Convert Table from Long to Wide with dplyr

Fan Wang

2022-07-16

Contents

1 Long Table to Wide Table

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 Compute Wide Table Cumulative Student Attendance based on Long Table Roster

There are N students in class, but only a subset of them attend class each day. If student id_i is in class on day Q, the teacher records on a sheet the date and the student ID. So if the student has been in class 10 times, the teacher has ten rows of recorded data for the student with two columns: column one is the student ID, and column two is the date on which the student was in class.

Suppose there were 50 students, who on average attended exactly 10 classes each during the semester, this means we have $10 \cdot 50$ rows of data, with differing numbers of rows for each student. This is shown as *df_panel_attend_date* generated below.

Now we want to generate a new dataframe, where each row is a date, and each column is a student. The values in the new dataframe shows, at the Q^{th} day (row), how many classes student i has attended so far. The following results is also in a REconTools Function. This is shown as *df_attend_cumu_by_day* generated below.

First, generate the raw data structure, *df_panel_attend_date*:

```
# Define
it_N <- 3
it_M <- 5
svr_id <- 'student_id'

# from : support/rand/fs_rand_draws.Rmd
set.seed(222)
df_panel_attend_date <- as_tibble(matrix(it_M, nrow=it_N, ncol=1)) %>%
  rowid_to_column(var = svr_id) %>%
  uncount(V1) %>%
  group_by(!!sym(svr_id)) %>% mutate(date = row_number()) %>%
  ungroup() %>%
```

```
  mutate(in_class = case_when(rnorm(n(),mean=0,sd=1) < 0 ~ 1, TRUE ~ 0)) %>%
  filter(in_class == 1) %>% select(!!sym(svr_id), date) %>%
  rename(date_in_class = date)

# Print
kable(df_panel_attend_date) %>%
  kable_styling_fc()
```

student_id	date_in_class
1	2
1	4
2	1
2	2
2	5
3	2
3	3
3	5

Second, we create a attendance column that is all 1. This is not useful for the long table, but useful for our conversion to wide.

```
# Define
svr_id <- 'student_id'
svr_date <- 'date_in_class'
st_idcol_prefix <- 'sid_'

# Generate cumulative enrollment counts by date
df_panel_attend_date_addone <- df_panel_attend_date %>% mutate(attended = 1)
kable(df_panel_attend_date_addone) %>%
  kable_styling_fc()
```

student_id	date_in_class	attended
1	2	1
1	4	1
2	1	1
2	2	1
2	5	1
3	2	1
3	3	1
3	5	1

Third, we convert the long table to wide. The unit of observation is student-day for the long table, and day for the wide table.

```
# Pivot Wide
df_panel_attend_date_wider <- df_panel_attend_date_addone %>%
  arrange(student_id) %>%
  pivot_wider(names_from = svr_id,
              values_from = attended)

# Sort and rename
# rename see: https://fanwangecon.github.io/R4Econ/amto/tibble/fs_tib_basics.html
ar_unique_ids <- sort(unique(df_panel_attend_date %>% pull(!!sym(svr_id))))
df_panel_attend_date_wider_sort <- df_panel_attend_date_wider %>%
```

```
    arrange(!!sym(svr_date)) %>%
    rename_at(vars(num_range('',ar_unique_ids))
              , list(~paste0(st_idcol_prefix, . , ''))
              )
kable(df_panel_attend_date_wider_sort) %>%
  kable_styling_fc()
```

date_in_class	sid_1	sid_2	sid_3
1	NA	1	NA
2	1	1	1
3	NA	NA	1
4	1	NA	NA
5	NA	1	1

Fourth, we could achieve what we have above by specifying more parameters in the *pivot_wider* function.

```
# Include name_prefix
df_panel_attend_date_wider_sort <- df_panel_attend_date_addone %>%
  arrange(student_id) %>%
  pivot_wider(id_cols = c("date_in_class"),
              names_from = svr_id,
              names_prefix = "sid_",
              values_from = attended) %>%
  arrange(date_in_class)
# Print
kable(df_panel_attend_date_wider_sort) %>%
  kable_styling_fc()
```

date_in_class	sid_1	sid_2	sid_3
1	NA	1	NA
2	1	1	1
3	NA	NA	1
4	1	NA	NA
5	NA	1	1

Fifth, sum across rows for each student (column) to get cumulative attendance for each student on each date.

```
# replace NA and cumusum again
# see: R4Econ/support/function/fs_func_multivar for renaming and replacing
df_attend_cumu_by_day <- df_panel_attend_date_wider_sort %>%
  mutate_at(vars(contains(st_idcol_prefix)), list(~replace_na(., 0))) %>%
  mutate_at(vars(contains(st_idcol_prefix)), list(~cumsum(.)))

kable(df_attend_cumu_by_day) %>%
  kable_styling_fc()
```

date_in_class	sid_1	sid_2	sid_3
1	0	1	0
2	1	2	1
3	1	2	2
4	2	2	2
5	2	3	3

Finally, the structure above is also a function in Fan's REconTools Package, here the function is tested:

```
# Parameters
df <- df_panel_attend_date
svr_id_i <- 'student_id'
svr_id_t <- 'date_in_class'
st_idcol_prefix <- 'sid_'

# Invoke Function
ls_df_rosterwide <- ff_panel_expand_longrosterwide(df, svr_id_t, svr_id_i, st_idcol_prefix)
df_roster_wide_func <- ls_df_rosterwide$df_roster_wide
df_roster_wide_cumu_func <- ls_df_rosterwide$df_roster_wide_cumu

# Print
print(df_roster_wide_func)
print(df_roster_wide_cumu_func)
```

1.2 Panel Long Attendance Roster and Score Card to Wide

In the prior example, at each date, we only had information about attendance, however, we might also know the exam score on each day when the student attends school. In the long table, this appears, in addition to *attended*, as an additional variable *score*. When we convert from long to wide, we will have 3 new columns for attendance and also 3 new columns for score. The 3 columns are for the three students, there will be five rows for the five days. Each row in the wide table is the attendance and score information for each day.

First, we add a random score column to the long dataframe created prior. Also add two other additional columns.

```
# Create score column
set.seed(123)
df_panel_attend_score_date <- df_panel_attend_date_addone %>%
  mutate(score = rnorm(dim(df_panel_attend_date_addone)[1], mean=70, sd=10)) %>%
  mutate(score = round(score, 2),
         other_var_1 = 1, other_var_2 = 2)

# Print
kable(df_panel_attend_score_date, caption="Attend and score info") %>%
  kable_styling_fc()
```

Attend and score info

student_id	date_in_class	attended	score	other_var_1	other_var_2
1	2	1	64.40	1	2
1	4	1	67.70	1	2
2	1	1	85.59	1	2
2	2	1	70.71	1	2
2	5	1	71.29	1	2
3	2	1	87.15	1	2
3	3	1	74.61	1	2
3	5	1	57.35	1	2

Second, convert both attended and score columns to wide at the same time. Note that we add "sid" in front of the index for each student. Note that *id_cols* picks the columns to keep in addition to the names_from and values_from columns. In this case, we are not keeping *other_var_1* and *other_var_2*.

```r
# Convert to wide
df_panel_attend_score_date_wide <- df_panel_attend_score_date %>%
  arrange(student_id) %>%
  pivot_wider(id_cols  = c("date_in_class"),
              names_from = svr_id,
              names_prefix = "sid",
              names_sep = "_",
              values_from = c(attended, score)) %>%
  arrange(date_in_class)

# Print
kable(df_panel_attend_score_date_wide, caption="Attend and score wide") %>%
  kable_styling_fc_wide()
```

Attend and score wide

date_in_class	attended_sid1	attended_sid2	attended_sid3	score_sid1	score_sid2	score_sid3
1	NA	1	NA	NA	85.59	NA
2	1	1	1	64.4	70.71	87.15
3	NA	NA	1	NA	NA	74.61
4	1	NA	NA	67.7	NA	NA
5	NA	1	1	NA	71.29	57.35