Convert Table from Wide to Long with dplyr

Fan Wang

2023-07-18

Contents

1 Wide to Long

Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools research support package, R4Econ examples page, PkgTestR packaging guide, or Stat4Econ course page.

Using the pivot_wider function in tidyr to reshape panel or other data structures

1.1 Wide to long panel, single variable

We have a matrix of values, the values are *ev*. Each row corresponds to a different value of the a variable, each column represents a different value of the z variable.

Based on this matrix, we create a table where each unit of observation is for a specific a and z variable combination. So the matrix is turned from wide to long.

The resulting long table has 5 variables

1. a: values of the a variable, the original matrix row labels
2. ai: an index from 1, indicating the original matrix row index
3. z: values of the z variable, the original matrix column lables
4. zi: an index from 1, indicating hte original matrix column index

First, we create the matrix.

```
# Generate A Matrix
set.seed(123)
ar_a <- c(1.1,5.1)
ar_z <- seq(-2.5, 2.53, length.out=11)
mt_ev = matrix(rnorm(length(ar_a)*length(ar_z)),
  nrow=length(ar_a), ncol=length(ar_z))

# Name Matrix
rownames(mt_ev) <- paste0('ai', seq(1:length(ar_a)))
colnames(mt_ev) <- paste0('zi', seq(1:length(ar_z)))

# to tibble
tb_ev <- as_tibble(mt_ev) %>% rowid_to_column(var = "ai")
```

Wide table

ai	zi1	zi2	zi3	zi4	zi5	zi6	zi7	zi8	zi9	
1	-0.5604756	1.5587083	0.1292877	0.4609162	-0.6868529	1.2240818	0.4007715	-0.5558411	0.4978505	
2	-0.2301775	0.0705084	1.7150650	-1.2650612	-0.4456620	0.3598138	0.1106827	1.7869131	-1.9666172	-

```
# Print
print(mt_ev)
```

```
##            zi1        zi2       zi3        zi4        zi5       zi6       zi7
## ai1 -0.5604756 1.55870831 0.1292877  0.4609162 -0.6868529 1.2240818 0.4007715
## ai2 -0.2301775 0.07050839 1.7150650 -1.2650612 -0.4456620 0.3598138 0.1106827
##            zi8        zi9       zi10       zi11
## ai1 -0.5558411  0.4978505  0.7013559 -1.0678237
## ai2  1.7869131 -1.9666172 -0.4727914 -0.2179749
```

```
# Display
kable(tb_ev, caption = "Wide table") %>% kable_styling_fc()
```

Second, we convert the table wide to long.

```
# longer
tb_ev_long <- tb_ev %>%
  pivot_longer(cols = starts_with('zi'),
               names_to = c('zi'),
               names_pattern = paste0("zi(.*)"),
               values_to = "ev") %>%
  mutate(zi = as.numeric(zi))

# Merge with a and z values
tb_ev_long <- tb_ev_long %>%
  left_join(as_tibble(ar_a) %>%
              rowid_to_column(var = "ai") %>%
              rename(a = value)
              , by = 'ai') %>%
  left_join(as_tibble(ar_z) %>%
              rowid_to_column(var = "zi") %>%
              rename(z = value),
            by = 'zi') %>%
  select(a,ai,z,zi,ev)

# Display
kable(tb_ev_long, caption = "Long table") %>% kable_styling_fc()
```

1.2 Wide to long panel, multiple variables

We have a dataset where each row contains data from a different year. We have four variables, observed wage, simulated wage, observed labor quantities, and simulated labor quantities.

We generate reshape this file to have four variables:

1. year
2. categorical for wage or quantity
3. categorical for observed or simulated
4. a numerical column with wage and quantity values

Long table

a	ai	z	zi	ev
1.1	1	-2.500	1	-0.5604756
1.1	1	-1.997	2	1.5587083
1.1	1	-1.494	3	0.1292877
1.1	1	-0.991	4	0.4609162
1.1	1	-0.488	5	-0.6868529
1.1	1	0.015	6	1.2240818
1.1	1	0.518	7	0.4007715
1.1	1	1.021	8	-0.5558411
1.1	1	1.524	9	0.4978505
1.1	1	2.027	10	0.7013559
1.1	1	2.530	11	-1.0678237
5.1	2	-2.500	1	-0.2301775
5.1	2	-1.997	2	0.0705084
5.1	2	-1.494	3	1.7150650
5.1	2	-0.991	4	-1.2650612
5.1	2	-0.488	5	-0.4456620
5.1	2	0.015	6	0.3598138
5.1	2	0.518	7	0.1106827
5.1	2	1.021	8	1.7869131
5.1	2	1.524	9	-1.9666172
5.1	2	2.027	10	-0.4727914
5.1	2	2.530	11	-0.2179749

This is different then the situation prior, because we are need to convert to long two different numerical variables that will be in the same long variable, but differentiated by two categorical variables (rather than one).

First, we create the matrix.

```
# Generate A Matrix
set.seed(123)
ar_year <- c(1995, 1997, 1999)
ar_vars <- c("wage_model", "quant_model", "wage_simu", "quant_simu")
mt_equi = matrix(rnorm(length(ar_year)*length(ar_vars)),
  nrow=length(ar_year), ncol=length(ar_vars))

# Name Matrix
rownames(mt_equi) <- ar_year
colnames(mt_equi) <- ar_vars

# to tibble
tb_equi <- as_tibble(mt_equi, rownames = "year")

# Print
print(mt_equi)
```

```
##      wage_model quant_model  wage_simu quant_simu
## 1995 -0.5604756  0.07050839  0.4609162 -0.4456620
## 1997 -0.2301775  0.12928774 -1.2650612  1.2240818
## 1999  1.5587083  1.71506499 -0.6868529  0.3598138
```

Wide table

year	wage_model	quant_model	wage_simu	quant_simu
1995	-0.5604756	0.0705084	0.4609162	-0.4456620
1997	-0.2301775	0.1292877	-1.2650612	1.2240818
1999	1.5587083	1.7150650	-0.6868529	0.3598138

Long table, Two Variables

year	variable	source	value
1995	wage	model	-0.5604756
1995	quant	model	0.0705084
1995	wage	simu	0.4609162
1995	quant	simu	-0.4456620
1997	wage	model	-0.2301775
1997	quant	model	0.1292877
1997	wage	simu	-1.2650612
1997	quant	simu	1.2240818
1999	wage	model	1.5587083
1999	quant	model	1.7150650
1999	wage	simu	-0.6868529
1999	quant	simu	0.3598138

```
# Display
kable(tb_equi, caption = "Wide table") %>% kable_styling_fc()
```

Second, we convert the table wide to long. We select columns that includes either wage or quant, see tidyselect Select variables that match a pattern for additional verbs for how to select variables.

```
# longer
tb_equi_long <- tb_equi %>%
  pivot_longer(cols = matches('wage|quant'),
               names_to = c('variable', 'source'),
               names_pattern = paste0("(.*)_(.*)"),
               values_to = "value")

# Display
kable(tb_equi_long, caption = "Long table, Two Variables") %>% kable_styling_fc()
```