In [166…

```
import pandas as pd # ładowanie biblioteki Pandas
```

In [167…

```
dict_city = {"City" : ["Warszawa", "Łódź", "Poznań"],
             "Population" : [12678079,   5398064,  1625631]}

df = pd.DataFrame(dict_city) # tworzenie ramki danych ze słownika
df
```

Out[167…

	City	Population
0	Warszawa	12678079
1	Łódź	5398064
2	Poznań	1625631

In [168…

```
df.to_csv("example.csv")
```

In [169…

```
df = pd.read_csv("IHME_PREM_WMN_HEALTH_2020_Y2011M10D11.csv", encoding = "utf-8"
```

In [170…

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

Out[170…

	observation_id	submitted_time	gender	age	geography	financial_situa
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	I cannot a enough foo my fa
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I cannot a enough foo my fa
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can afford t but nothing
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can afford and re expenses, and
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can afford and re expenses, and
5	wmn_4504301990051840	2020-07-10 11:27:16.581 UTC	Female	26 to 35 years old	City center or metropolitan area	I can comfor afford t clothes, and
6	wmn_4504322055602176	2020-07-09 20:43:11.055 UTC	Female	16 to 25 years old	City center or metropolitan area	I can afford and re expenses, and
7	wmn_4504369904222208	2020-07-18 12:52:31.482 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I cannot a enough foo my fa
8	wmn_4504469091123200	2020-07-16 16:03:44.066 UTC	Female	36 to 45 years old	City center or metropolitan area	I can afford and re expenses, but
9	wmn_4504687899574272	2020-07-17 07:16:32.082 UTC	Female	36 to 45 years old	City center or metropolitan area	I can afford and re expenses, but

10 rows × 46 columns

In [171…

```
df_T = df.T  # zamienia wiersze z kolumnami
```

In [172…

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

Out[172…

	observation_id	submitted_time	gender	age	geography	financial_situa
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	I cannot a enough foo my fa
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I cannot a enough foo my fa
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can afford t but nothing
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can afford and re expenses, and
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can afford and re expenses, and
5	wmn_4504301990051840	2020-07-10 11:27:16.581 UTC	Female	26 to 35 years old	City center or metropolitan area	I can comfor afford t clothes, and
6	wmn_4504322055602176	2020-07-09 20:43:11.055 UTC	Female	16 to 25 years old	City center or metropolitan area	I can afford and re expenses, and
7	wmn_4504369904222208	2020-07-18 12:52:31.482 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I cannot a enough foo my fa
8	wmn_4504469091123200	2020-07-16 16:03:44.066 UTC	Female	36 to 45 years old	City center or metropolitan area	I can afford and re expenses, but
9	wmn_4504687899574272	2020-07-17 07:16:32.082 UTC	Female	36 to 45 years old	City center or metropolitan area	I can afford and re expenses, but

10 rows × 46 columns

In [173…

```
df_T.head(10)
```

Out[173…

	0	1	
observation_id	wmn_4503683847159808	wmn_4503772699295744	wmn_4504010469146
submitted_time	2020-07-09 23:19:01.982 UTC	2020-07-09 21:22:15.864 UTC	2020-07-10 05:09:07
gender	Female	Female	Fe
age	26 to 35 years old	16 to 25 years old	16 to 25 years
geography	City center or metropolitan area	Rural	F
financial_situation	I cannot afford enough food for my family	I cannot afford enough food for my family	I can afford food, nothing
education	College or university	Secondary/high school	College or unive
employment_status	Unemployed	Student	Stu
ethnicity	Mestizo	Tagalog	Hiligay
religion	Catholicism	Muslim	Christi

10 rows × 12354 columns

In [174…

```
df.tail(10) # ostatnie 10 wierszy ramki danych
```

Out[174…

	observation_id	submitted_time	gender	age	geography	financial_
12344	wmn_6752309616050176	2020-07-11 14:37:07.551 UTC	Female	26 to 35 years old	City center or metropolitan area	I can cc af clothe
12345	wmn_6752631872815104	2020-07-10 02:25:50.01 UTC	Female	26 to 35 years old	City center or metropolitan area	I can a a expense
12346	wmn_6752968893530112	2020-07-14 05:19:46.429 UTC	Female	16 to 25 years old	City center or metropolitan area	I can a a expenses
12347	wmn_6753819934588928	2020-07-25 17:34:27.837 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can a a expense
12348	wmn_6753897143336960	2020-07-10 11:49:14.64 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can a a expense
12349	wmn_6754210441068544	2020-07-16 15:46:12.095 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can a a expense
12350	wmn_6754415891709952	2020-07-10 09:57:24.863 UTC	Female	16 to 25 years old	City center or metropolitan area	I can a a expense
12351	wmn_6754483574145024	2020-07-19 17:50:01.295 UTC	Female	26 to 35 years old	City center or metropolitan area	I can a a expense
12352	wmn_6755256899993600	2020-07-11 16:09:09.78 UTC	Female	36 to 45 years old	Rural	I can cc af clothe
12353	wmn_6755376524689408	2020-07-17 03:19:00.388 UTC	Female	26 to 35 years old	City center or metropolitan area	I can a a expenses

10 rows × 46 columns

In [175…

```
df.info() # informacja o ramce danych
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12354 entries, 0 to 12353
Data columns (total 46 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   observation_id                 12354 non-null  object 
 1   submitted_time                 12354 non-null  object 
 2   gender                         12354 non-null  object 
 3   age                            12354 non-null  object 
 4   geography                      12354 non-null  object 
 5   financial_situation            12354 non-null  object 
 6   education                      12354 non-null  object 
 7   employment_status              12354 non-null  object 
 8   ethnicity                      12354 non-null  object 
 9   religion                       12354 non-null  object 
 10  wmn_hh                         12354 non-null  object 
 11  wmn_pregnancy_desire           12354 non-null  object 
 12  wmn_pregnancy_change           11351 non-null  object 
 13  wmn_pregnancy_change_how       3030 non-null   object 
 14  wmn_con                        12354 non-null  object 
 15  wmn_con_type                   5426 non-null   object 
 16  wmn_pre_con_access_difficulty  3460 non-null   object 
 17  wmn_pre_missed_dose_pills      186 non-null    object 
 18  wmn_pre_con_needed             3185 non-null   object 
 19  wmn_pre_con_accessed           1825 non-null   object 
 20  wmn_pre_injectable_missed      94 non-null     float64
 21  wmn_pre_iud_missed             75 non-null     float64
 22  wmn_pre_con_missed_why         437 non-null    object 
 23  wmn_pre_con_missed_why_other   34 non-null     object 
 24  wmn_post_con_access_difficulty 3460 non-null   object 
 25  wmn_post_missed_dose_pills     184 non-null    object 
 26  wmn_post_con_needed            3133 non-null   object 
 27  wmn_post_con_accessed          1825 non-null   object 
 28  wmn_post_injectable_missed     110 non-null    float64
 29  wmn_post_iud_missed            60 non-null     float64
 30  wmn_post_con_missed_why        443 non-null    object 
 31  wmn_post_con_missed_why_other  23 non-null     object 
 32  wmn_alone                      12354 non-null  object 
 33  wmn_how_safe                   5282 non-null   object 
 34  wmn_safe_change                5282 non-null   object 
 35  wmn_safe_place                 5282 non-null   object 
 36  wmn_pre_safe_place             2795 non-null   object 
 37  wmn_post_safe_place            2795 non-null   object 
 38  wmn_safe_place_no_access       2795 non-null   object 
 39  wmn_safe_place_no_access_why   849 non-null    object 
 40  wmn_pre_help                   5282 non-null   object 
 41  wmn_post_help                  5282 non-null   object 
 42  wmn_post_no_help               5282 non-null   object 
 43  wmn_no_help_why                848 non-null    object 
 44  country                        12354 non-null  object 
 45  user_id                        12354 non-null  object 
dtypes: float64(4), object(42)
memory usage: 4.3+ MB
```

In [176…

```
df.shape # pokazuje, ile wierszy i kolumn znajduje się w ramce danych
```

Out[176…

```
(12354, 46)
```

In [177…

```
df.describe()
```

Out[177…

	wmn_pre_injectable_missed	wmn_pre_iud_missed	wmn_post_injectable_missed	wı
count	94.000000	7.500000e+01	110.000000	
mean	214.765957	1.033763e+07	232.845455	
std	2062.634850	8.952644e+07	2418.242269	
min	0.000000	0.000000e+00	0.000000	
25%	1.000000	1.000000e+00	1.000000	
50%	2.000000	2.000000e+00	2.000000	
75%	2.750000	3.000000e+00	3.000000	
max	20000.000000	7.753217e+08	25365.000000	

In [178…

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

Out[178…

	observation_id	submitted_time	gender	age	geography	financial_
count	12354	12354	12354	12354	12354	
unique	12354	12354	3	6	4	
top	wmn_6755376524689408	2020-07-17 03:19:00.388 UTC	Female	16 to 25 years old	City center or metropolitan area	I can af ar expenses
freq	1	1	12331	5672	5274	
mean	NaN	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	NaN	

11 rows × 46 columns

In [179…

```
df.dropna(how='all', inplace=True)#df.dropna(inplace=True) # usuwanie brakującyc
df.head()
```

Out[179…

	observation_id	submitted_time	gender	age	geography	financial_situa
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	I cannot a enough foo my fa
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I cannot a enough foo my fa
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can afford t but nothing
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can afford and re expenses, and
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can afford and re expenses, and

5 rows × 46 columns

In [180…

```
df["age"] # wybór kolumny
```

Out[180…

```
0        26 to 35 years old
1        16 to 25 years old
2        16 to 25 years old
3        16 to 25 years old
4        26 to 35 years old
                ...
12349    26 to 35 years old
12350    16 to 25 years old
12351    26 to 35 years old
12352    36 to 45 years old
12353    26 to 35 years old
Name: age, Length: 12354, dtype: object
```

In [181…

```
df[["observation_id","gender","age"]] # wybór kilku kolumn jednocześnie
```

Out[181…

	observation_id	gender	age
0	wmn_4503683847159808	Female	26 to 35 years old
1	wmn_4503772699295744	Female	16 to 25 years old
2	wmn_4504010469146624	Female	16 to 25 years old
3	wmn_4504035500752896	Female	16 to 25 years old
4	wmn_4504181395423232	Female	26 to 35 years old
...	...	...	...
12349	wmn_6754210441068544	Female	26 to 35 years old
12350	wmn_6754415891709952	Female	16 to 25 years old
12351	wmn_6754483574145024	Female	26 to 35 years old
12352	wmn_6755256899993600	Female	36 to 45 years old
12353	wmn_6755376524689408	Female	26 to 35 years old

12354 rows × 3 columns

In [182…

```
df_wybrane = df[["observation_id","gender","age"]]
df_wybrane.head(10)
```

Out[182…

	observation_id	gender	age
0	wmn_4503683847159808	Female	26 to 35 years old
1	wmn_4503772699295744	Female	16 to 25 years old
2	wmn_4504010469146624	Female	16 to 25 years old
3	wmn_4504035500752896	Female	16 to 25 years old
4	wmn_4504181395423232	Female	26 to 35 years old
5	wmn_4504301990051840	Female	26 to 35 years old
6	wmn_4504322055602176	Female	16 to 25 years old
7	wmn_4504369904222208	Female	16 to 25 years old
8	wmn_4504469091123200	Female	36 to 45 years old
9	wmn_4504687899574272	Female	36 to 45 years old

In [183…

```
df.head(10)
```

Out[183…

	observation_id	submitted_time	gender	age	geography	financial_situa
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	I cannot a enough foo my fa
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I cannot a enough foo my fa
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can afford t but nothing
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can afford and re expenses, and
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can afford and re expenses, and
5	wmn_4504301990051840	2020-07-10 11:27:16.581 UTC	Female	26 to 35 years old	City center or metropolitan area	I can comfor afford t clothes, and
6	wmn_4504322055602176	2020-07-09 20:43:11.055 UTC	Female	16 to 25 years old	City center or metropolitan area	I can afford and re expenses, and
7	wmn_4504369904222208	2020-07-18 12:52:31.482 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I cannot a enough foo my fa
8	wmn_4504469091123200	2020-07-16 16:03:44.066 UTC	Female	36 to 45 years old	City center or metropolitan area	I can afford and re expenses, but
9	wmn_4504687899574272	2020-07-17 07:16:32.082 UTC	Female	36 to 45 years old	City center or metropolitan area	I can afford and re expenses, but

10 rows × 46 columns

In [184…

```
df[["observation_id","gender","age"]] # wybór kilku kolumn jednocześnie
```

Out[184…

	observation_id	gender	age
0	wmn_4503683847159808	Female	26 to 35 years old
1	wmn_4503772699295744	Female	16 to 25 years old
2	wmn_4504010469146624	Female	16 to 25 years old
3	wmn_4504035500752896	Female	16 to 25 years old
4	wmn_4504181395423232	Female	26 to 35 years old
...	...	...	...
12349	wmn_6754210441068544	Female	26 to 35 years old
12350	wmn_6754415891709952	Female	16 to 25 years old
12351	wmn_6754483574145024	Female	26 to 35 years old
12352	wmn_6755256899993600	Female	36 to 45 years old
12353	wmn_6755376524689408	Female	26 to 35 years old

12354 rows × 3 columns

In [185…

```
df["age"] # wybór kolumny
```

Out[185…

```
0        26 to 35 years old
1        16 to 25 years old
2        16 to 25 years old
3        16 to 25 years old
4        26 to 35 years old
                ...
12349    26 to 35 years old
12350    16 to 25 years old
12351    26 to 35 years old
12352    36 to 45 years old
12353    26 to 35 years old
Name: age, Length: 12354, dtype: object
```

In [186…

```
df.age
```

Out[186…

```
0        26 to 35 years old
1        16 to 25 years old
2        16 to 25 years old
3        16 to 25 years old
4        26 to 35 years old
                ...
12349    26 to 35 years old
12350    16 to 25 years old
12351    26 to 35 years old
12352    36 to 45 years old
12353    26 to 35 years old
Name: age, Length: 12354, dtype: object
```

In [187…

```
df["age"][0]#indeks
```

Out[187…

```
'26 to 35 years old'
```

In [188…

```
df.loc[1:4, "observation_id":"geography"] # zakres
```

Out[188…

	observation_id	submitted_time	gender	age	geography
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban

In [189…

```
df.iloc[100:110, 0:3] #zakres2
```

Out[189…

	observation_id	submitted_time	gender
100	wmn_4522790814482432	2020-07-09 19:28:22.399 UTC	Female
101	wmn_4522800548413440	2020-07-10 00:35:20.912 UTC	Female
102	wmn_4522833599528960	2020-07-15 13:23:42.518 UTC	Female
103	wmn_4522903023648768	2020-07-11 23:35:50.9 UTC	Female
104	wmn_4522993754832896	2020-07-10 16:27:44.32 UTC	Female
105	wmn_4523099619065856	2020-07-22 15:36:10.319 UTC	Female
106	wmn_4523200282361856	2020-07-10 06:08:20.411 UTC	Female
107	wmn_4523217495785472	2020-07-16 21:41:36.802 UTC	Female
108	wmn_4523289704923136	2020-07-09 19:52:14.966 UTC	Female
109	wmn_4523372245680128	2020-07-10 15:57:04.585 UTC	Female

In [190…

```
df[df["gender"] == "Female"] # wg wartosci
```

Out[190…

	observation_id	submitted_time	gender	age	geography	financial_
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	I can enoug
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I can enoug
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can af but nc
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can a a expenses
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can a a expenses
...	...	...	...	...	...	
12349	wmn_6754210441068544	2020-07-16 15:46:12.095 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can a a expense
12350	wmn_6754415891709952	2020-07-10 09:57:24.863 UTC	Female	16 to 25 years old	City center or metropolitan area	I can a a expense
12351	wmn_6754483574145024	2020-07-19 17:50:01.295 UTC	Female	26 to 35 years old	City center or metropolitan area	I can a a expense
12352	wmn_6755256899993600	2020-07-11 16:09:09.78 UTC	Female	36 to 45 years old	Rural	I can cc af clothe
12353	wmn_6755376524689408	2020-07-17 03:19:00.388 UTC	Female	26 to 35 years old	City center or metropolitan area	I can a a expenses

12331 rows × 46 columns

In [191…

```
df[(df["gender"].str.startswith("F")) & (df["geography"] == "Rural")] # wiecej w
```

Out[191...

	observation_id	submitted_time	gender	age	geography	financial_situ
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I cannot enough fc my
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can afforc but nothir
10	wmn_4504764873441280	2020-07-23 00:29:41.766 UTC	Female	16 to 25 years old	Rural	I can afforc but nothir
13	wmn_4505544242233344	2020-07-10 01:19:58.4 UTC	Female	16 to 25 years old	Rural	I cannot enough fc my
14	wmn_4505874686279680	2020-07-18 06:43:04.442 UTC	Female	16 to 25 years old	Rural	I can affor and r expenses, bu
...	...	...	...	...	...	
12318	wmn_6748715063967744	2020-07-11 07:55:29.714 UTC	Female	26 to 35 years old	Rural	I can affor and r expenses, bu
12328	wmn_6749915977089024	2020-07-10 09:24:53.523 UTC	Female	16 to 25 years old	Rural	I can affor and r expenses, ar
12331	wmn_6750131060998144	2020-07-11 23:35:42.902 UTC	Female	36 to 45 years old	Rural	I can affor and r expenses, ar
12336	wmn_6750879257722880	2020-07-16 11:20:52.145 UTC	Female	26 to 35 years old	Rural	I can afforc but nothir
12352	wmn_6755256899993600	2020-07-11 16:09:09.78 UTC	Female	36 to 45 years old	Rural	I can comfc afforc clothes, a

3009 rows × 46 columns

In [192...

```
df[df["financial_situation"].str.contains("food", case=False, na=False)] # zawi
```

Out[192…

	observation_id	submitted_time	gender	age	geography	financial_
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	I can enoug
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I can enoug
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can af but nc
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can a a expense
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can a a expense
...	...	...	...	...	...	
12349	wmn_6754210441068544	2020-07-16 15:46:12.095 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can a a expense
12350	wmn_6754415891709952	2020-07-10 09:57:24.863 UTC	Female	16 to 25 years old	City center or metropolitan area	I can a a expense
12351	wmn_6754483574145024	2020-07-19 17:50:01.295 UTC	Female	26 to 35 years old	City center or metropolitan area	I can a a expense
12352	wmn_6755256899993600	2020-07-11 16:09:09.78 UTC	Female	36 to 45 years old	Rural	I can cc af clothe
12353	wmn_6755376524689408	2020-07-17 03:19:00.388 UTC	Female	26 to 35 years old	City center or metropolitan area	I can a a expense

12346 rows × 46 columns

In [193…

```
df[~df["financial_situation"].str.contains("food", case=False, na=False)] # nie
```

Out[193…

	observation_id	submitted_time	gender	age	geography	financial
197	wmn_4540938363076608	2020-07-20 01:56:02.127 UTC	Not Available	Not Available	Not Available	No
390	wmn_4576419524116480	2020-07-19 12:17:38.356 UTC	Not Available	Not Available	Not Available	No
1207	wmn_4725513139781632	2020-07-14 20:12:05.237 UTC	Not Available	Not Available	Not Available	No
1470	wmn_4780876039979008	2020-07-20 01:46:22.047 UTC	Not Available	Not Available	Not Available	No
5118	wmn_5434910908350464	2020-07-28 22:14:08.94 UTC	Not Available	Not Available	Not Available	No
6050	wmn_5607956781727744	2020-07-10 00:51:52.606 UTC	Not Available	Not Available	Not Available	No
6724	wmn_5729962914217984	2020-07-09 19:29:17.524 UTC	Not Available	Not Available	Not Available	No
9346	wmn_6206612076494848	2020-07-15 08:32:29.099 UTC	Not Available	Not Available	Not Available	No

8 rows × 46 columns

In [194…

```
df.drop("education", axis=1, inplace = True) # usuń kolumnę measure
```

In [195…

```
df = pd.read_csv("IHME_PREM_WMN_HEALTH_2020_Y2011M10D11.csv", encoding = "utf-8"
```

In [196…

```
df["education2"] = df["education"]
# utwórz nową kolumnę na podst. obecnej
df.head()
```

Out[196...

	observation_id	submitted_time	gender	age	geography	financial_situa
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	I cannot a enough foo my f
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I cannot a enough foo my f
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can afford t but nothing
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can afford and re expenses, and
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can afford and re expenses, and

5 rows × 47 columns

In [197...

```
df.drop("education2", axis=1, inplace = True) # usuń kolumnę education2
```

In [198...

```
df.head()
```

Out[198…

	observation_id	submitted_time	gender	age	geography	financial_situa
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	I cannot a enough foo my fa
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I cannot a enough foo my fa
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can afford f but nothing
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can afford and re expenses, and
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can afford and re expenses, and

5 rows × 46 columns

In [199…

```
df.rename(columns = {"gender": "gender2"}, inplace = True) # zmień nazwę kolumny
df.head()
```

Out[199…

	observation_id	submitted_time	gender2	age	geography	financial_situ
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	I cannot enough fo my
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I cannot enough fo my
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can afforc but nothir
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can afforc and r expenses, an
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can afforc and r expenses, an

5 rows × 46 columns

In [200…

```
df.to_csv("lab1result.csv")
```

In [201…

```
df.loc[:10].to_csv("lab1result_10.csv")
```

In [202…

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

Out[202…

```
'2020-07-28 23:18:51.484 UTC'
```

In [203…

```
df['submitted_time'].min() # min wartość jedna kolumna na raz
```

Out[203…

```
'2020-07-09 19:24:11.039 UTC'
```

In [204…

```
df['observation_id'].count() # liczba rekordów
```

Out[204…

```
np.int64(12354)
```

In [205…

```
df['geography'].unique() # wartości unikatowe
```

Out[205…

```
array(['City center or metropolitan area', 'Rural', 'Suburban/Peri-urban',
       'Not Available'], dtype=object)
```

In [206…

```
df['geography'].value_counts() # liczba rekordów pasujących do unikalnych wartoś
```

```
Out[206…   geography
           City center or metropolitan area    5274
           Suburban/Peri-urban                 4061
           Rural                               3011
           Not Available                          8
           Name: count, dtype: int64
```

```
In [207…   df.sort_values(['submitted_time'], ascending = True) # sortowanie
```

Out[207…

	observation_id	submitted_time	gender2	age	geography	financ
6185	wmn_5634978501361664	2020-07-09 19:24:11.039 UTC	Female	26 to 35 years old	City center or metropolitan area	I ca exper
10047	wmn_6333227041292288	2020-07-09 19:24:40.955 UTC	Female	26 to 35 years old	City center or metropolitan area	I ca exper
1054	wmn_4696779942789120	2020-07-09 19:25:02.99 UTC	Female	26 to 35 years old	Rural	I car but
8512	wmn_6062004084408320	2020-07-09 19:26:36.611 UTC	Female	16 to 25 years old	Rural	I enc
11833	wmn_6661922197078016	2020-07-09 19:26:36.853 UTC	Female	26 to 35 years old	City center or metropolitan area	I ca exper
...	...	...	...	...	...	
4205	wmn_5267029226684416	2020-07-28 21:56:48.289 UTC	Female	26 to 35 years old	City center or metropolitan area	I ca exper
5118	wmn_5434910908350464	2020-07-28 22:14:08.94 UTC	Not Available	Not Available	Not Available	
10662	wmn_6451972971692032	2020-07-28 22:58:20.61 UTC	Female	16 to 25 years old	City center or metropolitan area	I car but
11250	wmn_6558735138029568	2020-07-28 23:04:37.556 UTC	Female	26 to 35 years old	City center or metropolitan area	I car clo
9210	wmn_6182474712612864	2020-07-28 23:18:51.484 UTC	Female	16 to 25 years old	City center or metropolitan area	I ca exper

12354 rows × 46 columns

In [208…

```
df["observation_id2"] = df["observation_id"].str.split('_', n=1).str[1].astype("
```

In [209…

```
df.head(10)
```

Out[209…

	observation_id	submitted_time	gender2	age	geography	financial_situ
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	I cannot enough fo my
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	I cannot enough fo my
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I can afforc but nothir
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can afforc and r expenses, an
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can afforc and r expenses, an
5	wmn_4504301990051840	2020-07-10 11:27:16.581 UTC	Female	26 to 35 years old	City center or metropolitan area	I can comfc afforc clothes, ar
6	wmn_4504322055602176	2020-07-09 20:43:11.055 UTC	Female	16 to 25 years old	City center or metropolitan area	I can afforc and r expenses, an
7	wmn_4504369904222208	2020-07-18 12:52:31.482 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I cannot enough fo my
8	wmn_4504469091123200	2020-07-16 16:03:44.066 UTC	Female	36 to 45 years old	City center or metropolitan area	I can afforc and r expenses, bu
9	wmn_4504687899574272	2020-07-17 07:16:32.082 UTC	Female	36 to 45 years old	City center or metropolitan area	I can afforc and r expenses, bu

10 rows × 47 columns

In [210…

```
df.nlargest(10,'wmn_post_injectable_missed') # 10 najwyższych wartości dla kolum
```

Out[210…

	observation_id	submitted_time	gender2	age	geography	financial_
7521	wmn_5881289107570688	2020-07-17 20:17:24.774 UTC	Female	16 to 25 years old	Rural	I can enoug
8904	wmn_6129233710809088	2020-07-21 20:19:11.507 UTC	Female	16 to 25 years old	City center or metropolitan area	I can af but nc
2534	wmn_4962102554132480	2020-07-10 02:07:21.847 UTC	Female	26 to 35 years old	City center or metropolitan area	I can a a expenses
1078	wmn_4702068205158400	2020-07-23 03:51:29.797 UTC	Female	16 to 25 years old	City center or metropolitan area	I can a a expense
1795	wmn_4835970770010112	2020-07-11 15:40:23.91 UTC	Female	16 to 25 years old	Rural	I can af but nc
6698	wmn_5725235262521344	2020-07-27 05:00:13.102 UTC	Female	16 to 25 years old	City center or metropolitan area	I can enoug
7611	wmn_5894380838977536	2020-07-09 22:38:13.923 UTC	Female	36 to 45 years old	Suburban/Peri-urban	I can enoug
2673	wmn_4986662989070336	2020-07-11 16:06:53.503 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can enoug
3035	wmn_5053243672756224	2020-07-16 15:12:22.233 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I can a a expenses
4295	wmn_5278829582221312	2020-07-22 03:22:53.126 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can a a expenses

10 rows × 47 columns

In [211…

```
grupa = df.groupby("education").mean(numeric_only=True)
grupa
```

Out[211…

	wmn_pre_injectable_missed	wmn_pre_iud_missed	wmn_post_injectable_r
education			
College or university	626.750000	2.500000e+00	2.4
No formal education	3.000000	NaN	3.0
Not Available	NaN	NaN	
Post graduate	2.166667	2.142857e+00	2.1
Prefer not to answer	NaN	5.000000e-01	1.0
Primary school	2.500000	1.938304e+08	2.7
Secondary/high school	1.975000	2.941176e+00	636.0
Technical school	2.461538	5.181818e+00	2.4

In [212…

```
grupa.index # indeksy
```

Out[212…

```
Index(['College or university', 'No formal education', 'Not Available',
       'Post graduate', 'Prefer not to answer', 'Primary school',
       'Secondary/high school', 'Technical school'],
      dtype='object', name='education')
```

In [213…

```
grupa.columns # multiindeks
```

Out[213…

```
Index(['wmn_pre_injectable_missed', 'wmn_pre_iud_missed',
       'wmn_post_injectable_missed', 'wmn_post_iud_missed', 'observation_id2'],
      dtype='object')
```

In [214…

```
df_t = df.groupby('education').agg({'country': ['count'],
                        'wmn_post_injectable_missed': ['mean', 'median']})
# dla kolumny 'education' count (liczba unikalnych wartości), dla 'observation_i
df_t
```

Out[214…

	country	wmn_post_injectable_missed	
	count	mean	median
education			
College or university	5726	2.473684	1.5
No formal education	52	3.000000	3.0
Not Available	8	NaN	NaN
Post graduate	774	2.166667	2.0
Prefer not to answer	217	1.000000	1.0
Primary school	344	2.750000	2.5
Secondary/high school	3707	636.050000	2.0
Technical school	1526	2.450000	2.0

In [215…

```
df.sort_values(['wmn_post_injectable_missed'], ascending = False) # sortowanie
```

Out[215…

	observation_id	submitted_time	gender2	age	geography	financia
7521	wmn_5881289107570688	2020-07-17 20:17:24.774 UTC	Female	16 to 25 years old	Rural	I ca enou
8904	wmn_6129233710809088	2020-07-21 20:19:11.507 UTC	Female	16 to 25 years old	City center or metropolitan area	I can a but r
2534	wmn_4962102554132480	2020-07-10 02:07:21.847 UTC	Female	26 to 35 years old	City center or metropolitan area	I can expens
1078	wmn_4702068205158400	2020-07-23 03:51:29.797 UTC	Female	16 to 25 years old	City center or metropolitan area	I can expens
1795	wmn_4835970770010112	2020-07-11 15:40:23.91 UTC	Female	16 to 25 years old	Rural	I can a but r
...	...	...	...	...	...	
12349	wmn_6754210441068544	2020-07-16 15:46:12.095 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I can expens
12350	wmn_6754415891709952	2020-07-10 09:57:24.863 UTC	Female	16 to 25 years old	City center or metropolitan area	I can expens
12351	wmn_6754483574145024	2020-07-19 17:50:01.295 UTC	Female	26 to 35 years old	City center or metropolitan area	I can expens
12352	wmn_6755256899993600	2020-07-11 16:09:09.78 UTC	Female	36 to 45 years old	Rural	I can c a cloth
12353	wmn_6755376524689408	2020-07-17 03:19:00.388 UTC	Female	26 to 35 years old	City center or metropolitan area	I can expens

12354 rows × 47 columns

In [216…

```
pivot = pd.pivot_table(
    df,
    values="geography",        # kolumna z wartościami
    index="wmn_safe_place_no_access_why",       # wiersze
    columns="wmn_post_safe_place",       # kolumny
    aggfunc="count"          # funkcja agregująca (np. mean, sum, count)
)
pivot
```

Out[216…

wmn_post_safe_place	Decline to respond	Don't know	Every day	Never	Once a month	Once a week	Rarely
wmn_safe_place_no_access_why							
Afraid of consequences	NaN	NaN	8.0	1.0	9.0	14.0	11.0
Decline to respond	2.0	NaN	5.0	1.0	1.0	3.0	NaN
Fear of being infected with COVID-19	NaN	2.0	15.0	17.0	19.0	24.0	40.0
No transportation	1.0	1.0	11.0	3.0	11.0	8.0	10.0
Other	NaN	3.0	5.0	4.0	2.0	9.0	10.0
Place was closed or unavailable for reason other than lockdown	NaN	NaN	39.0	3.0	15.0	35.0	15.0
Unable to access place due to lockdown	6.0	4.0	189.0	46.0	47.0	108.0	92.0

In [217…

```
pivot.index
```

Out[217…

```
Index(['Afraid of consequences', 'Decline to respond',
       'Fear of being infected with COVID-19', 'No transportation', 'Other',
       'Place was closed or unavailable for reason other than lockdown',
       'Unable to access place due to lockdown'],
      dtype='object', name='wmn_safe_place_no_access_why')
```

In [218…

```
pivot.columns
```

Out[218…

```
Index(['Decline to respond', 'Don't know', 'Every day', 'Never',
       'Once a month', 'Once a week', 'Rarely'],
      dtype='object', name='wmn_post_safe_place')
```

In [219…

```
pivot2 = df.pivot_table(values='geography', index=['wmn_safe_place_no_access_why
                        margins=False, dropna=True, fill_value=None)
# ustaw multi-indeks
pivot2
```

Out[219…

	wmn_post_safe_place	Every day	Never	Once a week
wmn_safe_place_no_access_why	**wmn_post_injectable_missed**			
Fear of being infected with COVID-19	**2.0**	NaN	NaN	1.0
Other	**4.0**	NaN	1.0	NaN
Place was closed or unavailable for reason other than lockdown	**3.0**	1.0	NaN	NaN
	6.0	NaN	NaN	1.0
	25365.0	NaN	NaN	1.0
Unable to access place due to lockdown	**2.0**	1.0	NaN	NaN
	3.0	1.0	NaN	NaN
	0.0	1.0	NaN	NaN
	8.0	1.0	NaN	NaN
	12.0	NaN	NaN	1.0

In [220…

```
import matplotlib.pyplot as plt # zaimportuj moduł pyplot z biblioteki matplotli
%matplotlib inline
# wskazanie, że wykresy należy rysować bezpośrednio w zeszycie, a nie w osobnej
```

In [221…

```
pivot.plot(kind = 'line')
```

Out[221…

```
<Axes: xlabel='wmn_safe_place_no_access_why'>
```

In [222…

```
df_bar = df[(df['country']== 'Poland') & (df['gender2'] == 'Female')].pivot_tabl
                index='geography', columns='gender2', aggfunc='count',
                      fill_value=None, margins=False, dropna=True)
df_bar.plot(kind = 'line')
plt.ylabel('Y')
plt.title('tytul')
```

Out[222…

```
Text(0.5, 1.0, 'tytul')
```

In [223…

```
df1 = df
df1.rename(columns = {'observation_id': 'observation_iddf1', 'submitted_time':'s
df2 = df
df2.rename(columns = {'observation_id': 'observation_iddf2', 'submitted_time':'s
```

In [224…

```
dict_city = {"City" : ["Warszawa", "Łódź", "Poznań"],
            "Population" : [12678079,   5398064,  1625631]}

df1 = pd.DataFrame(dict_city) # tworzenie ramki danych ze słownika
dict_city2 = {"City" : ["Warszawa", "Łódź", "Poznań"],
            "Population" : [1267807933,   539806433,  162563133]}

df2 = pd.DataFrame(dict_city2) # tworzenie ramki danych ze słownika

merged_df = pd.merge(df1, df2, on=['City'], how='inner')
merged_df
```

Out[224…

	City	Population_x	Population_y
0	Warszawa	12678079	1267807933
1	Łódź	5398064	539806433
2	Poznań	1625631	162563133

In [225…

```
concat_df = pd.concat([df1, df2], axis=0)
concat_df
```

Out[225…

	City	Population
0	Warszawa	12678079
1	Łódź	5398064
2	Poznań	1625631
0	Warszawa	1267807933
1	Łódź	539806433
2	Poznań	162563133

In [226…

```
df_all = df
df_all["info"] = df_all["geography"] + df_all["financial_situation"] + df_all["e
df_all
```

Out[226…

	observation_iddf1	submitted_timedf1	gender2	age	geography	fina
0	wmn_4503683847159808	2020-07-09 23:19:01.982 UTC	Female	26 to 35 years old	City center or metropolitan area	e
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	e
2	wmn_4504010469146624	2020-07-10 05:09:07.359 UTC	Female	16 to 25 years old	Rural	I c b
3	wmn_4504035500752896	2020-07-11 16:59:49.85 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I c exp
4	wmn_4504181395423232	2020-07-11 18:43:35.954 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I c exp
...	...	...	...	...	...	
12349	wmn_6754210441068544	2020-07-16 15:46:12.095 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I c exp
12350	wmn_6754415891709952	2020-07-10 09:57:24.863 UTC	Female	16 to 25 years old	City center or metropolitan area	I c exp
12351	wmn_6754483574145024	2020-07-19 17:50:01.295 UTC	Female	26 to 35 years old	City center or metropolitan area	I c exp
12352	wmn_6755256899993600	2020-07-11 16:09:09.78 UTC	Female	36 to 45 years old	Rural	I c c
12353	wmn_6755376524689408	2020-07-17 03:19:00.388 UTC	Female	26 to 35 years old	City center or metropolitan area	I c exp

12354 rows × 48 columns

In [227…

```
# tworzymy listę
religions = ['Catholism', 'Muslim']
# za pomocą funkcji lambda określamy, że jeśli religia („religion”) jest zawarty
# będzie true, jeśli nie, to false
```

```
df_all['sprreligia'] = df_all['religion'].apply(lambda x: True if x in religions
df_all[df_all['sprreligia'] == True]
```

Out[227…

	observation_iddf1	submitted_timedf1	gender2	age	geography	fina
1	wmn_4503772699295744	2020-07-09 21:22:15.864 UTC	Female	16 to 25 years old	Rural	er
8	wmn_4504469091123200	2020-07-16 16:03:44.066 UTC	Female	36 to 45 years old	City center or metropolitan area	I c exp
12	wmn_4505272015126528	2020-07-16 17:21:24.686 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I c exp
20	wmn_4506834611470336	2020-07-10 05:23:02.996 UTC	Female	36 to 45 years old	City center or metropolitan area	I c exp
30	wmn_4508474986987520	2020-07-10 03:59:23.848 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I c exp
...	...	...	...	...	...	
12318	wmn_6748715063967744	2020-07-11 07:55:29.714 UTC	Female	26 to 35 years old	Rural	I c exp
12333	wmn_6750260782432256	2020-07-10 09:28:33.392 UTC	Female	26 to 35 years old	City center or metropolitan area	I c c
12334	wmn_6750274942402560	2020-07-19 09:22:24.206 UTC	Female	16 to 25 years old	City center or metropolitan area	I c b
12339	wmn_6751177925722112	2020-07-10 08:13:39.04 UTC	Female	16 to 25 years old	Suburban/Peri-urban	I c exp
12349	wmn_6754210441068544	2020-07-16 15:46:12.095 UTC	Female	26 to 35 years old	Suburban/Peri-urban	I c exp

1359 rows × 49 columns

In [228…

```
for chunk_df in pd.read_csv('IHME_PREM_WMN_HEALTH_2020_Y2011M10D11.csv',
                             chunksize = 50000):
    print("CHUNK DF")
    print(chunk_df.head())
```

```
CHUNK DF
         observation_id               submitted_time  gender  \
0  wmn_4503683847159808  2020-07-09 23:19:01.982 UTC  Female
1  wmn_4503772699295744  2020-07-09 21:22:15.864 UTC  Female
2  wmn_4504010469146624  2020-07-10 05:09:07.359 UTC  Female
3  wmn_4504035500752896   2020-07-11 16:59:49.85 UTC  Female
4  wmn_4504181395423232  2020-07-11 18:43:35.954 UTC  Female

                  age                         geography  \
0  26 to 35 years old  City center or metropolitan area
1  16 to 25 years old                             Rural
2  16 to 25 years old                             Rural
3  16 to 25 years old               Suburban/Peri-urban
4  26 to 35 years old               Suburban/Peri-urban

                                 financial_situation              education  \
0          I cannot afford enough food for my family  College or university
1          I cannot afford enough food for my family  Secondary/high school
2                I can afford food, but nothing else  College or university
3  I can afford food and regular expenses, and bu...  College or university
4  I can afford food and regular expenses, and bu...  College or university

    employment_status   ethnicity      religion  ... wmn_pre_safe_place  \
0          Unemployed     Mestizo   Catholicism  ...                NaN
1             Student     Tagalog        Muslim  ...                NaN
2             Student  Hiligaynon  Christianity  ...                NaN
3          Unemployed        Thai      Buddhism  ...                NaN
4  Employed full-time     African  Christianity  ...                NaN

  wmn_post_safe_place wmn_safe_place_no_access wmn_safe_place_no_access_why  \
0                 NaN                      NaN                          NaN
1                 NaN                      NaN                          NaN
2                 NaN                      NaN                          NaN
3                 NaN                      NaN                          NaN
4                 NaN                      NaN                          NaN

  wmn_pre_help wmn_post_help wmn_post_no_help wmn_no_help_why  \
0          NaN           NaN              NaN             NaN
1          NaN           NaN              NaN             NaN
2          NaN           NaN              NaN             NaN
3          NaN           NaN              NaN             NaN
4          NaN           NaN              NaN             NaN

                       country               user_id
0                      Ecuador  wmn_5900473574883328
1                  Philippines  wmn_5702261783658496
2                  Philippines  wmn_5652767014387712
3                     Thailand  wmn_6411372690669568
4  United Republic of Tanzania  wmn_6215734184378368

[5 rows x 46 columns]
```

In [229…

```
# zastosuj metodę groupby oddzielnie do każdej części, a następnie połącz wynik
new_df = pd.DataFrame() # pusta ramka danych
for chunk_df in pd.read_csv('IHME_PREM_WMN_HEALTH_2020_Y2011M10D11.csv',
                     chunksize = 50000):
    result = chunk_df.groupby(['geography', 'age']).agg(
        country_poland=('country', lambda x: (x == 'Poland').sum()),
        religion_count=('religion', 'count')
    )
    new_df = pd.concat([new_df,result])
```

In [230…

```
new_df.head()
```

Out[230…

		country_poland	religion_count
geography	**age**		
City center or metropolitan area	**16 to 25 years old**	6	2271
	26 to 35 years old	7	1975
	36 to 45 years old	6	844
	Not Available	0	2
	Over 45 years old	0	129

In [ ]: