

# Laboratorium Pandas II NS

## Zadanie 1:

### Grupowanie i agregacja (groupby)

Masz ramkę danych przedstawiającą sprzedaż w różnych regionach:

```
data = {
    'Region': ['Północ', 'Południe', 'Wschód', 'Zachód', 'Północ',
              'Południe'],
    'Produkt': ['A', 'A', 'B', 'B', 'C', 'C'],
    'Sprzedaż': [150, 200, 130, 180, 220, 170]
}
```

Wykonaj:

- Oblicz sumaryczną sprzedaż dla każdego regionu (użyj `groupby` i `sum`).
- Oblicz średnią sprzedaż dla każdego produktu (użyj `groupby` i `mean`).
- Znajdź region z najwyższą łączną sprzedażą (użyj `groupby` oraz `idxmax`).

## Zadanie 2:

### Praca z brakującymi danymi

Masz ramkę danych zawierającą informacje o klientach:

```
data = {
    'Imię': ['Jan', 'Anna', 'Piotr', 'Kasia'],
    'Wiek': [23, np.nan, 22, 24],
    'Miasto': ['Warszawa', 'Kraków', np.nan, 'Gdańsk']
}
```

Wykonaj:

- Znajdź wszystkie brakujące wartości (użyj `isna`).
- Wypełnij brakujące wartości w kolumnie 'Wiek' średnią wieku (użyj `fillna`).
- Usuń wiersze, które mają brakujące wartości w kolumnie 'Miasto' (użyj `dropna`).

## Zadanie 3:

### Operacje na indeksach

Masz ramkę danych:

```
data = {
    'ID': [101, 102, 103, 104],
    'Produkt': ['Laptop', 'Telefon', 'Tablet', 'Monitor'],
    'Cena': [4000, 2500, 1500, 800]
}
df = pd.DataFrame(data)
```

Wykonaj:

- Ustaw kolumnę 'ID' jako indeks (użyj `set_index`).
- Odczytaj wiersz o indeksie 102 (użyj `loc`).
- Zresetuj indeks do domyślnego (użyj `reset_index`).

## Zadanie 4:

### Łączenie ramek danych (merge)

Masz dwie ramki danych:

```
df1 = pd.DataFrame({
    'ID': [1, 2, 3],
    'Imię': ['Jan', 'Anna', 'Piotr']
})

df2 = pd.DataFrame({
    'ID': [2, 3, 4],
    'Wiek': [25, 30, 22]
})
```

Wykonaj:

- Połącz te ramki danych na podstawie kolumny 'ID' (operacja wewnętrznego łączenia).
- Połącz je operacją zewnętrznego łączenia.
- Wyjaśnij różnicę między tymi dwoma operacjami.

## Zadanie 5:

### Przekształcanie danych (pivot, melt)

Masz ramkę danych:

```
data = {
    'Rok': [2020, 2020, 2021, 2021],
    'Produkt': ['A', 'B', 'A', 'B'],
}
```

```
'Sprzedaż': [100, 150, 120, 160]
}
df = pd.DataFrame(data)
```

Wykonaj:

- Użyj funkcji `pivot_table`, aby przedstawić dane w formie tabeli przestawnej, gdzie indeks to 'Rok', a kolumny to 'Produkt'.
- Przekształć ramkę danych do postaci długiej (użyj `melt`).

## Zadanie 6:

### Tworzenie nowych kolumn na podstawie warunków

Masz ramkę danych z wynikami studentów:

```
data = {
    'Student': ['Anna', 'Bartek', 'Celina', 'Daniel'],
    'Matematyka': [45, 80, 75, 60],
    'Fizyka': [70, 55, 80, 90]
}
df = pd.DataFrame(data)
```

Wykonaj:

- Dodaj nową kolumnę 'Zaliczenie', która przyjmuje wartość 'Tak', jeśli średnia ocen wynosi co najmniej 60, w przeciwnym razie 'Nie' (użyj `apply`).
- Użyj funkcji `apply`, aby dodać kolumnę 'Ocena' według przedziałów: 0-50: 'nieodstateczny', 51-70: 'dostateczny', 71-85: 'dobry', powyżej 85: 'bardzo dobry'.

## Zadanie 7:

### Analiza statystyczna

Masz ramkę danych z wynikami sprzedaży:

```
data = {
    'Miesiąc': ['Styczeń', 'Styczeń', 'Luty', 'Luty', 'Marzec',
    'Marzec'],
    'Produkt': ['A', 'B', 'A', 'B', 'A', 'B'],
    'Sprzedaż': [120, 80, 200, 150, 90, 300]
}
df = pd.DataFrame(data)
```

Wykonaj:

- Oblicz medianę sprzedaży dla każdego miesiąca (użyj `groupby` i `median`).
- Oblicz maksymalną i minimalną sprzedaż dla każdego produktu (użyj `groupby`, `max` oraz `min`).
- Oblicz kwartyłowy rozkład sprzedaży (użyj `describe`).