

Zadanie zaliczeniowe z NYPD

12 XII 2022

Wersja 1.0

W tym roku prosimy Państwa o zbadanie zależności między dochodami państw (GDP - gross domestic product, PKB - produkt krajowy brutto), liczbami mieszkańców i emisją substancji lotnych.

Źródła danych

- [GDP](#) (plik csv, jest w spakowanym pliku zip). Tu są dane o GDP dla wszystkich krajów w latach 1960-2021 (niektórych danych brakuje).
- [Liczby mieszkańców](#) (plik csv, jest w spakowanym pliku zip). Dane o ludności w krajach też z lat 1960-2021.
- [Emisja CO2](#) (plik csv, jest też obok spakowany plik z danymi w formacie csv, i json w środku). Dane o emisji z lat 1751 (!) do 2014. Na stronie są też przykładowe fragmenty kodu pobierającego te dane.

Wszystkie pliki z danymi są też w naszym Moodle'u, razem z zadaniem.

Polecenie

Należy napisać w Pythonie program, który:

- Pobiera (używając argparse) z wiersza poleceń ścieżki do plików (w formacie csv) z danymi o GDP, populacji i emisji.
- Zakłada że są w formacie jaki dziś (choć w przyszłości danych może być więcej, czyli nie można założyć, że program ma działać dokładnie na tych plikach, które są dołączone do zadania, natomiast należy założyć, że format będzie zachowany) i je wczytuje.
- Czyści te dane (w tym punkcie nie ma wiele do zrobienia w odniesieniu do podanych źródeł danych).
- Wybiera tylko te lata, które są we wszystkich tabelach. Uwaga: chodzi o lata faktycznie występujące w tych tabelach, także w przyszłości, tzn. nie można w docelowej wersji na sztywno założyć (zaszyć w kodzie), że są to dane za lata 1960-2014), choć w pierwszych wersjach rozwiązania to może być wygodne.
- Scala dane po krajach i latach.
- Przeprowadza analizy (za pomocą bibliotek numpy lub pandas):

- Które kraje w poszczególnych latach z danymi, emitują najwięcej CO2 w przeliczeniu na mieszkańca. To znaczy generuje posortowaną po latach tabelkę pięcioma krajami o największej emisji na osobę (z podaną nazwą kraju, emisją na osobę i całkowitą emisją).
- Które kraje w poszczególnych latach z danymi mają największy przychód mieszkańca. To znaczy generuje posortowaną po latach tabelkę pięcioma krajami o największym dochodzie na mieszkańca (z podaną nazwą kraju, dochodem na mieszkańca i całkowitym dochodem).
- Które kraje (w przeliczeniu na mieszkańca) najbardziej zmniejszyły i zwiększyły przez ost. 10 lat (z danych) emisję CO2.

Dodatkowe wymagania

- Program powinien być podzielony na (co najmniej) dwa pythonowe moduły (nie muszą być duże) jeden zawierający bibliotekę operacji analizujących dane zgodnie z treścią zadania i drugi wywołujący te operacje.
- Oczekujemy, że źródła programu będą w repozytorium gita, z co najmniej pięcioma (5) uaktualnieniami (ang. commit),
- Oczekujemy złożenia (i przyjęcia) jednej prośby o dołączenie kodu (ang. pull requesta)?
- Testów jednostkowych kodu (tam gdzie to ma sens).
- Sprofilowania działania programu (należy dołączyć do rozwiązania plik z wynikiem profilowania, **nie** ma obowiązku przeprowadzania samych optymalizacji).
- Program powinien pozwalać zadawać w wierszu poleceń zakres lat (parametry -start rok i -koniec rok), wówczas badanych okres jest dodatkowo skracany do podanego przedziału. Jeśli przedział po tej operacji okaże się pusty, należy zgłosić (i gdzieś w programie obsłużyć) wyjątek.
- Program może mieć postać zwykłego programu lub Jupyterowego notatnika.
- Program należy dostarczyć w postaci pakietu instalowalnego poleceniem pip (w postaci `pip install ./ścieżka/do_pakietu`).
- Jeśli program wykryje jakieś niezgodności w danych (np. brak jakiegoś Państwa w niektórych danych), to powinien wypisać czytelny komunikat i działać dalej.

Sposób zaliczania

- Na tym przedmiocie jest egzamin ustny, ma on formę przedstawienia swojego programu zaliczeniowego prowadzącemu (zwykle ze swojej grupy)
- W czasie tej rozmowy oczekujemy:
 - Przedstawienia działania programu.
 - Przedstawienia i uzasadnienia swoich decyzji projektowych (np. podziału programu na moduły).
 - Przedstawienia swojego repozytorium z kodem.
- W programie oceniamy:

- Podział na moduły, funkcje, wprowadzenie parametrów,
- Jakość kodu (nazwy zmiennych i funkcji, komentarze).
- Sugerujemy stosowanie narzędzi automatycznie sprawdzających (część) zaleceń dotyczących czytelności kodu. Takie narzędzia mogą Państwo mieć wbudowane w Państwa IDE (np. W Pycharm) lub mogą Państwo skorzystać z zewnętrznych narzędzi (pylint, flake, ...),
- Warto również przeczytać zalecenia z [Przewodnika po stylu kodu w Pythonie PEP-8](#).
- Pokrycie kodu testami.
- Użycie bibliotek przedstawianych podczas zajęć
- Dostarczone pliki (np. wynik profilowania).

Inne możliwości

- Można wybrać inny temat zadania, który spełnia następujące warunki:
 - Jest nie mniejszej trudności niż tu przedstawione zadanie.
 - Został uzgodniony i zaakceptowany przez prowadzącego zajęcia w grupie.

Życzymy powodzenia!