The motivating case is something that happens all the time when working with social science data. We’ll load the tidyverse, and then quickly make up some sample data to work with.

library(tidyverse)

gen\_cats <- function(x, N = 1000) {

sample(x, N, replace = TRUE)

}

set.seed(101)

N <- 1000

income <- rnorm(N, 100, 50)

vars <- list(stratum = c(1:8),

sex = c("M", "F"),

race = c("B", "W"),

educ = c("HS", "BA"))

df <- as\_tibble(map\_dfc(vars, gen\_cats))

df <- add\_column(df, income)

What we have are measures of sex, race, stratum (from a survey, say), education, and income. Of these, everything is categorical except income. Here’s what it looks like:

df

## # A tibble: 1,000 x 5

## stratum sex race educ income

##

## 1 6 F W HS 83.7

## 2 5 F W BA 128.

## 3 3 F B HS 66.3

## 4 3 F W HS 111.

## 5 6 M W BA 116.

## 6 7 M B HS 159.

## 7 8 M W BA 131.

## 8 3 M W BA 94.4

## 9 7 F B HS 146.

## 10 2 F W BA 88.8

## # … with 990 more rows

Let’s say we want to transform this to a wider format, specifically by widening the educ column, so we end up with columns for both the HS and BA categories, and as we do so we want to calculate both the mean of income and the total n within each category of educ.

For comparison, one could do this with data.table in the following way:

data.table::setDT(df)

df\_wide\_dt <- data.table::dcast(df, sex + race + stratum ~ educ,

fun = list(mean, length),

value.var = "income")

head(df\_wide\_dt)

## sex race stratum income\_mean\_BA income\_mean\_HS income\_length\_BA income\_length\_HS

## 1: F B 1 93.78002 99.25489 19 6

## 2: F B 2 89.66844 93.04118 11 16

## 3: F B 3 112.38483 94.99198 13 16

## 4: F B 4 107.57729 96.06824 14 15

## 5: F B 5 91.02870 92.56888 11 15

## 6: F B 6 92.99184 116.06218 15 15

Until recently, widening or spreading on multiple values like this was kind of a pain when working in the tidyverse. (The code there still works fine.) Previously, you had to put spread() and gather() through a slightly tedious series of steps, best wrapped in a function you’d have to write yourself. No more! Since tidyr v1.0.0 has been released, though, the new function pivot\_wider() (and its complement, pivot\_longer()) make this common operation more accessible.

Here’s how to do it now. Remember that in the tidyverse approach, we’ll first do the summary calculations, mean and length, respectively, though we’ll use dplyr’s n() for the latter. Then we widen the long result.

tv\_pivot <- df %>%

group\_by(sex, race, stratum, educ) %>%

summarize(mean\_inc = mean(income),

n = n()) %>%

pivot\_wider(names\_from = (educ),

values\_from = c(mean\_inc, n))

This gives us an object that’s equivalent to the df\_wide\_dt object created by data.table.

tv\_pivot

## # A tibble: 32 x 7

## # Groups: sex, race, stratum [32]

## sex race stratum mean\_inc\_BA mean\_inc\_HS n\_BA n\_HS

##

## 1 F B 1 93.8 99.3 19 6

## 2 F B 2 89.7 93.0 11 16

## 3 F B 3 112. 95.0 13 16

## 4 F B 4 108. 96.1 14 15

## 5 F B 5 91.0 92.6 11 15

## 6 F B 6 93.0 116. 15 15

## 7 F B 7 102. 121. 13 13

## 8 F B 8 105. 88.3 14 8

## 9 F W 1 92.6 110. 19 13

## 10 F W 2 98.5 101. 15 19

## # … with 22 more rows

And there you have it. Be sure to check out the complement of pivot\_wider(), pivot\_longer(), also.