

Pogoda

Hanna Taraszkiewicz, Maja Węgrzyk

4 Lutego 2026

1 Pobieranie

Na początek kod sięga do dodatkowych bibliotek, które pozwalają na wykonywanie nowych komend. `"os"` pozwala uruchamiać polecenia systemowe, a `"time"`, zmusza kod do robienia kilkusekundowych przerw w wybranych miejscach.

Następnie utworzone zostają foldery `"data"` oraz `"raw"`, wewnątrz niego, jeśli już istnieją ponowne uruchomienie programu ich nie zduplikuje.

Program tworzy dwie pętle, pierwszą przechodzącą przez lata 2001 - 2023, a drugą, wewnątrz niej przez kolejne miesiące każdego roku.

`"Url = [...]"` urzywa stworzonych w pętli dat by pobrać kolejne pliki z wszystkich miesięcy lat 2001 - 2023. Następnie `"output = f"data/raw/[...].zip"` zapisuje je w folderze `"raw"` pod nazwą odpowiadającą miesiącowi roku i w formacie `"zip"`.

Pętla jest domknięta przez: `"os.system(f'curl -L "{url}" -o "{output}"')"` komendę, która pobiera pliki z internetu zgodnie z kolejnymi datami oraz zapisuje je pod odpowiednią nazwą.

Ostatnie polecenie upewnia się, że między pobraniem każdego pliku program odczeka sekundę, dzięki czemu nagle pobranie durzej ilości plików, bo aż 276, nie powinno zostać odczytane jako cyberatak.

2 Rozpakowywanie

Jak wyrzej, stworzony zostaje nieduplikujący się przez pnowne uruchomienie kodu folder, nazwany `"unpacked"`, do którego rozpakowane zostaną pliki `"zip"`.

Poraz kolejny program tworzy pętlę lat i miesięcy, oraz nazwę pliku `"zip"`, która będzie zmieniać się przy każdym kroku pętli, umorzliwiając działanie programu na wszystkich pobranych wcześniej plikach poprzez wywołanie ich nazwy. `"Out_dir = "data/unpacked""` określa, że pliki zostaną rozpakowane do folderu `"unpacked"`. `"os.system(f'unzip -o "{zip_path}" -d "{out_dir}"')"` uruchamia system rozpakowujący do odpowiedniego folderu, w razie potrzeby (na wypadek ponownego uruchomienia w celu zapobiegnięcia utworzeniu zbędnych kopii) nadpisując istniejące w nim pliki.

3 Scalanie

W istniejącym już folderze "data" tworzony zostaje plik "dane.csv". Na wypadek kilkukrotnego uruchomienia kodu zostaje on nadpisany. "encoding='cp1250'" umożliwia w nim istnienie polskich liter.

Ponownie utworzona jest pętla, oraz odpowiadające kolejnym datom nazwy plików "csv" do odczytu.

Następnie plik zostaje otwarty, przeczytany (przez kod) linijka po linijce, pozbawiony cudzysłowów, zapisany do pliku "dane.csv" i zamknięty. Po każdej iteracji pętli program wyświetla informację o przeczytaniu kolejnego pliku.

Następnie cały plik "dane.csv" zostaje zamknięty, zawierając dane z wszystkich pobranych wcześniej plików z lat 2001 - 2023

4 Charakterystyki danych

Dodatkowe biblioteki, "numpy", rozpoznawane przez kod jako skrót "np", oraz "matplotlib.pyplot", jako "plt", pozwalają na korzystanie z zaawansowanych narzędzi matematycznych oraz tworzenie wykresów na bazie dostępnych programowi danych.

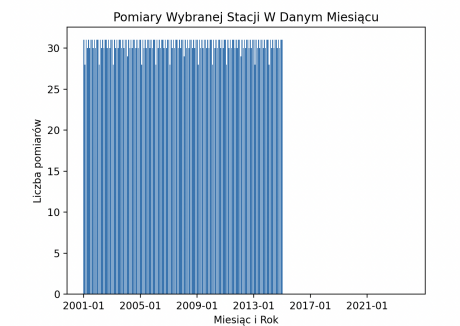
Program wczytuje utworzony uprzednio scalony plik, traktując go w całości jako tekst, oddzielając kolejne kolumny przecinkami. Następnie sprawdza ilość linijek tekstu, różnych stacji pomiarowych, oraz niepowtarzających się dat.

Utworzona zostaje tablica składająca się z niepowtarzalnego ID stacji, jej nazwy, oraz daty pomiaru. Na bazie tej tablicy system sprawdza ilość różnych dat, tworzy jej listę dla każdej stacji a następnie sprawdza jaka jest ilość stacji z pełną satą pomiarów.

Wybrana została stacja z pełną historią, która wypisuje wszystkie daty dla których ma pomiary, a więc wszystkie dni wszystkich miesięcy lat 2001 - 2023, co urzyte będzie przy tworzeniu wykresu.

Stacja z niepełną historią pomiarów, zostaje wybrana z "stacje_z_data", na bazie swojego unikalnego ID. Następnie program liczy ile dni pomiarowych ta stacja miała dla danego miesiąca i roku. Ta ilość zostaje urzyta w wykresie.

Z powodu nagromadzenia nazw miesięcy na osi OX ich ilość została zmniejszona by wykres wyglądał przejrzystiej.



Rysunek 1: Schemat pomiarów stacji w danym miesiącu

5 Ekstrema tempreatur

Na początek program importuje potrzebne biblioteki.

Utworzona zostaje pusta lista na wiersze, która zostanie urzyta później.

Kod otwiera plik `"dane.csv"` oraz upewnia się że zostanie on zamknięty po wypełnieniu swojej funkcji. Przetwarza wszystkie dane pliku zostawiając jedynie piątą i siódmą kolumnę - te w których znajdują się informacje o maksymalnej i minimalnej temperaturze.

Zapisuje te dane w postaci tablicy NumPy, co pozwala na łatwiejszą nimi manipulację.

Zamienia tekst na liczby, umozliwiając porównywanie ich wielkości. Następnie szuka ekstremów każdej kolumny, maksimów dla najwyższych odnotowanych w danym dniu temperatur i minimów dla najniższych. Zapisuje te dane oraz odpowiadające im wiersze (w których zawierają się nazwy stacji badawczych i daty wykonnia pomiarów).

Następnie wyświetla wyniki pokazując ekstremalne temperatury oraz datę, nazwę i ID stacji która je odnotowała.

6 Porównanie

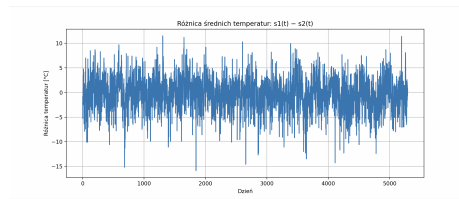
Wybrane zostają dwie stacje z pełną historią odczytów, a pobrane z nich dane są zapisane jako `s1` i `s2`.

Daty i odnotowane temperatury zostają zapisane tak by program mógł je odczytać jako słowniki, których kluczami są dni pobrania danych a wartością średnia odnotowana temperatura.

Dla dni w których stacje miały pomiar (więc wszystkich w latach 2001 - 2023) sortuje daty chronologicznie.

Następnie odejmuje dane z `s1` od `s2` dla każdego z tych dni, zwracając listę różnic średnich dziennych temperatur.

Program tworzy numery kolejnych dni, a następnie przygotowuje wykres



Rysunek 2: Różnice średnich temperatur między dwoma stacjami

rysując na nim linie odpowiadające zmianą w różnicy temperatur między stacjami.

Kod znajduje największą różnicę temperatur poprzez zidentyfikowanie najwyższego punktu na wykresie i wyświetla odnaleziony wynik.