Przetwarzanie plików

Zadanie 10.1 (*MG; Listing katalogu on-line*)**.**
Strona internetowa *cran.rstudio.com/src/base/R-3/* udostępnia przykładowy listing plików standardowo generowany przez serwer Apache (w formie dokumentu HTML). Napisz funkcję `getApacheDirListing()`, która dla danego adresu URL (I argument) zwraca ramkę danych zawierającą informacje udostępnione w listingu. Wynikowy obiekt powinien składać się z czterech kolumn: `url` – URL pliku (napis), `name` – nazwa pliku (napis), `modtime` – czas ostatniej modyfikacji pliku (`POSIXct`) oraz `size` – przybliżony rozmiar w bajtach (liczba rzeczywista, założenie: 1 KB to 1000 B).

Zadanie 10.2 (*MG*)**.**
Napisz funkcję, która wyznacza łączny rozmiar wszystkich plików w podanym katalogu i jego podkatalogach.

Zadanie 10.3 (*MG*)**.**
Napisz funkcję `scal_pliki()`, która dokonuje złączenia wszystkich plików o rozszerzeniu `ext` (domyślnie `.txt`) znajdujących się w podanym katalogu `dir`. Ścieżkę dostępu do pliku wynikowego określa argument `out`.

Zadanie 10.4 (*MG*)**.**
Napisz funkcję, która znajduje ścieżki dostępu do `k` (domyślnie 10) najstarszych plików w podanym katalogu. Posortuj wynik względem rozmiaru plików.

Zadanie 10.5 (*MG*)**.**
Załóżmy, że w katalogu `dir` znajduje się kolekcja Twoich ulubionych seriali. Na przykład niech będą to pliki o następujących nazwach:

```
## [1] "Bolek_i_Lolek Season 1 Episode 4.avi"
## [2] "bolekilolek s01e13.flv"
## [3] "Bolek.i.Lolek.s02e03.tylkonauzytekwlasny.mp4"
```

Napisz funkcję, która zmienia nazwy wszystkich plików na spójne, postaci `SxxEyy`, gdzie `xx` – numer sezonu, `yy` – numer odcinka, np. `S01E04.avi`, `S01E13.flv` i `S02E03.mp4`.

Zadanie 10.6 (*MG*)**.**
Strona Wikipedii *en.wikipedia.org/wiki/List_of_cities_by_latitude* zawiera wykaz miejscowości uporządkowanych względem szerokości geograficznej. Większość podlinkowanych podstron zawierających informacje o miastach ma sekcję *Climate Data*, która podaje dane na temat klimatu. Wydobądź dane dla lipca dla wszystkich miast i przedstaw w postaci jednej ramki danych.

Zadanie 10.7 (*MG*)**.**
Strona *cran-logs.rstudio.com/* zawiera dane, oddzielnie dla każdego dnia, na temat wszystkich pobrań pakietów R z serwera identyfikowanego jako *0-Cloud*. Określ nazwy dziesięciu najczęściej pobieranych pakietów w ostatnim miesiącu.

Zadanie 10.8 (*MG*)**.**
Wczytaj plik JSON, np. wygenerowany przy użyciu zapytania do *Yahoo! Finance API*.

Zadanie 10.9 (*MG*)**.**
Wczytaj plik XML, np. wygenerowany przy użyciu zapytania do *Yahoo! Finance API*.

Zadanie 10.10 (*MG; Wydobywanie adresów stron WWW*)**.**
Napisz funkcję `extractUrls()`, która z danego pliku tekstowego `infname` (I argument) wyłuska wszystkie adresy internetowe postaci `http://*` lub `www.*`. Rezultaty przedstaw jako wektor napisów.

Zadanie 10.11 (*MG; Pobieraczek*)**.**
Napisz funkcję `massDownload()`, która dla danego wektora napisów `urls` (I argument) zawierającego n adresów URL różnych stron internetowych pobierze i zapisze je w oddzielnych plikach `1.html`, ..., n`.html` we wskazanym katalogu `outdir` (II argument).

Zadanie 10.12 (*MG; Zliczanie liczby wystąpień wszystkich słów*)**.**
Napisz funkcję `words()`, która dla danego pliku tekstowego (I argument, `fname`) zwróci ramkę danych o dwóch kolumnach zawierającą wszystkie występujące słowa (`word`) oraz ich liczby wystąpień (`count`). Ramka danych powinna być posortowana nierosnąco względem liczby wystąpień słów.

Zadanie 10.13 (*MG; Zmiana formatu plików CSV*)**.**
Napisz funkcję `CSVToCSV2()`, która zamieni separatory pól „`,`” na „`;`” oraz separatory części ułamkowej liczb „`.`” na „`,`” bez wywoływania `read.table()`, `write.table()` i ich pochodnych. Wejście: plik `infname` (I argument), wyjście: `outfname` (II argument). Uważaj, na to, by nie zmieniać zawartości napisów (zawartych w cudzysłowie).

Zadanie 10.14 (*MG; BibTEX*)**.**
Napisz funkcję `BibTeX2data.frame()`, która jako argument przyjmuje jeden napis `fname` określający ścieżkę dostępu do pliku zawierającego bibliografię w formacie BibTEX-a (nt. tego formatu poczytaj w internecie).

Funkcja powinna zwracać ramkę danych. Każdy wiersz ramki określa inny wpis bibliograficzny. Kolumny:

1. typ wpisu (np. `article`, `book`, ...);
2. identyfikator wpisu (np. `paper123`);
... + dodatkowe pola postaci `pole=...` z podanego pliku (np. `journal`, `authors`, `pages`, ...).

Jeśli jakieś pole nie występuje w danym wpisie (np. `journal` najczęściej nie występuje w `book`), to wstaw na to miejsce `NA`.

Przykładowy plik BibTEX-a:

```
@article{paper1,
   author={J. Kowalski and Y. Hui},
   journal={Journal of Everything},
   title={P = NP},
   year={1999}
}

@book{xyz,
   author="E. Schmidt",
   publisher="PWN",
   year={2013},
   title={A general theory of everything}
}
```

Wynik:

```
     type     id                 author               journal
1 article paper1 J. Kowalski and Y. Hui Journal of Everything
2    book    xyz             E. Schmidt                  <NA>

                          title year publisher
                        P = NP 1999      <NA>
 A general theory of everything 2013       PWN
```