Zadanie: Analiza otwartych danych z użyciem NumPy i pandas

Cel

Poznać praktyczny workflow analizy danych: import z otwartego źródła (CSV), czyszczenie, transformacje wektorowe (NumPy), agregacje i łączenie tabel (pandas), a na końcu eksport wyników i krótkie wnioski.

Dane

Wybierz **jedno publiczne źródło danych (CSV)**, np.:

- Our World in Data (OWID) – dowolny temat (energia, zdrowie, edukacja),
- Bank Danych Lokalnych GUS – eksport CSV dla wybranego wskaźnika,
- dowolny portal „Open Data" miasta/państwa (transport, środowisko, finanse),
- alternatywnie: jakiekolwiek repozytorium z plikami **CSV ≥ 10 000 wierszy**.

Jeśli pracujesz offline, możesz użyć własnego pliku CSV o podobnej wielkości (np. logi sprzedaży, zamówienia, pomiary, zdarzenia).

Zadania (krok po kroku)

1. **Import danych**
    - Wczytaj CSV do pandas.DataFrame.
    - Pokaż podstawowe info: liczba wierszy/kolumn, typy (.info(), .describe()).
2. **Czyszczenie i typy**
    - Ustandaryzuj nagłówki (małe litery, _ zamiast spacji).
    - Przekonwertuj kolumny dat/czasu na datetime (jeśli istnieją).
    - Uzupełnij lub usuń brakujące wartości: wyjaśnij wybór (np. medianą, forward fill).
3. **NumPy – transformacje wektorowe**
    - Dodaj **min. 2 kolumny wyliczane** z użyciem NumPy (np. skalowanie, z-score, log1p, przedział kwantylowy, binning).
    - Zademonstruj **operacje logiczne** NumPy (maski) do filtrowania wierszy.
4. **Agregacje i grupowanie (pandas)**
    - Wykonaj **min. 3 agregacje** (groupby) dla sensownych kategorii (np. kraj/miasto/produkt/tydzień).
    - Pokaż **top-N** (np. 10 największych/na 100 k mieszkańców itp.).
5. **Czas i okna (jeśli masz daty)**
    - Zrób **resampling** (np. do tygodni/miesięcy) i **rolling window** (np. średnia krocząca z 7/30 okresów).
6. **Łączenie danych (join/merge)**

- Dołącz **drugą tabelę** (np. słownik kraj→region, kody ISO, populacja lub cennik/klasyfikacja).
- Użyj merge/join, wyjaśnij klucz łączenia i sprawdź duplikaty/niezmatchowane rekordy.

7. **Pivot/pivot_table**
   - Zbuduj **tabelę przestawną** (kolumny=czas/region, index=kategoria, values=metryka, aggfunc).
8. **Walidacja**
   - Pokaż **3 szybkie testy jakości** (np. brak wartości ujemnych w kolumnie X, suma = 100%, liczba unikalnych kluczy po merge).
9. **Eksport wyników**
   - Zapisz do CSV:
     - wyniki_agregacje.csv (Twoje kluczowe grupowania),
     - wyniki_pivot.csv (tabela przestawna),
     - opcjonalnie: sample_100.csv (losowa próbka 100 wierszy do wglądu).
10. **Wnioski (max 10 zdań)**
    - Odpowiedz na **2–3 pytania biznesowe/analityczne**, które sam(a) zdefiniujesz na starcie (np. „które regiony rosną najszybciej?”, „co napędza zmienność?”).
    - Zapisz je na końcu notatnika w sekcji „Wnioski”.

Podpowiedzi / wskazówki

- Użyteczne funkcje: pd.to_datetime, df.assign, np.where, np.log1p, df.groupby().agg, df.sort_values, df.nlargest, df.merge, pd.crosstab/pivot_table, df.resample('W'), df.rolling(7).mean().
- Sprawdź brakujące wartości: df.isna().mean().sort_values().
- Po merge od razu policz „sieroty”: df_merged[df_merged['_merge']!='both'] (ustaw indicator=True).

Bonus (opcjonalnie)

- Krótki **profil danych** (np. własna funkcja raportująca NA%, typy, kardynalność).
- **Wizualizacja** 1–2 kluczowych metryk (matplotlib/plotly).
- Prosty **model bazowy** (np. średnia krocząca do prognozy kolejnego okresu) i porównanie z wartością rzeczywistą.