

## 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 11	Szymon Nycz Informatyka II stopień, niestacjonarne, 1 semestr, gr.1b
--	---

### 1. Polecenie:

Premise General Population COVID-19 Health Services Disruption Survey 2020

<http://ghdx.healthdata.org/record/ihme-data/premise-general-population-covid-19-health-services-disruption-survey-2020>

### 2. Link do repozytorium:

Link: [https://github.com/Maciek332/Semestr\\_1\\_Nycz/tree/master/NoD/Lab\\_3](https://github.com/Maciek332/Semestr_1_Nycz/tree/master/NoD/Lab_3)

### 3. Opis programu opracowanego

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

```
▶ #tworzenie ramki danych ze słownika
data = pd.read_csv('IHME_PREM_GEN_POP_2020_Y2021M10011.csv', encoding='utf-8')

data_frame = pd.DataFrame(data)
wyświetlenie ramki danych
data_frame

[37] ✓ 0.6s
```

	observation_id	submitted_time	gender	age	geography	financial_situation	education	employment_status	ethnicity
0	gp_4503617949401088	2020-07-07 14:48:29.83 UTC	Male	Under 16	Suburban/Peri-urban	I can afford food and regular expenses, but no...	Secondary/high school	Employed full-time	Ankole
1	gp_4503631639609344	2020-07-09 13:22:37.107 UTC	Female	26 to 35 years old	City center or metropolitan area	I cannot afford enough food for my family	College or university	Unemployed	Mestizo
2	gp_4503700758593536	2020-07-04 18:53:36.471 UTC	Male	36 to 45 years old	City center or metropolitan area	I can comfortably afford food, clothes, and fu...	Primary school	Employed full-time	Non-hispanic White
3	gp_4503737805832192	2020-07-12 17:58:20.798 UTC	Male	26 to 35 years old	Rural	I can afford food and regular expenses, and bu...	Technical school	Student	Mestizo
4	gp_4503819343101952	2020-07-06 00:20:22.983 UTC	Male	26 to 35 years old	Suburban/Peri-urban	I can afford food and regular expenses, and bu...	Technical school	Employed full-time	Mestizo
...	...	...	...	...	...	...	...	...	...
52485	gp_4765406044572606	2020-07-02 03:27:53.604	Female	36 to 45 years old	Suburban/Peri-urban	I can afford food, clothes, and fu...	Secondary/high school	Employed part-time	Not

- Wyświetl pierwsze 5 wierszy

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

[38] ✓ 0.0s
```

	observation_id	submitted_time	gender	age	geography	financial_situation	education	employment_status	ethnicity	religion
0	gp_4503617949401088	2020-07-07 14:48:29.83 UTC	Male	Under 16	Suburban/Peri-urban	I can afford food and regular expenses, but no...	Secondary/high school	Employed full-time	Ankole	Christianity
1	gp_4503631639609344	2020-07-09 13:22:37.107 UTC	Female	26 to 35 years old	City center or metropolitan area	I cannot afford enough food for my family	College or university	Unemployed	Mestizo	Catholicism
2	gp_4503700758593536	2020-07-04 18:53:36.471 UTC	Male	36 to 45 years old	City center or metropolitan area	I can comfortably afford food, clothes, and fu...	Primary school	Employed full-time	Non-hispanic White	Agnosticism
3	gp_4503737805832192	2020-07-12 17:58:20.798 UTC	Male	26 to 35 years old	Rural	I can afford food and regular expenses, and bu...	Technical school	Student	Mestizo	...
4	gp_4503819343101952	2020-07-06 00:20:22.983 UTC	Male	26 to 35 years old	Suburban/Peri-urban	I can afford food and regular expenses, and bu...	Technical school	Employed full-time	Mestizo	...

[5 rows x 48 columns]

- Sprawdź podstawowe informacje o danych

```

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

```

[39] ✓ 0.0s

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52490 entries, 0 to 52489
Data columns (total 48 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   observation_id                       52490 non-null  object
 1   submitted_time                      52490 non-null  object
 2   gender                             52469 non-null  object
 3   age                                 52490 non-null  object
 4   geography                          52490 non-null  object
 5   financial_situation                 52490 non-null  object
 6   education                          52490 non-null  object
 7   employment_status                  52490 non-null  object
 8   ethnicity                          52490 non-null  object
 9   religion                           52490 non-null  object
10   gp_hh                              52478 non-null  object
11   gp_pre_provider_need                52490 non-null  object
12   gp_pre_provider_condition           21777 non-null  object
13   gp_pre_provider_condition_other     2982 non-null  object
14   gp_pre_provider_visit               21777 non-null  object
15   gp_pre_provider_where               9972 non-null  object
16   gp_pre_provider_where_other         803 non-null   object
17   gp_pre_provider_num_visit           19037 non-null  object
18   gp_pre_provider_why                 8363 non-null  object
19   gp_pre_provider_why_other           398 non-null   object
...
47   user_id                             52490 non-null  object
dtypes: float64(3), object(45)
memory usage: 19.2+ MB
None

```

- Wyświetl podstawowe statystyki opisowe

```

# Wyświetl podstawowe statystyki opisowe
print(data_frame.describe())

```

[40] ✓ 0.0s

```

...

```

	gp_pre_income	gp_post_income	weight
count	5.249000e+04	5.249000e+04	52490.000000
mean	1.905125e+56	1.905125e+59	1.795820
std	4.364774e+58	4.364774e+61	0.384507
min	0.000000e+00	0.000000e+00	1.000000
25%	1.500000e+02	7.000000e+01	1.544666
50%	3.425000e+03	2.000000e+03	1.730488
75%	2.500000e+04	2.000000e+04	1.908871
max	1.000000e+61	1.000000e+64	6.411420

- Obliczanie podstawowych statystyk

- Oblicz średnią

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

[41] ✓ 0.0s

... Średnia liczba dzieci: 1.9051247856734615e+56

- Oblicz medianę

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

[42] ✓ 0.0s

... Mediana liczby dzieci: 3425.0

- Oblicz odchylenie standardowe

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

[43] ✓ 0.0s

... Odchylenie standardowe liczby dzieci: 4.364773517232549e+58

- 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)
```

[44] ✓ 0.0s

... Brakujce wartoci w kadej kolumnie:

observation_id	0
submitted_time	0
gender	21
age	0
geography	0
financial_situation	0
education	0
employment_status	0
ethnicity	0
religion	0
gp_hh	12
gp_pre_provider_need	0
gp_pre_provider_condition	30713
gp_pre_provider_condition_other	49508
gp_pre_provider_visit	30713
gp_pre_provider_where	42518
gp_pre_provider_where_other	51687
gp_pre_provider_num_visit	33453
gp_pre_provider_why	44127
gp_pre_provider_why_other	52092

- Uzupełnij brakujące wartości średnią

```
#Uzupełnij brakujące wartości średnią w kolumnie liczba dzieci
data_frame["gp_pre_income"].fillna(data_frame["gp_pre_income"].mean(), inplace=True)

[45] ✓ 0.0s

... C:\Users\Szymon\AppData\Local\Temp\ipykernel_20160\1050778849.py:2: FutureWarning: A value is
The behavior will change in pandas 3.0. This inplace method will never work because the inter

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

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

- Usuń wiersze w których brakuje danych

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

[46] ✓ 0.0s
```

- Wykrywanie wartości odstających

- Oblicz IQR

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

[47] ✓ 0.0s

... 24850.0
```

- Zidentyfikuj wartości odstające

```
#Zidentyfikuj wartości odstające
outliers = data_frame[(data_frame["gp_pre_income"] < (Q1 - 1.5 * IQR)) | (data_frame["gp_pre_income"] > (Q3 + 1.5 * IQR))]
print("Wartosci odstajace :")
print(outliers)

[48] ✓ 0.0s

... Wartosci odstajace :
   observation_id  submitted_time  gender \
4  gp_4503819343101952  2020-07-06 00:20:22.983 UTC  Male
5  gp_4503868601008128  2020-07-11 17:45:09.631 UTC  Male
18 gp_4504344939724800  2020-07-04 17:45:35.164 UTC  Male
19 gp_4504387688071168  2020-07-12 16:14:12.613 UTC  Male
26 gp_4504536465276928  2020-07-03 18:11:18.738 UTC  Female
...      ...      ...      ...
52466 gp_6754360325570560  2020-06-30 20:37:40.345 UTC  Female
52468 gp_6754417841012736  2020-07-03 20:25:54.494 UTC  Male
52474 gp_6754574808645632  2020-07-10 18:04:16.896 UTC  Male
52482 gp_6754988065030144  2020-07-03 03:50:43.956 UTC  Female
52486 gp_6755213279232000  2020-07-02 07:07:44.269 UTC  Male

   age  geography \
4  26 to 35 years old  Suburban/Peri-urban
5  26 to 35 years old  Suburban/Peri-urban
18  26 to 35 years old  City center or metropolitan area
19  16 to 25 years old  City center or metropolitan area
26  26 to 35 years old  Suburban/Peri-urban
...      ...      ...
52466  16 to 25 years old  City center or metropolitan area
```

- Wykonaj wykres rozrzutu



- Dodaj nową kolumnę

```
#Dodaj nową kolumnę
data_frame["nowa"] = data_frame["gp_pre_income"] / 2
```

- Grupuj dane według kolumny

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

[51] ✓ 0.0s

...

```
Coś:
gender
Female                2.366555e+08
Male                  2.867301e+56
Prefer not to respond  7.089249e+12
Name: gp_pre_income, dtype: float64
```

- Posortuj dane według kolumny

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

[52] ✓ 0.1s

... Dane posortowane według wieku:

	observation_id	submitted_time	gender	age
0	gp_4503617949401088	2020-07-07 14:48:29.83 UTC	Male	Under 16
39805	gp_6212113459838976	2020-07-01 16:26:44.48 UTC	Male	Under 16
39774	gp_6210954120658944	2020-07-18 20:27:53.831 UTC	Female	Under 16
7745	gp_4835939500425216	2020-07-03 07:36:54.128 UTC	Male	Under 16
39769	gp_6210720447594496	2020-07-07 19:59:10.754 UTC	Male	Under 16

  

	geography	financial_situation
0	Suburban/Peri-urban	I can afford food and regular expenses, but no...
39805	Rural	I can afford food, but nothing else
39774	Rural	I can afford food and regular expenses, and bu...
7745	Rural	I can afford food and regular expenses, but no...
39769	Rural	I cannot afford enough food for my family

  

	education	employment_status	ethnicity	religion
0	Secondary/high school	Employed full-time	Ankole	Christianity
39805	Secondary/high school	Student	Sarakole	Muslim
39774	Technical school	Employed part-time	Igbo	Christianity
7745	Secondary/high school	Self-employed	Bisaya	Other
39769	Primary school	Self-employed	Malinke	Muslim

  

	gp_post_labor_force	gp_pre_labor_force	gp_unemployment_why
0	No	No	NaN
39805	No	No	NaN

## 4. Wnioski

Rozpoczęliśmy od załadowania danych z pliku CSV i zaprezentowania podstawowych informacji o zbiorze danych. Następnie obliczyliśmy kluczowe statystyki opisowe dla wybranych kolumn, aby lepiej zrozumieć rozkład danych. Brakujące dane mogły wpłynąć na jakość analizy, dlatego musieliśmy je zidentyfikować i odpowiednio się nimi zająć. Wartości odstające mogły zniekształcać wyniki, więc również je wykryliśmy. Analizując zależności między różnymi kolumnami, obliczyliśmy współczynniki korelacji. Na końcu dokonaliśmy transformacji danych, tworząc nowe kolumny, grupując je i sortując.

Podczas realizacji tych kroków opanowaliśmy zarówno podstawowe, jak i bardziej zaawansowane techniki manipulacji danymi w Pandas. Zrozumieliśmy, jak efektywnie ładować, analizować i przekształcać dane, co stanowi kluczową umiejętność w pracy z danymi oraz analizie danych.