SPRAWOZDANIE

Zajęcia: Nauka o danych I

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 3 Data 19.10.2024 Temat: "Wykorzystanie pakietu Pandas do manipulacji i przetwarzania danych w Pythonie" Wariant 10	Anna Więzik Informatyka II stopień, niestacjonarne, 1 semestr, gr.1b

1. Polecenie:

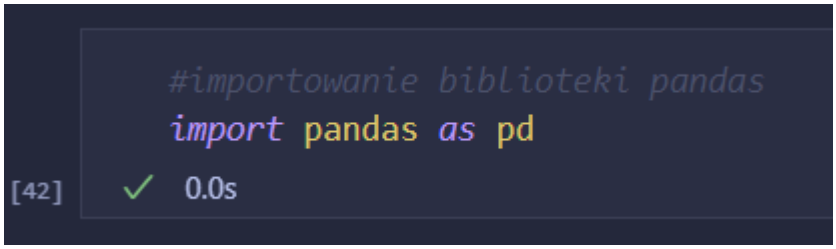
   Premise Child Health COVID-19 Health Services Disruption Survey 2020
   http://ghdx.healthdata.org/record/ihme-data/premise-c hild-health-covid-19-health-services-disruption-survey-2 020

2. Link do repozytorium:

   Link: https://github.com/AnaShiro/NoD1_2024

3. Opis programu opracowanego

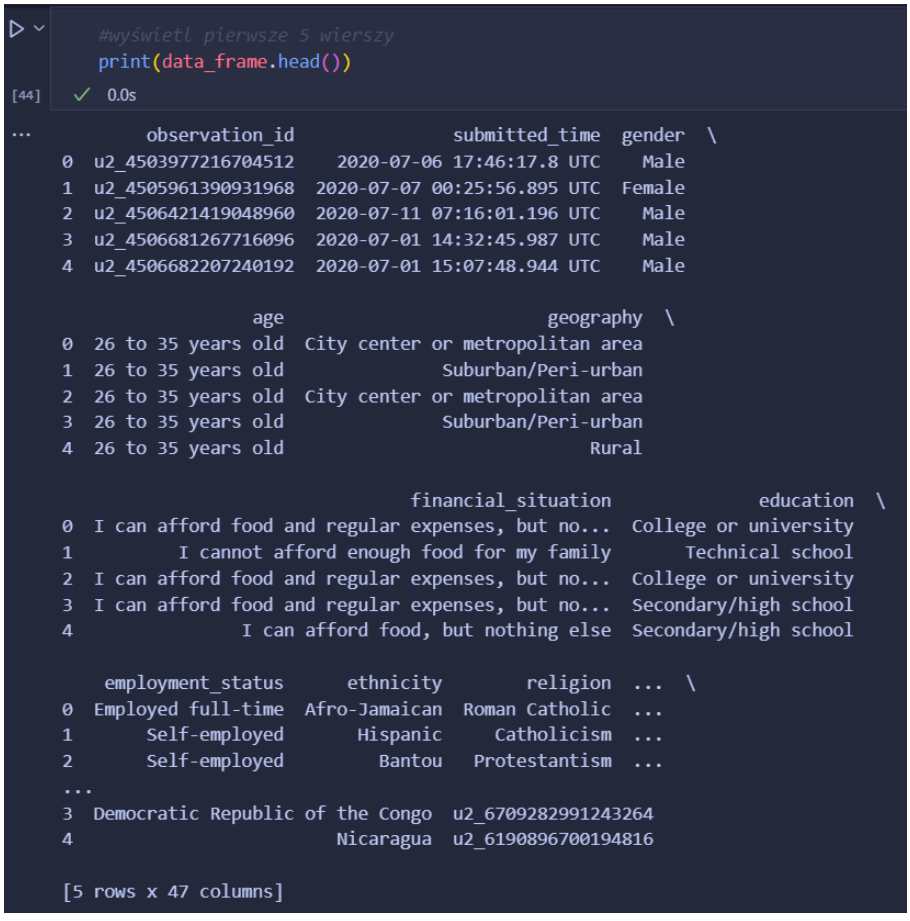
- Wczytywanie danych i wyświetlanie podstawowych informacji
  - Wczytaj dane z pliku



```
#importowanie biblioteki pandas
import pandas as pd
```

[42] ✓ 0.0s

  - Wyświetl pierwsze 5 wierszy



```
#wyświetl pierwsze 5 wierszy
print(data_frame.head())
```

[44] ✓ 0.0s

```
...          observation_id              submitted_time  gender  \
0  u2_4503977216704512    2020-07-06 17:46:17.8 UTC    Male
1  u2_4505961390931968  2020-07-07 00:25:56.895 UTC  Female
2  u2_4506421419048960  2020-07-11 07:16:01.196 UTC    Male
3  u2_4506681267716096  2020-07-01 14:32:45.987 UTC    Male
4  u2_4506682207240192  2020-07-01 15:07:48.944 UTC    Male

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

                                financial_situation             education  \
0  I can afford food and regular expenses, but no...  College or university
1         I cannot afford enough food for my family       Technical school
2  I can afford food and regular expenses, but no...  College or university
3  I can afford food and regular expenses, but no...  Secondary/high school
4               I can afford food, but nothing else  Secondary/high school

     employment_status      ethnicity        religion  ...  \
0  Employed full-time  Afro-Jamaican  Roman Catholic  ...
1       Self-employed       Hispanic     Catholicism  ...
2       Self-employed         Bantou   Protestantism  ...
...
3  Democratic Republic of the Congo  u2_6709282991243264
4                         Nicaragua  u2_6190896700194816

[5 rows x 47 columns]
```

- Sprawdź podstawowe informacje o danych

```
#sprawdź podstawowe informacje o danych
print(data_frame.info())
```

[45] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7228 entries, 0 to 7227
Data columns (total 47 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   observation_id              7228 non-null   object
 1   submitted_time              7228 non-null   object
 2   gender                      7228 non-null   object
 3   age                         7228 non-null   object
 4   geography                   7228 non-null   object
 5   financial_situation         7228 non-null   object
 6   education                   7228 non-null   object
 7   employment_status           7228 non-null   object
 8   ethnicity                   7228 non-null   object
 9   religion                    7228 non-null   object
 10  u2_hh                       7227 non-null   object
 11  u2_child_count              7228 non-null   int64
 12  u2_child1_age_months        7228 non-null   float64
 13  u2_child1_sex               7228 non-null   object
 14  u2_child2_age_months        2756 non-null   float64
 15  u2_child2_sex               2756 non-null   object
 16  u2_child3_age_months        922 non-null    float64
 17  u2_child3_sex               922 non-null    object
 18  u2_vaccine_card             7228 non-null   object
 19  u2_pre_vaccines             7228 non-null   object
...
 46  user_id                     7228 non-null   object
dtypes: float64(5), int64(1), object(41)
memory usage: 2.6+ MB
None
```

- Wyświetl podstawowe statystyki opisowe

```
# Wywietl podstawowe statystyki opisowe
print(data_frame.describe())
```

[59] ✓ 0.0s

```
       u2_child_count  u2_child1_age_months  u2_child2_age_months  \
count     7228.000000           7228.000000           2756.000000
mean         1.508854              7.646526              7.163831
std          0.710711              6.993561              6.906221
min          1.000000              0.000000              0.000000
25%          1.000000              2.000000              2.000000
50%          1.000000              5.000000              5.000000
75%          2.000000             12.000000             10.000000
max          3.000000             25.000000             25.000000

       u2_child3_age_months  u2_pre_vaccine_count  u2_post_vaccine_count  \
count            922.000000          5.149000e+03           4.451000e+03
mean               6.629826          5.839653e+05           3.009780e+06
std                6.477690          2.273222e+07           1.514366e+08
min                0.000000          0.000000e+00           0.000000e+00
25%                2.000000          1.000000e+00           1.000000e+00
50%                5.000000          2.000000e+00           2.000000e+00
75%                9.000000          3.000000e+00           3.000000e+00
max               25.000000          1.124894e+09           9.983722e+09
```

- Obliczanie podstawowych statystyk
  - Oblicz średnią

```
#Oblicz średni dla kolumny
mean_child_count = data_frame["u2_child_count"].mean()
print(f"Średnia liczba dzieci: {mean_child_count}")
```

[47] ✓ 0.0s

```
Średnia liczba dzieci: 1.5088544548976204
```

  - Oblicz medianę

```
#Oblicz median dla kolumny
median_child_count = data_frame["u2_child_count"].median()
print(f"Mediana liczby dzieci: {median_child_count}")
```

[48] ✓ 0.0s

```
Mediana liczby dzieci: 1.0
```

  - Oblicz odchylenie standardowe

```
#Oblicz odchylenie standardowe dla kolumny
std_child_count = data_frame["u2_child_count"].std()
print(f"Odchylenie standardowe liczby dzieci: {std_child_count}")
```

[49] ✓ 0.0s

```
Odchylenie standardowe liczby dzieci: 0.7107112373052482
```

- Identyfikacja i obsługa brakujących danych
  - Sprawdź brakujące wartości

```
#Sprawdź brakujące wartości
missing_values = data_frame.isnull().sum()
print("Brakujce wartoci w kadej kolumnie:")
print(missing_values)
```

[50] ✓ 0.0s

```
Brakujce wartoci w kadej kolumnie:
observation_id                0
submitted_time                0
gender                        0
age                           0
geography                     0
financial_situation           0
education                     0
employment_status             0
ethnicity                     0
religion                      0
u2_hh                         1
u2_child_count                0
u2_child1_age_months          0
u2_child1_sex                 0
u2_child2_age_months       4472
u2_child2_sex              4472
u2_child3_age_months       6306
```

  - Uzupełnij brakujące wartości średnią

```
#Uzupełnij brakujce wartoci redni w kolumnie liczba dzieci
data_frame["u2_child_count"].fillna(data_frame["u2_child_count"].mean(), inplace=True)
```

[60] ✓ 0.0s

```
C:\Users\Szymon\AppData\Local\Temp\ipykernel_2632\2284081746.py:2: FutureWarning: A value is
The behavior will change in pandas 3.0. This inplace method will never work because the inte

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: va

  data_frame["u2_child_count"].fillna(data_frame["u2_child_count"].mean(), inplace=True)
```

  - Usuń wiersze w których brakuje danych

```
#Usuń wiersze, gdzie brakuje danych w kolumnie
data_frame.dropna(subset=["u2_child_count"], inplace=True)
```

[52] ✓ 0.0s

- Wykrywanie wartości odstających
  - Oblicz IQR

```
#Oblicz IQR
Q1 = data_frame["u2_child_count"].quantile(0.25)
Q3 = data_frame["u2_child_count"].quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

[62] ✓ 0.0s

```
1.0
```

  - Zidentyfikuj wartości odstające

```
#Zidentyfikuj wartoci odstajce
outliers = data_frame[(data_frame["u2_child_count"] < (Q1 - 1.5 * IQR)) | (data_frame["u2_child_count"] > (Q3 + 1.5 * IQR))]
print("Wartoci odstajce :")
print(outliers)
```

[54] ✓ 0.0s

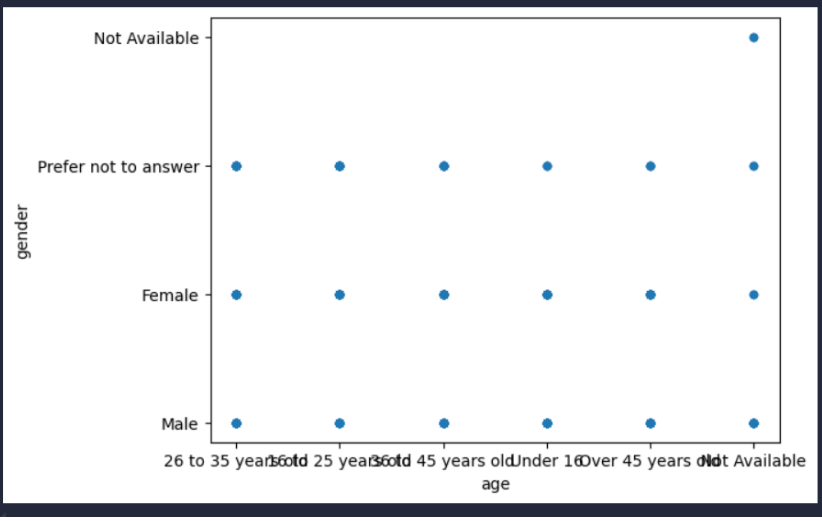
```
Wartoci odstajce :
Empty DataFrame
Columns: [observation_id, submitted_time, gender, age, geography, financial_situation, education, employment_status, ethnicity, reli
Index: []

[0 rows x 47 columns]
```

- Analiza zależności między kolumnami
  - Wykonaj wykres rozrzutu

```
#Wykonaj wykres rozrzutu
data_frame.plot.scatter(x="age", y="gender")
```

[55] ✓ 0.5s

```
<Axes: xlabel='age', ylabel='gender'>
```



- Przekształcenie danych
  - Dodaj nową kolumnę

```
#Dodaj nowa kolumne
data_frame["byleco"] = data_frame["u2_child_count"] / data_frame["u2_child_count"]
```

[56] ✓ 0.0s

- Grupuj dane według kolumny

```
#Grupuj dane wedug kolumny 'region' i oblicz średnią
grouped = data_frame.groupby("gender")["u2_child_count"].mean()
print("Coś:")
print(grouped)
```

[57] ✓ 0.0s

```
Coś:
gender
Female                  1.415692
Male                    1.542283
Not Available           1.000000
Prefer not to answer    1.740000
Name: u2_child_count, dtype: float64
```

- Posortuj dane według kolumny

```
# Posortuj dane wedug kolumny
df_sorted = data_frame.sort_values(by="age", ascending=False)
print("Dane posortowane według wieku:")
print(df_sorted.head())
```

[58] ✓ 0.0s

```
Dane posortowane według wieku:
          observation_id              submitted_time  gender       age  \
1290  u2_4925422526791680  2020-07-12 01:51:11.179 UTC  Female  Under 16
2513  u2_5293163465146368  2020-06-30 18:43:51.855 UTC    Male  Under 16
922   u2_4806821165662208   2020-07-01 17:56:20.74 UTC  Female  Under 16
5389  u2_6185680486268928  2020-07-03 12:48:16.841 UTC    Male  Under 16
5392  u2_6187233382236160  2020-07-02 02:24:18.759 UTC    Male  Under 16

               geography                                 financial_situation  \
1290               Rural  I can afford food and regular expenses, but no...
2513               Rural          I cannot afford enough food for my family
922                Rural               I can afford food, but nothing else
5389               Rural          I cannot afford enough food for my family
5392  Suburban/Peri-urban          I cannot afford enough food for my family

                education   employment_status                ethnicity  \
1290  Secondary/high school             Student                    Thai
2513  Secondary/high school  Employed full-time  Black or African American
922   Secondary/high school             Student                  Pashtun
5389  Secondary/high school             Student                    Khmer
5392  Secondary/high school             Student                    Khmer

          religion  ... u2_post_provider_need  \
1290       Buddhism  ...                   No
2513     Catholicism                         No
```

4. Wnioski

Zaczęliśmy od wczytania danych z pliku CSV i wyświetlenia podstawowych informacji o zbiorze danych. Następnie obliczyliśmy podstawowe statystyki opisowe dla wybranych kolumn, aby zrozumieć rozkład danych. Brakujące dane mogły wpłynąć na jakość analizy, więc musieliśmy je zidentyfikować i odpowiednio obsłużyć. Wartości odstające mogły zaburzać wyniki analizy, dlatego musieliśmy je zidentyfikować. Zbadaliśmy zależności między różnymi kolumnami poprzez obliczenie współczynników korelacji. Na koniec przekształciliśmy dane, tworząc nowe kolumny, grupując dane i sortując je.

Wykonując te zadania, nauczyliśmy się podstawowych i zaawansowanych technik manipulacji danymi w Pandas. Zrozumieliśmy, jak efektywnie wczytywać, analizować i przekształcać dane, co jest niezbędną umiejętnością w analizie danych i nauce o danych.