R Do Anything Function over Dataframe Subset and Stack Output Dataframes, (MxP by N) to (MxQ by N+Z-1)

Fan Wang

2020-05-27

Contents

1 (MxP by N) to (MxQ by N+Z)

Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools Package, R Code Examples Repository (bookdown site), or Intro Stats with R Repository (bookdown site).

There is a dataframe composed of M mini-dataframes. Group by a variable that identifies each unique sub-dataframe, and use the sub-dataframes with P rows as inputs to a function.

The function outputs Q *by* Z rows and columns of results, stack the results. The output file has MxQ rows and the Z columns of additional results should be appended.

1.1 Generate the MxP by N Dataframe

M Grouping characteristics, P rows for each group, and N Variables.

1. M are individuals
2. P are dates
3. A wage variable for individual wage at each date. And a savings varaible as well.

```r
# Define
it_M <- 3
it_P <- 5
svr_m <- 'group_m'
svr_mp <- 'info_mp'

# dataframe
set.seed(123)
df_panel_skeleton <- as_tibble(matrix(it_P, nrow=it_M, ncol=1)) %>%
  rowid_to_column(var = svr_m) %>%
  uncount(V1) %>%
  group_by(!!sym(svr_m)) %>% mutate(!!sym(svr_mp) := row_number()) %>%
  ungroup() %>%
  rowwise() %>% mutate(wage = rnorm(1, 100, 10),
                       savings = rnorm(1, 200, 30)) %>%
  ungroup() %>%
  rowid_to_column(var = "id_ji")

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

id_ji	group_m	info_mp	wage	savings
1	1	1	94.39524	253.6074
2	1	2	97.69823	214.9355
3	1	3	115.58708	141.0015
4	1	4	100.70508	221.0407
5	1	5	101.29288	185.8163
6	2	1	117.15065	167.9653
7	2	2	104.60916	193.4608
8	2	3	87.34939	169.2199
9	2	4	93.13147	178.1333
10	2	5	95.54338	181.2488
11	3	1	112.24082	149.3992
12	3	2	103.59814	225.1336
13	3	3	104.00771	204.6012
14	3	4	101.10683	165.8559
15	3	5	94.44159	237.6144

1.2 Subgroup Compute and Expand

Use the M sub-dataframes, generate Q by Z result for each of the M groups. Stack all results together.

Base on all the wages for each individual, generate individual specific mean and standard deviations. Do this for three things, the wage variable, the savings variable, and the sum of wage and savings:

1. $Z=2$: 2 columns, mean and standard deviation
2. $Q=3$: 3 rows, statistics based on wage, savings, and the sum of both

First, here is the processing function that takes the dataframe as input, with a parameter for rounding:

```
# define function
ffi_subset_mean_sd <- function(df_sub, it_round=1) {
  #' A function that generates mean and sd for several variables
  #'
  #' @description
  #' Assume there are two variables in df_sub wage and savings
  #'
  #' @param df_sub dataframe where each individual row is a different
  #' data point, over which we compute mean and sd, Assum there are two
  #' variables, savings and wage
  #' @param it_round integer rounding for resulting dataframe
  #' @return a dataframe where each row is aggregate for a different type
  #' of variablea and each column is a different statistics

  fl_wage_mn = mean(df_sub$wage)
  fl_wage_sd = sd(df_sub$wage)

  fl_save_mn = mean(df_sub$savings)
  fl_save_sd = sd(df_sub$savings)
```

```r
  fl_wgsv_mn = mean(df_sub$wage + df_sub$savings)
  fl_wgsv_sd = sd(df_sub$wage + df_sub$savings)

  ar_mn <- c(fl_wage_mn, fl_save_mn, fl_wgsv_mn)
  ar_sd <- c(fl_wage_sd, fl_save_sd, fl_wgsv_sd)
  ar_st_row_lab <- c('wage', 'savings', 'wage_and_savings')

  mt_stats <- cbind(ar_mn, ar_sd)
  mt_stats <- round(mt_stats, it_round)

  ar_st_varnames <- c('mean', 'sd', 'variables')
  df_combine <- as_tibble(mt_stats) %>%
    add_column(ar_st_row_lab) %>%
    rename_all(~c(ar_st_varnames)) %>%
    select(variables, 'mean', 'sd') %>%
    rowid_to_column(var = "id_q")

  return(df_combine)
}
# testing function
ffi_subset_mean_sd(df_panel_skeleton %>% filter(!!sym(svr_m)==1))
```

Second, call *ffi_subset_mean_sd* function for each of the groups indexed by j and stack results together with j index:

1. group by
2. call function
3. unnest

```r
# run group stats and stack dataframes
df_outputs <- df_panel_skeleton %>% group_by(!!sym(svr_m)) %>%
  do(df_stats = ffi_subset_mean_sd(., it_round=2)) %>%
  unnest() %>%
  rowid_to_column(var = "id_mq")
# print
kable(df_outputs) %>% kable_styling_fc()
```

id_mq	group_m	id_q	variables	mean	sd
1	1	1	wage	101.94	8.11
2	1	2	savings	203.28	42.33
3	1	3	wage_and_savings	305.22	34.83
4	2	1	wage	99.56	11.63
5	2	2	savings	178.01	10.34
6	2	3	wage_and_savings	277.56	15.48
7	3	1	wage	103.08	6.39
8	3	2	savings	196.52	37.86
9	3	3	wage_and_savings	299.60	33.50

In the resulting file, we went from a matrix with MxP rows to a matrix with MxQ Rows.