

Ramki danych cz. 2

Wszystkie zadania rozwiązujemy bez używania pętli. Zbiory danych można pobrać ze strony <http://gagolewski.com/resources/>.

Zadanie 7.1 (MG).

Wykonaj następujące polecenia.

- (a) Wybierz 100 losowych wierszy z ramki danych `airports`.
- (b) Wybierz 5% wierszy w sposób losowy.
- (c) Wybierz 100 pierwszy wierszy.
- (d) Wybierz 100 ostatnich wierszy.

Perform what follows.

- (a) *Select 100 random rows from `airports`.*
- (b) *Select 5% random rows.*
- (c) *Select first 100 rows.*
- (d) *Select last 100 rows.*

Zadanie 7.2 (MG).

Na podstawie ramki danych `winequality-all` stwórz ramkę zawierającą tylko białe wina. Następnie podziel ją w sposób losowy na dwie ramki: `wines_train` (80% wszystkich wierszy) oraz `wines_test` (pozostałe 20%).

In the `winequality-all` data set, leave out only the white wines. Partition the resulting data frame randomly into two data frames: `wines_train` (80% of the rows) and `wines_test` (remaining 20%).

Zadanie 7.3 (MG).

Do ramki danych `vehicles` z pakietu `fueleconomy` (`fueleconomy::vehicles`) dodaj nową kolumnę `z_pty` oraz `z_hwy`, takie, że są to ustandaryzowane (z -scores) zmienne `pty` (`city`-) and `hwy` (`highway-fuel economy`, in mpg) w grupach wyznaczonych przez kolumnę `class`. Innymi słowy, standaryzujemy ($\text{średnia} = 0$, $\text{odchylenie standardowe} = 1$) zmienne w każdej podgrupie. Na początku jednak zamień jednostki z *mile-per-gallon* na $l/100\text{km}$.

Given `fueleconomy::vehicles`, add new columns `z_pty` and `z_hwy`, that give the z -scores of `pty` (`city`-) and `hwy` (`highway-fuel economy`, in mpg) relative to (grouped by) `class`; in other words, standardize ($\text{mean}=0$, $\text{sd}=1$) the corresponding variables within each subgroup. However, first convert the input values to $l/100\text{ km}$.

Zadanie 7.4 (AO; „Rozwijanie” macierzy).

Napisz funkcję `rozwin()`, która przekształca daną macierz rozmiaru $n \times m$ (niekoniecznie liczbową) z ustawionym atrybutem `dimnames` na ramkę danych zawierającą nm obserwacji i trzy kolumny o nazwach zadanych przy użyciu odpowiedniego argumentu funkcji. Wartości z macierzy mają znajdować się w pierwszej kolumnie, a w kolejnych dwóch – kombinacje nazw wierszy i kolumn odpowiadające podanym poziomom czynnika.

Na przykład obiekt `WorldPhones` (wbudowany) zawiera dane o liczbie telefonów (w tysiącach) w różnych regionach świata w wybranych latach. Wynikiem wywołania `rozwin(WorldPhones, c("ile", "gdzie", "kiedy"))` może być:

```
      ile      gdzie kiedy
...
2  60423    N.Amer  1956
3  64721    N.Amer  1957
...
9  29990    Europe  1956
10 32510    Europe  1957
...
```

Zadanie 7.5 (*MG; „Odwijanie” macierzy*).

Napisz funkcję odwrotną do funkcji z zad. 7.4. Dana jest ramka danych zawierająca nm wierszy oraz 3 kolumny (pierwsza – dowolnego typu, druga i trzecia – typu czynnikowego, odpowiednio o n i m poziomach). Obserwacje zawierają wszystkie możliwe kombinacje poziomów dwóch czynników, ale nie możemy założyć, że są one koniecznie ułożone w jakimś określonym porządku (funkcja ma działać dla dowolnej permutacji obserwacji). Wynikiem ma być macierz rozmiaru $n \times m$ o elementach pochodzących z pierwszej kolumny ramki danych. Atrybut `dimnames` ustawiamy na podstawie wartości poziomów pierwszego i drugiego czynnika.