

Mesh Matrices, Arrays and Scalars Values of States, Choices, Shocks and Parameters in R Together

Fan Wang

2020-04-18

Contents

1	Mesh Matrices, Arrays and Scalars	1
1.1	Mesh Two or More Vectors with <code>expand_grid</code>	1
1.2	Mesh Arrays with <code>expand_grid</code>	2

1 Mesh Matrices, Arrays and Scalars

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](#)).

- `r.expand.grid` meshed array to matrix
- `r.meshgrid`
- `r.array` to matrix
- `r.reshape` array to matrix
- `dplyr` permutations rows of matrix and element of array
- `tidyr.expand_grid` mesh matrix and vector

1.1 Mesh Two or More Vectors with `expand_grid`

In the example below, we have a matrix that is 2 by 2 (endogenous states), a vector that is 3 by 1 (choices), and another matrix that is 4 by 3 (exogenous states shocks).

We want to generate a tibble dataset that meshes the matrix and the vector, so that all combinations show up. Additionally, we want to add some additional values that are common across all rows to the meshed dataframe.

Note `expand_grid` is a from `tidyr` 1.0.0.

```
# A. Generate the 5 by 2 Matrix (ENDO STATES)
# it_child_count = N, the number of children
it_N_child_cnt = 2
# P fixed parameters, nN is N dimensional, nP is P dimensional
ar_nN_A = seq(-2, 2, length.out = it_N_child_cnt)
ar_nN_alpha = seq(0.1, 0.9, length.out = it_N_child_cnt)
fl_rho = 0.1
fl_lambda = 1.1
mt_nP_A_alpha = cbind(ar_nN_A, ar_nN_alpha, fl_rho, fl_lambda)
ar_st_varnames <- c('s_A', 's_alpha', 'p_rho', 'p_lambda')
tb_states_endo <- as_tibble(mt_nP_A_alpha) %>%
  rename_all(~c(ar_st_varnames)) %>%
  rowid_to_column(var = "state_id")
```

```

# B. Choice Grid
it_N_choice_cnt = 3
fl_max = 10
fl_min = 0
ar_nN_d = seq(fl_min, fl_max, length.out = it_N_choice_cnt)
ar_st_varnames <- c('c_food')
tb_choices <- as_tibble(ar_nN_d) %>%
  rename_all(~c(ar_st_varnames)) %>%
  rowid_to_column(var = "choice_id")

# C. Shock Grid
set.seed(123)
it_N_shock_cnt = 4
ar_nQ_shocks = exp(rnorm(it_N_shock_cnt, mean=0, sd=1))
ar_st_varnames <- c('s_eps')
tb_states_exo <- as_tibble(ar_nQ_shocks) %>%
  rename_all(~c(ar_st_varnames)) %>%
  rowid_to_column(var = "shock_id")

# dataframe expand with other non expanded variables
ar_st_varnames <-
tb_states_shk_choices <- tb_states_endo %>%
  expand_grid(tb_choices) %>%
  expand_grid(tb_states_exo) %>%
  select(state_id, choice_id, shock_id,
         s_A, s_alpha, s_eps, c_food,
         p_rho, p_lambda)

# display
kable(tb_states_shk_choices) %>% kable_styling_fc()

```

Using `expand_grid` directly over arrays

```

# expand grid with dplyr
expand_grid(x = 1:3, y = 1:2, z = -3:-1)

```

1.2 Mesh Arrays with `expand.grid`

Given two arrays, mesh the two arrays together.

```

# use expand.grid to generate all combinations of two arrays

it_ar_A = 5
it_ar_alpha = 10

ar_A = seq(-2, 2, length.out=it_ar_A)
ar_alpha = seq(0.1, 0.9, length.out=it_ar_alpha)

mt_A_alpha = expand.grid(A = ar_A, alpha = ar_alpha)

mt_A_meshed = mt_A_alpha[,1]
dim(mt_A_meshed) = c(it_ar_A, it_ar_alpha)

mt_alpha_meshed = mt_A_alpha[,2]

```

state_id	choice_id	shock_id	s_A	s_alpha	s_eps	c_food	p_rho	p_lambda
1	1	1	-2	0.1	0.5709374	0	0.1	1.1
1	1	2	-2	0.1	0.7943926	0	0.1	1.1
1	1	3	-2	0.1	4.7526783	0	0.1	1.1
1	1	4	-2	0.1	1.0730536	0	0.1	1.1
1	2	1	-2	0.1	0.5709374	5	0.1	1.1
1	2	2	-2	0.1	0.7943926	5	0.1	1.1
1	2	3	-2	0.1	4.7526783	5	0.1	1.1
1	2	4	-2	0.1	1.0730536	5	0.1	1.1
1	3	1	-2	0.1	0.5709374	10	0.1	1.1
1	3	2	-2	0.1	0.7943926	10	0.1	1.1
1	3	3	-2	0.1	4.7526783	10	0.1	1.1
1	3	4	-2	0.1	1.0730536	10	0.1	1.1
2	1	1	2	0.9	0.5709374	0	0.1	1.1
2	1	2	2	0.9	0.7943926	0	0.1	1.1
2	1	3	2	0.9	4.7526783	0	0.1	1.1
2	1	4	2	0.9	1.0730536	0	0.1	1.1
2	2	1	2	0.9	0.5709374	5	0.1	1.1
2	2	2	2	0.9	0.7943926	5	0.1	1.1
2	2	3	2	0.9	4.7526783	5	0.1	1.1
2	2	4	2	0.9	1.0730536	5	0.1	1.1
2	3	1	2	0.9	0.5709374	10	0.1	1.1
2	3	2	2	0.9	0.7943926	10	0.1	1.1
2	3	3	2	0.9	4.7526783	10	0.1	1.1
2	3	4	2	0.9	1.0730536	10	0.1	1.1

```
dim(mt_alpha_meshed) = c(it_ar_A, it_ar_alpha)
```

```
# display
kable(mt_A_meshed) %>%
  kable_styling_fc()
```

-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2

```
kable(mt_alpha_meshed) %>%
  kable_styling_fc_wide()
```

0.1	0.1888889	0.2777778	0.3666667	0.4555556	0.5444444	0.6333333	0.7222222	0.8111111	0.9
0.1	0.1888889	0.2777778	0.3666667	0.4555556	0.5444444	0.6333333	0.7222222	0.8111111	0.9
0.1	0.1888889	0.2777778	0.3666667	0.4555556	0.5444444	0.6333333	0.7222222	0.8111111	0.9
0.1	0.1888889	0.2777778	0.3666667	0.4555556	0.5444444	0.6333333	0.7222222	0.8111111	0.9
0.1	0.1888889	0.2777778	0.3666667	0.4555556	0.5444444	0.6333333	0.7222222	0.8111111	0.9

Two Identical Arrays, individual attributes, each column is an individual for a matrix, and each row is also an individual.

```
# use expand.grid to generate all combinations of two arrays
```

```

it_ar_A = 5

ar_A = seq(-2, 2, length.out=it_ar_A)
mt_A_A = expand.grid(Arow = ar_A, Arow = ar_A)
mt_Arow = mt_A_A[,1]
dim(mt_Arow) = c(it_ar_A, it_ar_A)
mt_Acol = mt_A_A[,2]
dim(mt_Acol) = c(it_ar_A, it_ar_A)

# display
kable(mt_Arow) %>%
  kable_styling_fc()

```

-2	-2	-2	-2	-2
-1	-1	-1	-1	-1
0	0	0	0	0
1	1	1	1	1
2	2	2	2	2

```

kable(mt_Acol) %>%
  kable_styling_fc()

```

-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2