

Przekształcanie danych: `pivot_longer()` i `pivot_wider()`

Statystyka I z R

Bartosz Maćkiewicz

Przekształcanie danych z `tidyr`

Do tej pory wykonywaliśmy różne operacje na ramkach danych: wybieraliśmy wiersze i kolumny, dodawaliśmy nowe zmienne oraz obliczaliśmy statystyki dla grup. Czasami jednak przed przystąpieniem do analizy trzeba zmienić sam układ danych. Te same informacje możemy bowiem zapisać na kilka sposobów, a nie każdy z nich będzie równie wygodny do dalszej pracy.

Omówimy sobie dwie funkcje z pakietu `tidyr`, który należy do poznawanego przez Państwa `tidyverse`: `pivot_longer` oraz `pivot_wider`. Pierwsza pozwala przekształcić dane z formatu “szerokiego” do “długiego”, druga umożliwia przekształcenie w przeciwną stronę. Za chwilę zobaczymy na przykładzie, co właściwie oznaczają te nazwy.

Ładujemy pakiety. Jeżeli mają Państwo już zainstalowany `tidyverse`, nie trzeba instalować osobno `tidyr`.

```
# install.packages('tidyverse')  
library(tidyverse)
```

Format szeroki i długi

Do ćwiczeń wykorzystamy dane UNICEF dotyczące śmiertelności dzieci poniżej 5. roku życia. Plik `unicef-u5mr.csv` znajdują Państwo w załącznikach do notatnika. Proszę zapisać go w tym samym katalogu co plik `Qmd` i ustawić ten katalog jako katalog roboczy, kiedy wykonują Państwo kod w konsoli.

```
df <- read.csv('unicef-u5mr.csv')
```

Skrót **U5MR** oznacza *under-five mortality rate*. Jest to prawdopodobieństwo śmierci dziecka przed ukończeniem 5. roku życia, wyrażone na 1000 żywych urodzeń. Więcej o tym wskaźniku mogą Państwo przeczytać na [stronie UNICEF](#). W naszym pliku znajdują się dane historyczne, z kolumnami dla lat 1950–2015.

Cała tabela ma sporo kolumn, więc na początek wybierzmy trzy kraje i trzy lata. Użyjemy do tego znanych już funkcji `filter` i `select`.

```
fragment <- df %>%  
  filter(CountryName %in% c('France', 'Germany', 'Poland')) %>%  
  select(CountryName, U5MR.2000, U5MR.2005, U5MR.2010)  
  
knitr::kable(fragment)
```

CountryName	U5MR.2000	U5MR.2005	U5MR.2010
France	5.4	4.6	4.3
Germany	5.4	4.7	4.2
Poland	9.3	7.6	5.8

Nasze dane są w **formacie szerokim**: jeden wiersz odpowiada jednemu krajowi, a wartości wskaźnika z kolejnych lat znajdują się w osobnych kolumnach. Na przykład informacje o Polsce w 2000 i 2005 roku są zapisane w tym samym wierszu, ale w dwóch różnych kolumnach.

Możemy jednak zapisać te same informacje w **formacie długim**. Wtedy jeden wiersz będzie odpowiadał jednej parze kraj–rok. Zamiast osobnych kolumn dla kolejnych lat będziemy mieć kolumnę z rokiem oraz kolumnę z wartością wskaźnika. Nazwa kraju będzie się więc powtarzała w kilku wierszach. To jest w porządku: każdy z tych wierszy będzie dotyczył innego roku.

Po co dokonywać takiej konwersji? Wszystko zależy od tego, co chcemy zrobić z danymi. Format szeroki bywa wygodny, kiedy oglądamy tabelę i porównujemy wartości z kilku lat. Format długi ułatwi nam między innymi rysowanie wykresów i obliczanie statystyk według kraju albo roku. Nie zmieniamy przy tym wartości wskaźnika, tylko sposób ich zapisania.

`pivot_longer`

Funkcja `pivot_longer` pozwala nam przenieść informacje z nazw kolumn do nowej kolumny. Spróbujmy najpierw przekształcić nasz mały fragment danych.

```
fragment_dlugie <- pivot_longer(
  fragment,
  cols = starts_with('U5MR.'),
  names_to = 'year',
  values_to = 'U5MR'
)

knitr::kable(fragment_dlugie)
```

CountryName	year	U5MR
France	U5MR.2000	5.4
France	U5MR.2005	4.6
France	U5MR.2010	4.3
Germany	U5MR.2000	5.4
Germany	U5MR.2005	4.7
Germany	U5MR.2010	4.2
Poland	U5MR.2000	9.3
Poland	U5MR.2005	7.6
Poland	U5MR.2010	5.8

Przyjrzyjmy się argumentom tego wywołania:

- Pierwszy argument to ramka danych, którą przekształcamy.
- `cols` określa kolumny, z których chcemy zebrać wartości. Za pomocą `starts_with('U5MR.')` wybieramy te, których nazwy zaczynają się od `U5MR.`. Kolumna `CountryName` pozostaje kolumną identyfikującą kraj.
- `names_to` określa nazwę nowej kolumny, do której trafią dotychczasowe nazwy kolumn. U nas będzie to `year`.
- `values_to` określa nazwę kolumny, do której trafią wartości. Nazwiemy ją `U5MR`.

Widzimy, że zamiast trzech wierszy mamy teraz dziewięć: po jednym dla każdej kombinacji kraju i roku. Proszę sprawdzić, gdzie znalazła się wartość dla Polski z 2005 roku. Jest to dokładnie ta sama liczba, którą mieliśmy w szerokiej tabeli.

Nazwy kolumn i liczby

W kolumnie `year` znajdują się na razie napisy takie jak `U5MR.2000`. Chcielibyśmy mieć tam sam rok. W pliku CSV nazwy mają postać `U5MR 2000`, ale użyte przez nas `read.csv` domyślnie zamienia spacje w nazwach kolumn na kropki. Stąd właśnie `U5MR.2000`.

Możemy od razu usunąć ten wspólny początek nazw za pomocą argumentu `names_prefix`. Argument ten przyjmuje wyrażenie regularne. Zapis `U5MR\\.` oznacza tutaj dosłowny tekst `U5MR.` na początku nazwy. Dwa ukośniki odwrotne są potrzebne w napisie R, żeby kropka była traktowana jako kropka, a nie znak dopasowujący dowolny znak.

Spróbujmy teraz wykonać przekształcenie na całym zbiorze danych. Dodamy także `mutate`, żeby przekonwertować rok z napisu na liczbę całkowitą.

```
df_dlugie <- df %>%
  pivot_longer(
    cols = starts_with('U5MR.'),
    names_to = 'year',
    names_prefix = 'U5MR\\.',
    values_to = 'U5MR'
  ) %>%
  mutate(year = as.integer(year))
```

Obejrzyjmy ponownie te same kraje i lata, które wybraliśmy na początku.

```
df_dlugie %>%
  filter(CountryName %in% c('France', 'Germany', 'Poland'),
         year %in% c(2000, 2005, 2010)) %>%
  knitr::kable()
```

CountryName	year	U5MR
France	2000	5.4
France	2005	4.6
France	2010	4.3
Germany	2000	5.4
Germany	2005	4.7
Germany	2010	4.2
Poland	2000	9.3
Poland	2005	7.6
Poland	2010	5.8

Tym razem w kolumnie `year` mamy liczby. Możemy więc używać roku w porównaniach, porządkować obserwacje chronologicznie albo nanieść go na liczbową oś wykresu.

Co z brakującymi wartościami?

W naszym pliku nie dla każdego kraju i roku mamy podaną wartość wskaźnika. Takie braki są oznaczone jako NA. `pivot_longer` domyślnie je zachowuje: w danych długich otrzymamy wiersz z nazwą kraju, rokiem i brakującą wartością U5MR.

```
df_dlugie %>%  
  filter(is.na(U5MR)) %>%  
  head(5) %>%  
  knitr::kable()
```

CountryName	year	U5MR
Afghanistan	1950	NA
Afghanistan	1951	NA
Afghanistan	1952	NA
Afghanistan	1953	NA
Afghanistan	1954	NA

Brak danych nie oznacza zera. NA mówi nam, że w tym pliku nie podano wartości dla danej pary kraj-rok. Zastąpienie go zerem zmieniłoby znaczenie danych. Samo przekształcanie tabeli nie jest powodem, żeby tak robić.

Funkcja ma również argument `values_drop_na`, który pozwala pominąć wiersze z brakującymi wartościami. W naszym przykładzie pozostawiamy jego domyślną wartość `FALSE`, dzięki czemu zachowujemy wszystkie pary kraj-rok z wyjściowej tabeli.

`pivot_wider`

Spróbujmy teraz wrócić do poprzedniego układu. Chcemy, żeby każdy kraj zajmował jeden wiersz, a każdy rok miał swoją kolumnę. Służy do tego funkcja `pivot_wider`.

```
df_szerokie <- df_dlugie %>%  
  pivot_wider(  
    id_cols = CountryName,  
    names_from = year,  
    values_from = U5MR,  
    names_prefix = 'U5MR.'  
  )
```

Argument `id_cols` wskazuje kolumnę identyfikującą wiersze wynikowej tabeli. U nas jest to kraj. `names_from` mówi, skąd wziąć nazwy nowych kolumn, a `values_from` określa kolumnę z wartościami, które mają się w nich znaleźć.

Dodaliśmy też `names_prefix = 'U5MR.'`, żeby odtworzyć nazwy takie jak `U5MR.2000`. W `pivot_wider` podajemy po prostu tekst, który ma zostać dopisany przed rokiem; nie jest to wyrażenie regularne. Dlatego tutaj nie potrzebujemy ukośników przed kropką.

Sprawdźmy znany już fragment tabeli.

```
df_szerokie %>%
  filter(CountryName %in% c('France', 'Germany', 'Poland')) %>%
  select(CountryName, U5MR.2000, U5MR.2005, U5MR.2010) %>%
  knitr::kable()
```

CountryName	U5MR.2000	U5MR.2005	U5MR.2010
France	5.4	4.6	4.3
Germany	5.4	4.7	4.2
Poland	9.3	7.6	5.8

Otrzymaliśmy te same wartości co na początku. Proszę zwrócić uwagę, że przekształcenie nie wymagało obliczania żadnych średnich ani sum. Każda wartość miała po prostu trafić do właściwej komórki.

Jedna wartość dla pary kraj–rok

W tym przykładzie dla każdej pary kraj–rok mamy dokładnie jeden wiersz. To ważne, bo w szerokiej tabeli taka para wskazuje jedną komórkę. Gdybyśmy mieli dwa różne wyniki dla tego samego kraju i roku, funkcja nie mogłaby sama zdecydować, który z nich ma tam umieścić.

Możemy sprawdzić, czy jakaś para występuje więcej niż raz, używając `count`. Funkcja ta zlicza wiersze dla każdej kombinacji wskazanych kolumn i zapisuje ich liczbę w kolumnie `n`.

```
df_dlugie %>%
  count(CountryName, year) %>%
  filter(n > 1)
```

```
# A tibble: 0 x 3
# i 3 variables: CountryName <chr>, year <int>, n <int>
```

Pusty wynik oznacza, że nie znaleźliśmy takich par. W innych danych powtórzenia mogą wynikać na przykład z kilku pomiarów wykonanych w tym samym roku. Wtedy trzeba zdecydować, czy rozróżnić je dodatkową kolumną, czy podsumować pomiary. `pivot_wider` może zwrócić kolumny zawierające listy, gdy do jednej komórki pasuje kilka wartości. Nie oznacza to, że obliczył za nas średnią. Na razie wystarczy, żeby zapamiętali Państwo, skąd może wziąć się taki wynik.

Wykres dla Polski

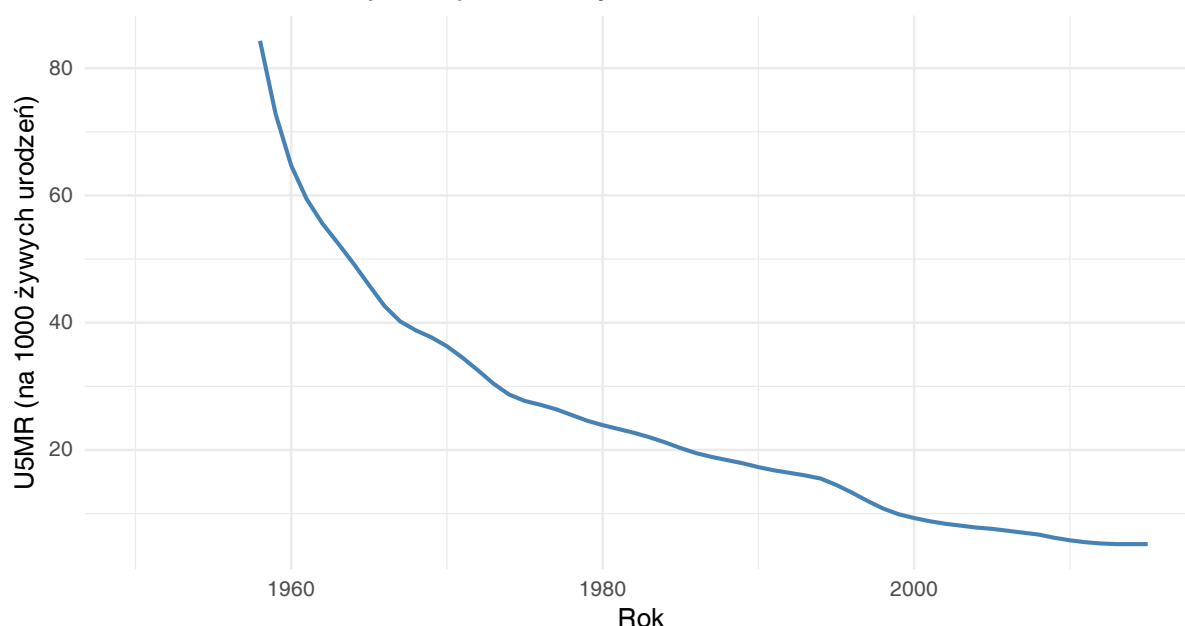
Mając dane w formacie długim, możemy łatwo narysować wykres pokazujący zmiany śmiertelności dzieci w Polsce. Najpierw wybierzmy odpowiednie wiersze i uporządkujmy je według roku.

```
df_polska <- df_dlugie %>%  
  filter(CountryName == 'Poland') %>%  
  arrange(year)
```

Teraz rok możemy przypisać do osi poziomej, a wartość wskaźnika do pionowej. Dla Polski w części najstarszych lat brakuje wartości wskaźnika. Argument `na.rm = TRUE` w `geom_line` pozwoli pominąć te braki przy rysowaniu bez wypisywania ostrzeżenia. W ramce danych nadal pozostają one oznaczone jako NA.

```
ggplot(df_polska, aes(x = year, y = U5MR)) +  
  geom_line(colour = 'steelblue', linewidth = 0.8,  
            na.rm = TRUE) +  
  labs(  
    title = 'Śmiertelność dzieci poniżej 5. roku życia w Polsce',  
    x = 'Rok',  
    y = 'U5MR (na 1000 żywych urodzeń)'  
  ) +  
  theme_minimal()
```

Śmiertelność dzieci poniżej 5. roku życia w Polsce



Widzimy, jak wartość wskaźnika malała w latach objętych naszym zbiorem. W szerokiej tabeli kolejne lata były nazwami kolumn. Po przekształceniu stały się wartościami jednej zmiennej, którą możemy bezpośrednio wykorzystać na wykresie. To jeden z powodów, dla których warto umieć przechodzić między tymi układami danych.

Jeżeli spotkają Państwo `melt` i `dcast`

W starszych materiałach i przykładach mogą Państwo trafić na pakiet `reshape2`. Funkcja `melt` służyła w nich do przekształcania danych w kierunku długiego formatu, a `dcast` do tworzenia szerokiej tabeli. W tym notatniku te operacje wykonujemy za pomocą `pivot_longer` i `pivot_wider` z pakietu `tidyr`.

Więcej przykładów znajdą Państwo w [oficjalnym wprowadzeniu do przekształcania danych w `tidyr`](#). Przy samodzielnej pracy warto też zaglądać do dokumentacji `?pivot_longer` i `?pivot_wider`.