SPRAWOZDANIE

Zajęcia: Nauka o danych I
Prowadzący: prof. dr hab. Vasyl Martsenyuk

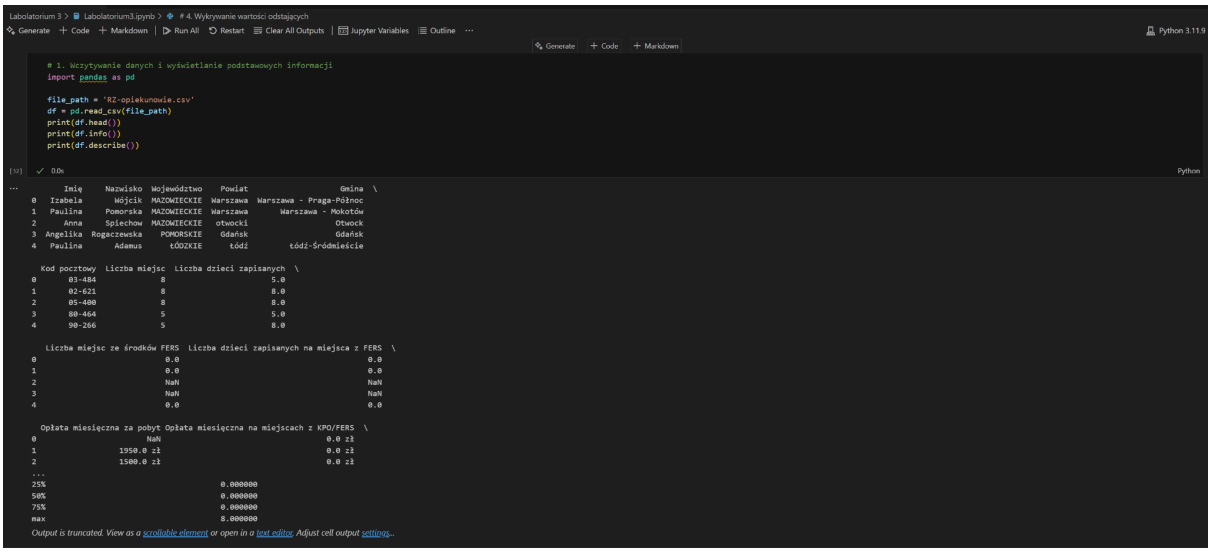
Laboratorium Nr 3 Data 19.10.2024 Temat: Wykorzystanie pakietu Pandas do manipulacji i przetwarzania danych w Pythonie	Bartosz Bieniek Informatyka II stopień, stacjonarne, 1 semestr, gr.A

1. Polecenie

Zadanie wymagało wykonania następujących kroków:
1. Wczytywanie danych i wyświetlanie podstawowych informacji
2. Obliczanie podstawowych statystyk
3. Identyfikacja i obsługa brakujących danych
4. Wykrywanie wartości odstających
5. Analiza zależności między kolumnami
6. Przekształcanie danych

2. Opis programu opracowanego:

https://github.com/mindgoner/Studia/tree/master/Nauka%20o%20Danych/Laboratorium%203



Rys. 1. Fragment kodu i wynik jego wykonania

Podano dane w formacie CSV, które zostały wczytane do DataFrame. Wyświetlono pierwsze kilka wierszy, informacje o strukturze danych oraz podstawowe statystyki opisowe.

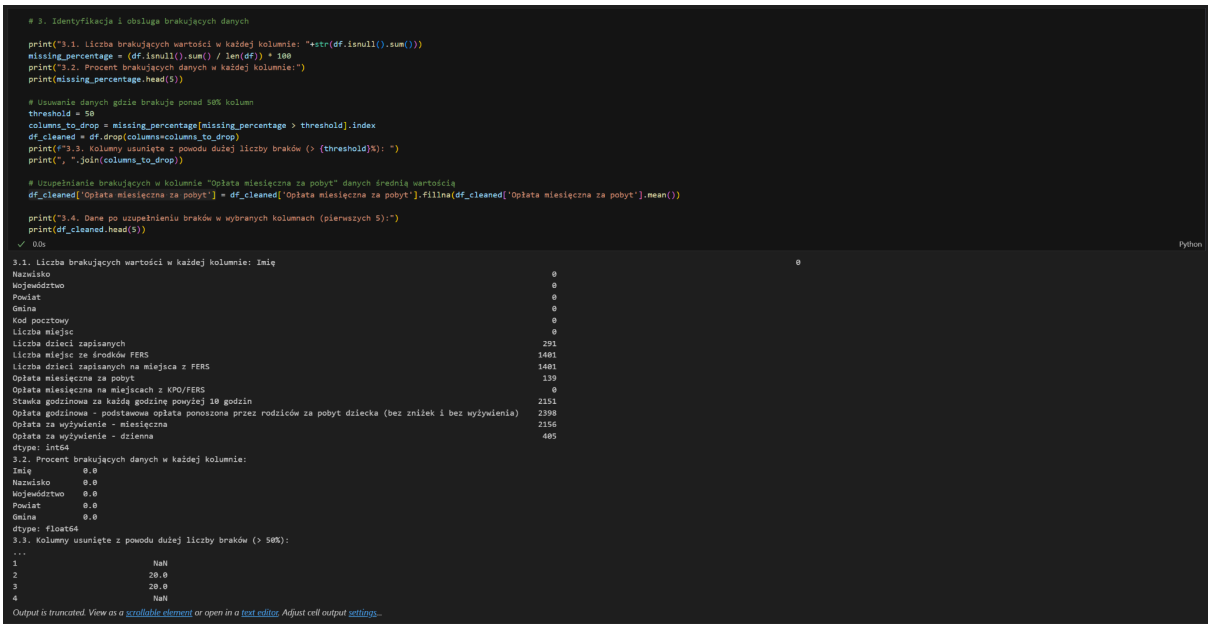
```
# 2. Obliczanie podstawowych statystyk
print("2.1. Średnia liczba miejsc: "+str(df['Liczba miejsc'].mean()))
print("2.2. Średnia liczba zapisanych dzieci: "+str(df['Liczba dzieci zapisanych'].mean()))
print("2.3. Suma miejsc ze środków FERS: "+str(df['Liczba miejsc ze środków FERS'].sum()))
df['Opłata miesięczna za pobyt'] = pd.to_numeric(df['Opłata miesięczna za pobyt'].str.replace(' zł', ''), errors='coerce')
print("2.4. Maksymalna opłata miesięczna za pobyt: "+str(df['Opłata miesięczna za pobyt'].max()))
df['Opłata za wyżywienie - dzienna'] = pd.to_numeric(df['Opłata za wyżywienie - dzienna'].str.replace(' zł', ''), errors='coerce')
print("2.5. Minimalna dzienna opłata za wyżywienie: "+str(df['Opłata za wyżywienie - dzienna'].min()))
```

✓ 0.0s

```
1. Średnia liczba miejsc: 6.238877755511022
2. Średnia liczba zapisanych dzieci: 4.544918330308853
3. Suma miejsc ze środków FERS: 515.0
4. Maksymalna opłata miesięczna za pobyt: 3700.0
5. Minimalna dzienna opłata za wyżywienie: 0.0
```

Rys. 2. Fragment kodu i wynik jego wykonania

Na podstawie wybranych kolumn obliczono średnie, sumy i minimalne/maksymalne wartości. Statystyki te pomogły zrozumieć rozkład danych w zbiorze.



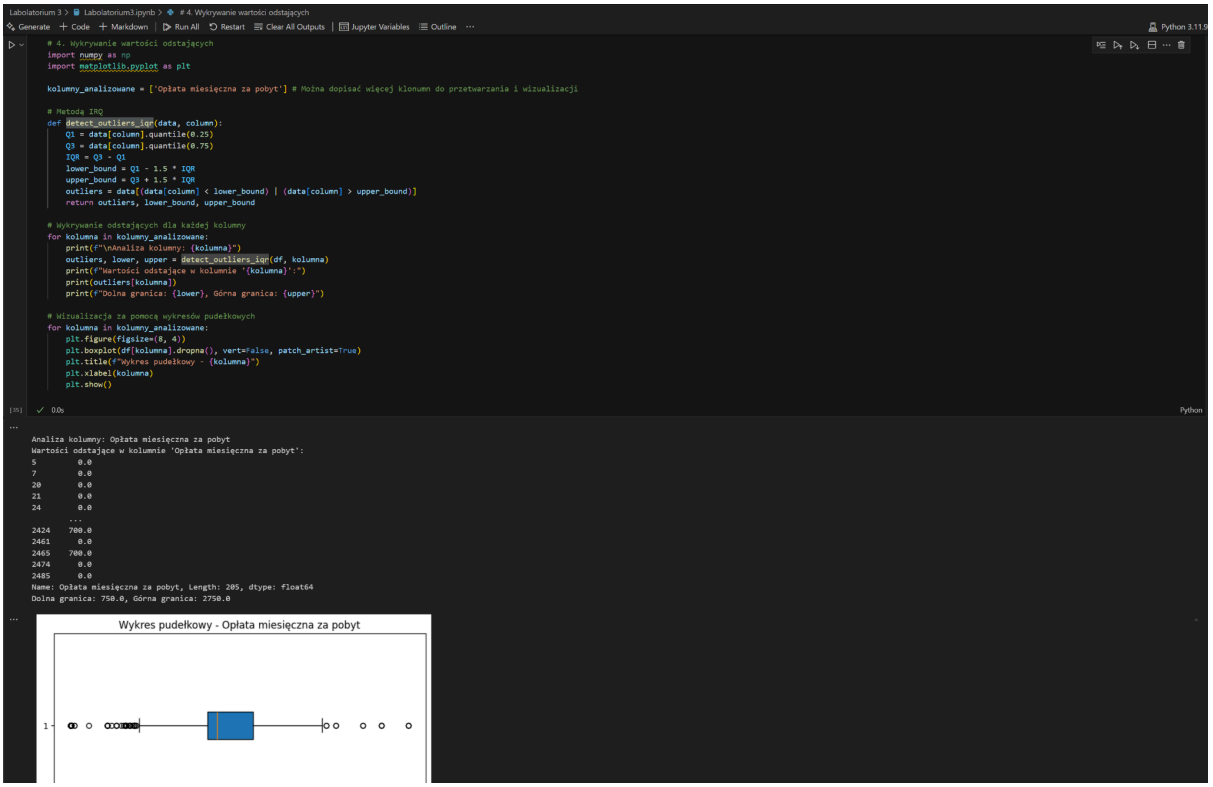
```
# 3. Identyfikacja i obsługa brakujących danych

print("3.1. Liczba brakujących wartości w każdej kolumnie: "+str(df.isnull().sum()))
missing_percentage = (df.isnull().sum() / len(df)) * 100
print("3.2. Procent brakujących danych w każdej kolumnie:")
print(missing_percentage.head(5))

# Usuwanie danych gdzie brakuje ponad 50% kolumn
threshold = 50
columns_to_drop = missing_percentage[missing_percentage > threshold].index
df_cleaned = df.drop(columns=columns_to_drop)
print(f"3.3. Kolumny usunięte z powodu dużej liczby braków (> {threshold}%): ")
print(", ".join(columns_to_drop))

# Uzupełnianie brakujących w kolumnie "Opłata miesięczna za pobyt" danych średnią wartością
df_cleaned['Opłata miesięczna za pobyt'] = df_cleaned['Opłata miesięczna za pobyt'].fillna(df_cleaned['Opłata miesięczna za pobyt'].mean())

print("3.4. Dane po uzupełnieniu braków w wybranych kolumnach (pierwszych 5):")
print(df_cleaned.head(5))
```

✓ 0.0s Python

```
3.1. Liczba brakujących wartości w każdej kolumnie: Imię                                                  0
Nazwisko                                                  0
Województwo                                               0
Powiat                                                    0
Gmina                                                     0
Kod pocztowy                                              0
Liczba miejsc                                             0
Liczba dzieci zapisanych                                291
Liczba miejsc ze środków FERS                          1401
Liczba dzieci zapisanych na miejsca z FERS             1401
Opłata miesięczna za pobyt                              139
Opłata miesięczna na miejscach z KPO/FERS                 0
Stawka godzinowa za każdą godzinę powyżej 10 godzin    2151
Opłata godzinowa - podstawowa opłata ponoszona przez rodziców za pobyt dziecka (bez zniżek i bez wyżywienia)    2398
Opłata za wyżywienie - miesięczna                      2156
Opłata za wyżywienie - dzienna                          405
dtype: int64
3.2. Procent brakujących danych w każdej kolumnie:
Imię            0.0
Nazwisko        0.0
Województwo     0.0
Powiat          0.0
Gmina           0.0
dtype: float64
3.3. Kolumny usunięte z powodu dużej liczby braków (> 50%):
...
1                         NaN
2                        20.0
3                        20.0
4                         NaN
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Rys. 3. Fragment kodu i wynik jego wykonania

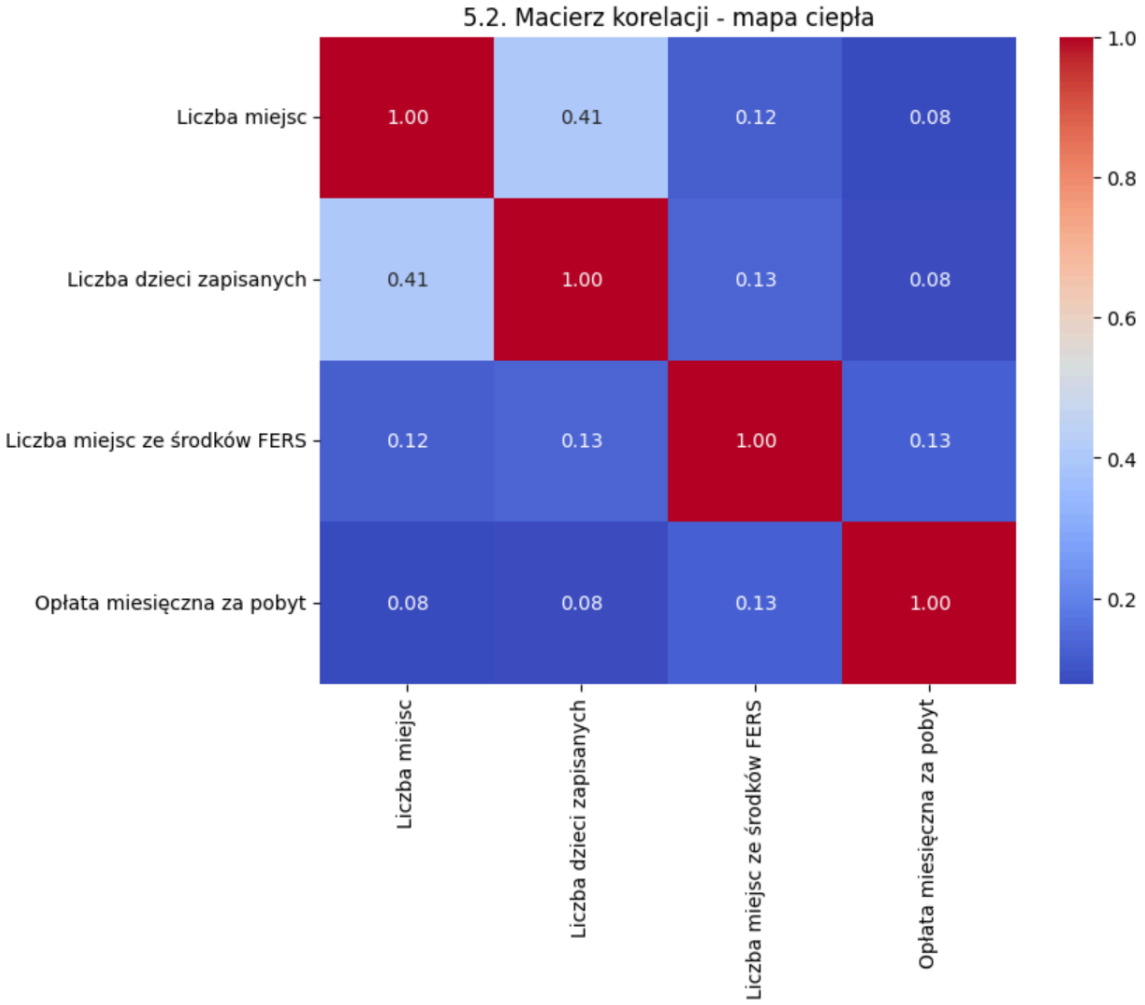
Zidentyfikowano kolumny zawierające brakujące wartości oraz obliczono ich procentowy udział. Użyto różnych metod uzupełniania braków w danych, usunięcia kolumn z dużą liczbą braków, co pozwoliło przygotować dane do dalszej analizy.



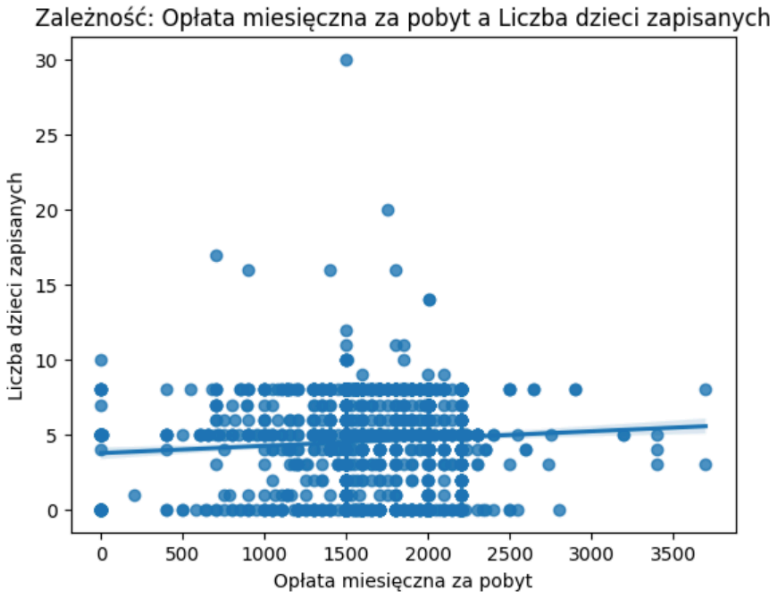
Rys. 4. Fragment kodu i wynik jego wykonania

Za pomocą metody IQR wykryto wartości odstające w danych. Wizualizacja tych wartości za pomocą wykresów pudełkowych ułatwiła zrozumienie ich rozmieszczenia.

Rys. 5. Fragment kodu

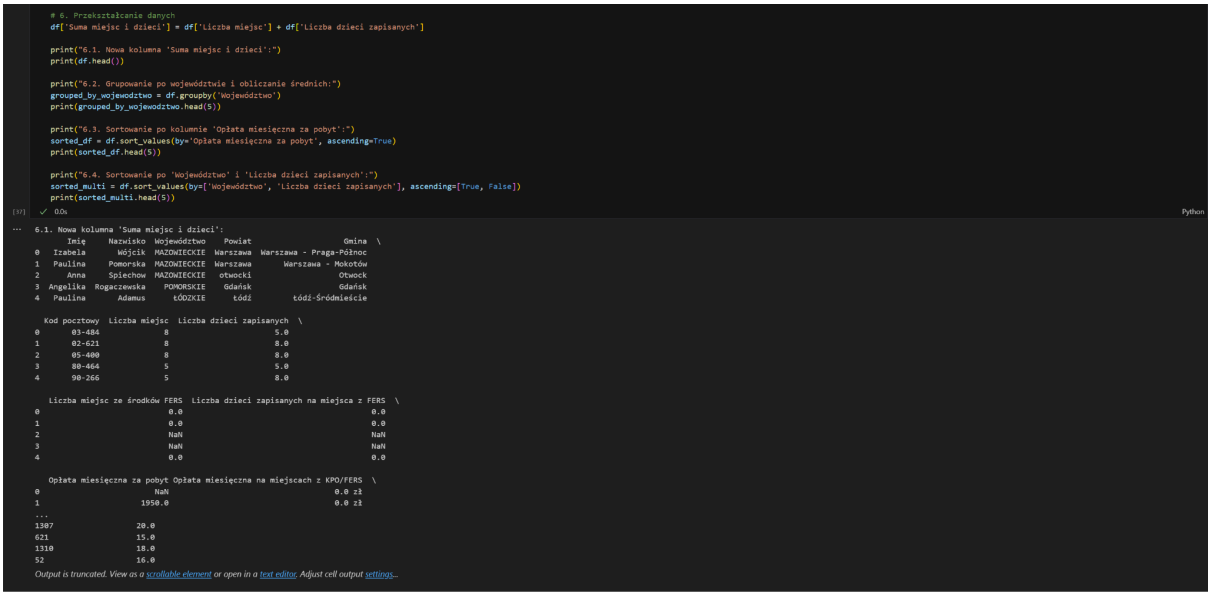


Rys. 6. Macierz korelacji - mapa ciepła



Rys. 7. Macierz korelacji - mapa ciepła

Obliczono korelacje między wybranymi kolumnami liczbowymi oraz przeprowadzono analizę wizualną tych zależności. Użyto wykresów punktowych i heatmap, które pomogły w zrozumieniu powiązań pomiędzy danymi.



```
# 6. Przekształcanie danych
df['Suma miejsc i dzieci'] = df['Liczba miejsc'] + df['Liczba dzieci zapisanych']

print("6.1. Nowa kolumna 'Suma miejsc i dzieci':")
print(df.head())

print("6.2. Grupowanie po województwie i obliczanie średnich:")
grouped_by_wojewodztwo = df.groupby('Województwo')
print(grouped_by_wojewodztwo.head(5))

print("6.3. Sortowanie po kolumnie 'Opłata miesięczna za pobyt':")
sorted_df = df.sort_values(by='Opłata miesięczna za pobyt', ascending=True)
print(sorted_df.head(5))

print("6.4. Sortowanie po 'Województwo' i 'Liczba dzieci zapisanych':")
sorted_multi = df.sort_values(by=['Województwo', 'Liczba dzieci zapisanych'], ascending=[True, False])
print(sorted_multi.head(5))
```

```
[37] ✓ 0.0s                                                                    Python

6.1. Nowa kolumna 'Suma miejsc i dzieci':
        Imię     Nazwisko  Województwo    Powiat                  Gmina  \
0   Izabela       Wójcik  MAZOWIECKIE  Warszawa  Warszawa - Praga-Północ
1   Paulina     Pomorska  MAZOWIECKIE  Warszawa       Warszawa - Mokotów
2      Anna     Spiechow  MAZOWIECKIE   otwocki                 Otwock
3  Angelika  Rogaczewska     POMORSKIE    Gdańsk                 Gdańsk
4   Paulina       Adamus      ŁÓDZKIE      łódź         łódź-Śródmieście

  Kod pocztowy  Liczba miejsc  Liczba dzieci zapisanych  \
0      03-484              8                       5.0
1      02-621              8                       8.0
2      05-400              8                       8.0
3      80-464              5                       5.0
4      90-266              5                       8.0

   Liczba miejsc ze środków FERS  Liczba dzieci zapisanych na miejsca z FERS  \
0                            0.0                                         0.0
1                            0.0                                         0.0
2                            NaN                                         NaN
3                            NaN                                         NaN
4                            0.0                                         0.0

   Opłata miesięczna za pobyt Opłata miesięczna na miejscach z KPO/FERS  \
0                         NaN                                    0.0 zł
1                      1950.0                                    0.0 zł
...
1307                   20.0
621                    15.0
1310                   18.0
52                     16.0
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Rys. 8. Fragment kodu i wynik jego wykonania

Stworzono nową kolumnę na podstawie dwóch istniejących. Następnie dane zostały posortowane według wybranych kolumn, a także grupowane po regionie (województwo), co pozwoliło na dalsze analizy.

3. Wnioski

W trakcie przetwarzania danych zauważono, że wiele kolumn zawiera brakujące wartości. Obsługa tych braków poprzez uzupełnianie, usuwanie kolumn czy konwersję pozwala przygotować dane do dalszej analizy.

Wartości odstające często wpływają na wyniki analiz, dlatego ich identyfikacja i wizualizacja za pomocą wykresów pudełkowych jest niezbędna do poprawnej interpretacji danych.

Przeprowadzenie analizy korelacji oraz wizualizacja tych zależności za pomocą mapy ciepła pozwala na zrozumienie, jak poszczególne cechy danych są ze sobą powiązane, co może prowadzić do bardziej trafnych decyzji analitycznych.