nazwę kolumny do usunięcia

# Sprawdzenie, czy kolumna istnieje w danych
if kolumna_do_usuniecia in ramka_danych_csv.columns:
    ramka_danych_csv = ramka_danych_csv.drop(columns=[kolumna_do_usuniecia])
    print(f"\nKolumna '{kolumna_do_usuniecia}' została usunięta.")

# Wyświetlenie aktualnych kolumn po usunięciu
print("\nAktualne kolumny w ramce danych:")
print(ramka_danych_csv.columns)
```

Aktualne kolumny w ramce danych:

```
Index(['location_id', 'location_name', 'year', 'sex_id', 'sex', 'age_group_id',
       'age_group_name', 'acause', 'cause_name', 'risk_id', 'risk_name',
       'metric', 'mean', 'lower', 'upper'],
      dtype='object')
```

In [30]: # 22. Zmiana nazwy kolumny w ramce danych

```
stara_nazwa = "spending_mean" # Podaj aktualną nazwę kolumny
nowa_nazwa = "average_spending" # Podaj nową nazwę kolumny

# Sprawdzenie, czy kolumna istnieje
if stara_nazwa in ramka_danych_csv.columns:
    ramka_danych_csv = ramka_danych_csv.rename(columns={stara_nazwa: nowa_nazwa})
    print(f"\nZmieniono nazwę kolumny '{stara_nazwa}' na '{nowa_nazwa}' .")
else:
    print(f"\nBłąd: Kolumna '{stara_nazwa}' nie istnieje w ramce danych.")

# Wyświetlenie aktualnych nazw kolumn
print("\nAktualne kolumny w ramce danych:")
print(ramka_danych_csv.columns)
```

Błąd: Kolumna 'spending\_mean' nie istnieje w ramce danych.

Aktualne kolumny w ramce danych:

```
Index(['location_id', 'location_name', 'year', 'sex_id', 'sex', 'age_group_id',
       'age_group_name', 'acause', 'cause_name', 'risk_id', 'risk_name',
       'metric', 'mean', 'lower', 'upper'],
      dtype='object')
```

In [32]: # 20. Zmień nazwę kolumny  
`ramka_danych_csv = ramka_danych_csv.rename(columns={'location_name': 'location_c')}`

In [34]: `print(ramka_danych_csv.columns)`

```
Index(['location_id', 'location_country', 'year', 'sex_id', 'sex',
       'age_group_id', 'age_group_name', 'acause', 'cause_name', 'risk_id',
       'risk_name', 'metric', 'mean', 'lower', 'upper'],
      dtype='object')
```

In [35]: # 21. Zachowaj ramkę danych jako plik csv na komputerze  
`ramka_danych_csv.to_csv('updated_data.csv', index=False)`

In [36]: # 22. Wyświetlić liczbę wierszy  
`print(len(ramka_danych_csv))`

1241

In [37]: # 23. Wyświetlić wartości unikatowe w kolumnie  
`print(ramka_danych_csv['location_country'].unique())`

['United States of America']

In [38]: # 24. Wyświetlić liczby rekordów odpowiadających do wartości  
`print(ramka_danych_csv['location_country'].value_counts())`

```
location_country
United States of America    1241
Name: count, dtype: int64
```

In [40]: # 25. Sortowanie wierszy ramki danych według wartości określonej kolumny (malejąco)  
`ramka_danych_csv = ramka_danych_csv.sort_values(by='location_country', ascending=False)`

In [41]: # 26. Wyświetlić wierszy dla 10 największych (najmniejszych) wartości określonej kolumny  
`print(ramka_danych_csv.nlargest(10, 'upper'))`  
`print(ramka_danych_csv.nsmallest(10, 'upper'))`

	location_id	location_country	year	sex_id	sex	\
147	102	United States of America	2016	3	Both	
238	102	United States of America	2016	3	Both	
251	102	United States of America	2016	3	Both	
142	102	United States of America	2016	2	Female	
171	102	United States of America	2016	3	Both	
74	102	United States of America	2016	3	Both	
146	102	United States of America	2016	3	Both	
137	102	United States of America	2016	1	Male	
145	102	United States of America	2016	3	Both	
460	102	United States of America	2016	3	Both	
	age_group_id	age_group_name	acause			\
147	22	All Ages	_all			
238	22	All Ages	_all			
251	22	All Ages	_all			
142	22	All Ages	_all			
171	22	All Ages	_all			
74	22	All Ages	_all			
146	154	65 plus	_all			
137	22	All Ages	_all			
145	171	45 to 64	_all			
460	22	All Ages	_dube			
		cause_name	risk_id			\
147		All causes	108			
238		All causes	341			
251		All causes	367			
142		All causes	108			
171		All causes	110			
74		All causes	98			
146		All causes	108			
137		All causes	108			
145		All causes	108			
460	Diabetes and urogenital, blood, and endocrine ...		105			
	risk_name	metric	mean	lower	upper	\
147	High body-mass index	2016 US Dollar	238503.4055	178217.18940		
238	High systolic blood pressure	2016 US Dollar	179903.7277	164459.06720		
251	High fasting plasma glucose	2016 US Dollar	171932.5283	154821.07370		
142	High body-mass index	2016 US Dollar	133082.2466	100602.17950		
171	Dietary risks	2016 US Dollar	143551.7276	130343.09990		
74	Tobacco	2016 US Dollar	129959.6416	116812.01360		
146	High body-mass index	2016 US Dollar	101327.8710	69506.57512		
137	High body-mass index	2016 US Dollar	105421.1590	76396.68290		
145	High body-mass index	2016 US Dollar	109030.3703	85125.38710		
460	High fasting plasma glucose	2016 US Dollar	120366.3232	113086.49690		
	upper					\
147	291573.7532					
238	195962.0712					
251	191940.1801					
142	162581.4562					
171	156080.6003					
74	143525.1170					
146	132702.6808					
137	131279.1270					
145	128130.7642					
460	126741.3779					
	location_id	location_country	year	sex_id	sex	\

1033	102	United States of America	2016	2	Female
1149	102	United States of America	2016	2	Female
874	102	United States of America	2016	2	Female
1028	102	United States of America	2016	1	Male
1038	102	United States of America	2016	3	Both
730	102	United States of America	2016	1	Male
735	102	United States of America	2016	2	Female
688	102	United States of America	2016	1	Male
693	102	United States of America	2016	2	Female
849	102	United States of America	2016	2	Female

	age_group_id	age_group_name	acause	cause_name	\
1033	158	<20 years	cvd	Cardiovascular diseases	
1149	170	20 to 44	msk	Musculoskeletal disorders	
874	158	<20 years	cirrhosis	Cirrhosis	
1028	158	<20 years	cvd	Cardiovascular diseases	
1038	158	<20 years	cvd	Cardiovascular diseases	
730	158	<20 years	_neo	Neoplasms	
735	158	<20 years	_neo	Neoplasms	
688	158	<20 years	_neo	Neoplasms	
693	158	<20 years	_neo	Neoplasms	
849	170	20 to 44	_neuro	Neurological disorders	

	risk_id	risk_name	metric	mean	\
1033	126	Occupational risks	2016 US Dollar	0.058442	
1149	341	Impaired kidney function	2016 US Dollar	0.304079	
874	102	Alcohol use	2016 US Dollar	0.376233	
1028	126	Occupational risks	2016 US Dollar	0.375501	
1038	126	Occupational risks	2016 US Dollar	0.433943	
730	103	Drug use	2016 US Dollar	0.717406	
735	103	Drug use	2016 US Dollar	0.950784	
688	89	Other environmental risks	2016 US Dollar	0.642990	
693	89	Other environmental risks	2016 US Dollar	0.634566	
849	105	High fasting plasma glucose	2016 US Dollar	0.662740	

	lower	upper
1033	0.018899	0.105215
1149	0.217169	0.433294
874	0.217804	0.589010
1028	0.138020	0.642662
1038	0.187088	0.706757
730	0.501235	0.984388
735	0.689007	1.298707
688	0.107717	1.450370
693	0.105842	1.507581
849	0.100641	1.695272

```
In [42]: # 27. Wyświetlić wierszy dla 10 największych wartości określonej kolumny pod warunkiem, że kolumna location_country ma wartość 'Global'
filtered_data = ramka_danych_csv[ramka_danych_csv['location_country'] == 'Global']
print(filtered_data.nlargest(10, 'lower'))
```

Empty DataFrame

Columns: [location\_id, location\_country, year, sex\_id, sex, age\_group\_id, age\_group\_name, acause, cause\_name, risk\_id, risk\_name, metric, mean, lower, upper]  
Index: []

```
In [44]: # 28. Grupowanie wierszy według wartości kolumny skategoryzowanej, potem uśrednienie
grouped_mean = ramka_danych_csv.groupby(['location_country', 'year']).mean()
print(grouped_mean)
```

```

location_id    sex_id   age_group_id \
location_country      year
United States of America 2016      102.0  2.020145  132.887188

risk_id          mean        lower \
location_country      year
United States of America 2016  129.024174  7523.642696  5972.186569

upper
location_country      year
United States of America 2016  9227.09898

```

In [45]: # 29. Grupowanie wierszy według wartości kolumny skategoryzowanej, potem uśredni  
grouped\_stats = ramka\_danych\_csv.groupby('upper').agg({  
 'upper': ['mean', 'count'],  
 'upper': 'median',  
 'year': 'max'  
})  
print(grouped\_stats)

	upper	year
upper		
0.105215	0.105215	2016
0.433294	0.433294	2016
0.589010	0.589010	2016
0.642662	0.642662	2016
0.706757	0.706757	2016
...	...	...
156080.600300	156080.600300	2016
162581.456200	162581.456200	2016
191940.180100	191940.180100	2016
195962.071200	195962.071200	2016
291573.753200	291573.753200	2016

[1145 rows x 2 columns]

In [46]: # 30. Wyświetlić nazwy kolumn indeksu złożonego  
print(grouped\_mean.index.names)

['location\_country', 'year']

In [47]: # 31. Posortować kolumnę, indeksu złożonego  
sorted\_index = grouped\_mean.sort\_index()  
print(sorted\_index)

```

location_id    sex_id   age_group_id \
location_country      year
United States of America 2016      102.0  2.020145  132.887188

risk_id          mean        lower \
location_country      year
United States of America 2016  129.024174  7523.642696  5972.186569

upper
location_country      year
United States of America 2016  9227.09898

```

In [48]: # 32. Stworzyć tabelę przedstawiącą (pivot table) na podstawie ramki danych  
pivot\_table = pd.pivot\_table(ramka\_danych\_csv, values='upper', index=['location\_'
print(pivot\_table)

```
year                      2016
location_country
United States of America  9227.09898
```

```
In [49]: # 33. Wyświetlić indeksy i kolumny tabeli przestawnej
print(pivot_table.index)
print(pivot_table.columns)

Index(['United States of America'], dtype='object', name='location_country')
Index([2016], dtype='int64', name='year')

In [50]: # 34. Utwórz indeks złożony tabeli przestawnej i wyświetl go
pivot_table = pivot_table.reset_index()
pivot_table.set_index(['location_country'] + list(pivot_table.columns[1:]), inplace=True)
print(pivot_table.index)

MultiIndex([('United States of America', 9227.098980246341)],
           names=['location_country', 2016])

In [2]: 35# Importujemy potrzebne biblioteki
import pandas as pd
import matplotlib.pyplot as plt

36-37# Załaduj plik CSV
file_path = 'IHME_USA_RISK_SPENDING_2016_Y2020M09D29.csv'
df = pd.read_csv(file_path)

# Ustawienie, by wykresy były widoczne bezpośrednio w notebooku
%matplotlib inline

# Przekształcamy dane do formatu "długiego" z użyciem melt
df_melted = df[['year', 'risk_name', 'age_group_name', 'mean']]

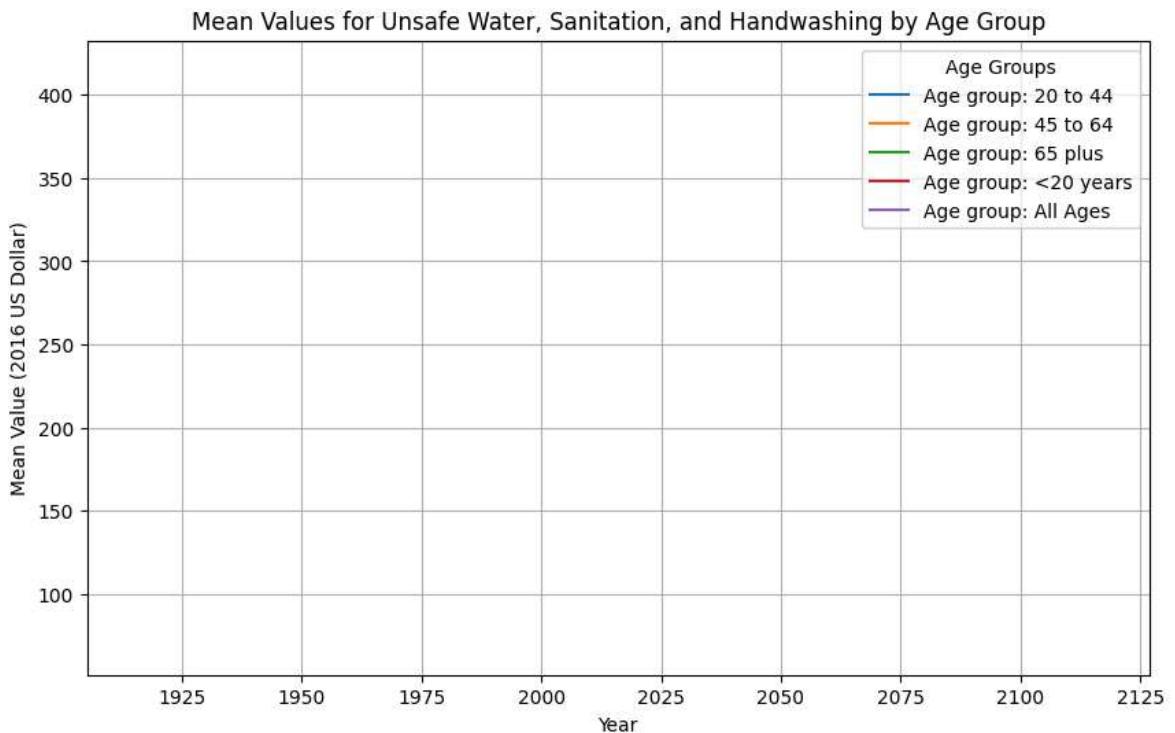
# Filtrujemy dane, np. dla konkretnego ryzyka
df_risk = df_melted[df_melted['risk_name'] == 'Unsafe water, sanitation, and handwashing']

# Grupujemy dane według roku i grupy wiekowej, obliczając średnią wartość
df_grouped = df_risk.groupby(['year', 'age_group_name'])['mean'].mean().reset_index()

# Tworzymy wykres
plt.figure(figsize=(10, 6))
for age_group in df_grouped['age_group_name'].unique():
    data = df_grouped[df_grouped['age_group_name'] == age_group]
    plt.plot(data['year'], data['mean'], label=f'Age group: {age_group}')

# Dodajemy etykiety, legendę i tytuł
plt.xlabel('Year')
plt.ylabel('Mean Value (2016 US Dollar)')
plt.title('Mean Values for Unsafe Water, Sanitation, and Handwashing by Age Group')
plt.legend(title='Age Groups')
plt.grid(True)

# Pokaż wykres
plt.show()
```



```
In [3]: # 38 Importujemy potrzebne biblioteki
import pandas as pd
import matplotlib.pyplot as plt

# Załaduj plik CSV
file_path = 'IHME_USA_RISK_SPENDING_2016_Y2020M09D29.csv'
df = pd.read_csv(file_path)

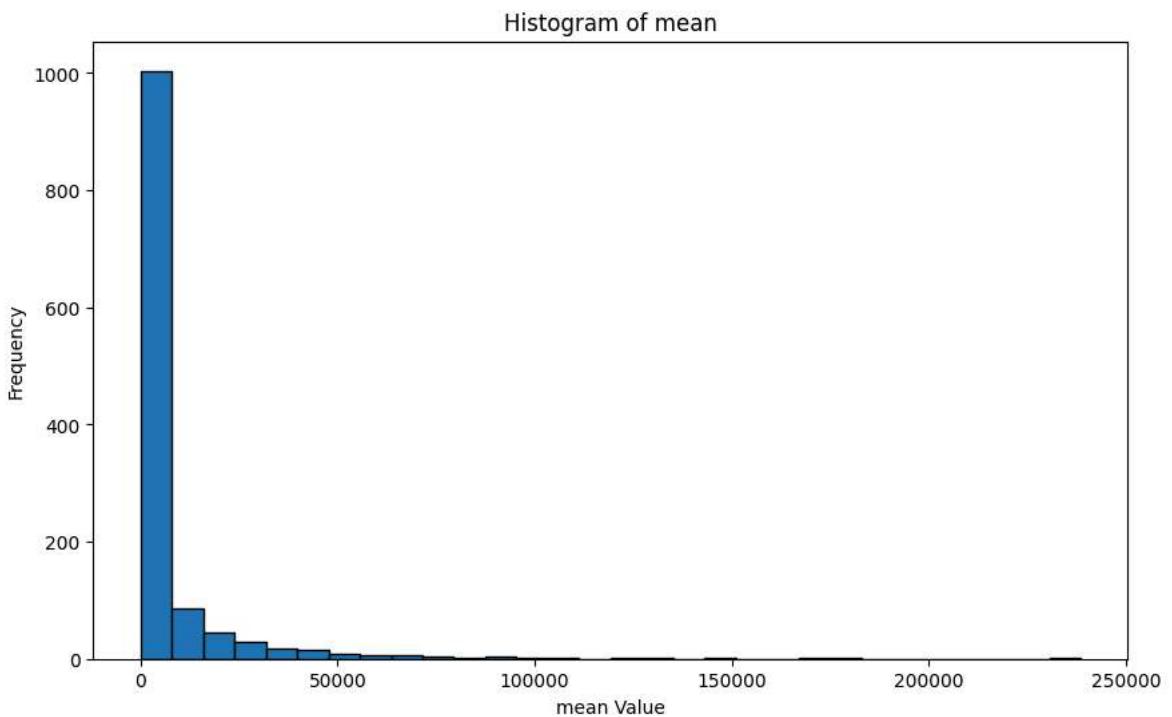
# Ustawienie, by wykresy były widoczne bezpośrednio w notebooku
%matplotlib inline

# Wybieramy kolumnę, np. 'mean', do narysowania histogramu
column_to_plot = 'mean'

# Rysowanie histogramu
plt.figure(figsize=(10, 6))
plt.hist(df[column_to_plot], bins=30, edgecolor='black')

# Dodajemy etykiety i tytuł
plt.xlabel(f'{column_to_plot} Value')
plt.ylabel('Frequency')
plt.title(f'Histogram of {column_to_plot}')

# Pokaż wykres
plt.show()
```



```
In [4]: # 39. Przedstawić sposoby łączenia ramek danych za pomocą metod merge i concat
# Metoda merge:
df1 = pd.DataFrame({'key': ['A', 'B', 'C'], 'value1': [1, 2, 3]})
df2 = pd.DataFrame({'key': ['A', 'B', 'D'], 'value2': [4, 5, 6]})
merged_df = pd.merge(df1, df2, on='key', how='inner')
print(merged_df)
# Metoda konkatenacji:
df3 = pd.DataFrame({'key': ['A', 'B'], 'value1': [7, 8]})
df4 = pd.DataFrame({'key': ['C', 'D'], 'value1': [9, 10]})
concatenated_df = pd.concat([df3, df4], ignore_index=True)
print(concatenated_df)
```

	key	value1	value2
0	A	1	4
1	B	2	5
	key	value1	
0	A	7	
1	B	8	
2	C	9	
3	D	10	

```
In [6]: # 40 Importujemy potrzebne biblioteki
import pandas as pd

# Załaduj plik CSV
file_path = 'IHME_USA_RISK_SPENDING_2016_Y2020M09D29.csv'
df = pd.read_csv(file_path)

# Ustawienie, by wykresy były widoczne bezpośrednio w notebooku
%matplotlib inline

# 1. Dodanie kolumny różnicy między 'upper' a 'lower'
df['difference_upper_lower'] = df['upper'] - df['lower']

# 2. Dodanie kolumny średniej wartości z 'mean', 'upper' i 'Lower'
df['average_value'] = (df['mean'] + df['upper'] + df['lower']) / 3

# 3. Dodanie kolumny, która będzie wynikiem mnożenia 'mean' przez 1.1 (np. 10% w
```

```
df['mean_increased_10_percent'] = df['mean'] * 1.1

# Wyświetlamy pierwsze 5 wierszy, aby sprawdzić nowe kolumny
df[['year', 'risk_name', 'mean', 'upper', 'lower', 'difference_upper_lower', 'aver']]
```

Out[6]:

	year	risk_name	mean	upper	lower	difference_upper_lower	aver
0	2016	Unsafe water, sanitation, and handwashing	52.810750	76.466271	34.776353		41.689918
1	2016	Unsafe water, sanitation, and handwashing	48.000244	70.893323	31.644490		39.248833
2	2016	Unsafe water, sanitation, and handwashing	69.765933	101.797946	45.764417		56.033530
3	2016	Unsafe water, sanitation, and handwashing	88.960404	128.091201	59.168652		68.922549
4	2016	Unsafe water, sanitation, and handwashing	259.537331	375.015989	171.226033		203.789956
							2

In [8]:

```
# 41 Importujemy bibliotekę
import pandas as pd

# Załaduj plik CSV
file_path = 'IHME_USA_RISK_SPENDING_2016_Y2020M09D29.csv'
df = pd.read_csv(file_path)

# 1. Tworzenie nowej kolumny "risk_level" na podstawie wartości 'mean'
mean_threshold = df['mean'].mean() # Obliczamy średnią wartość mean
df['risk_level'] = df['mean'].apply(lambda x: 'High' if x > mean_threshold else 'Low')

# 2. Zamiana nazw płci na skróconą formę ("Male" → "M", "Female" → "F")
df['sex_short'] = df['sex'].apply(lambda x: 'M' if x == 'Male' else 'F')

# 3. Kategoryzacja wieku (jeśli "plus" w nazwie grupy wiekowej, oznaczamy jako "Senior")
df['age_category'] = df['age_group_name'].apply(lambda x: 'Senior' if 'plus' in x else 'Young')

# Wyświetlamy kilka pierwszych wierszy z nowymi kolumnami
df[['year', 'risk_name', 'mean', 'risk_level', 'sex', 'sex_short', 'age_group_name']]
```

Out[8]:

	year	risk_name	mean	risk_level	sex	sex_short	age_group_name	age_cat
0	2016	Unsafe water, sanitation, and handwashing	52.810750	Low	Male	M	<20 years	
1	2016	Unsafe water, sanitation, and handwashing	48.000244	Low	Male	M	20 to 44	
2	2016	Unsafe water, sanitation, and handwashing	69.765933	Low	Male	M	45 to 64	
3	2016	Unsafe water, sanitation, and handwashing	88.960404	Low	Male	M	65 plus	
4	2016	Unsafe water, sanitation, and handwashing	259.537331	Low	Male	M	All Ages	



In [12]:

```
# 42. Przedstawić możliwości pracy z dużymi plikami przy użyciu argumentu chunks
chunksize = 10000
for chunk in pd.read_csv('IHME_USA_RISK_SPENDING_2016_Y2020M09D29.csv', chunksize=chunksize):
    print(chunk.head(4))
```

```
location_id          location_name  year  sex_id  sex  age_group_id \
0      102  United States of America  2016      1  Male      158
1      102  United States of America  2016      1  Male      170
2      102  United States of America  2016      1  Male      171
3      102  United States of America  2016      1  Male      154

age_group_name  acause  cause_name  risk_id \
0    <20 years    _all  All causes     82
1    20 to 44     _all  All causes     82
2    45 to 64     _all  All causes     82
3   65 plus      _all  All causes     82

risk_name      metric      mean \
0 Unsafe water, sanitation, and handwashing  2016 US Dollar  52.810750
1 Unsafe water, sanitation, and handwashing  2016 US Dollar  48.000244
2 Unsafe water, sanitation, and handwashing  2016 US Dollar  69.765933
3 Unsafe water, sanitation, and handwashing  2016 US Dollar  88.960404

lower      upper
0  34.776353  76.466271
1  31.644490  70.893323
2  45.764417 101.797946
3  59.168652 128.091201
```

In [ ]: