R dplyr Group by Index and Generate Panel Data Structure

Fan Wang

2020-06-13

Contents

1 Generate Panel Structure **1**
1.1 Balanced Panel Skeleton 1
1.2 Panel of Children with Height Growth 1
1.3 Create Group IDs 2

1 Generate Panel Structure

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).

1.1 Balanced Panel Skeleton

There are N individuals, each could be observed M times. In the example below, there are 3 students, each observed over 4 dates. This just uses the uncount function from *tidyr*.

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

# dataframe
df_panel_skeleton <- as_tibble(matrix(it_M, nrow=it_N, ncol=1)) %>%
  rowid_to_column(var = svr_id) %>%
  uncount(V1) %>%
  group_by(!!sym(svr_id)) %>% mutate(!!sym(svr_date) := row_number()) %>%
  ungroup()

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

1.2 Panel of Children with Height Growth

Given N individuals, each with G observations. There is an initial height variable and height grows every year. There are growth variables, variables for cumulative growth and variables for height at each age for each child.

Individuals are defined by gender (1 = female), race (1=asian), and birth height. Within individual yearly information includes height at each year of age.

student_id	class_day
1	1
1	2
1	3
1	4
1	5
2	1
2	2
2	3
2	4
2	5
3	1
3	2
3	3
3	4
3	5

```
# Define
it_N <- 5
it_M <- 3
svr_id <- 'indi_id'
svr_gender <- 'female'
svr_asian <- 'asian'
svr_age <- 'year_of_age'
# Define Height Related Variables
svr_brthgt <- 'birth_height'
svr_hgtgrow <- 'hgt_growth'
svr_hgtgrow_cumu <- 'hgt_growcumu'
svr_height <- 'height'

# panel dataframe following
set.seed(123)
df_panel_indiage <- as_tibble(matrix(it_M, nrow=it_N, ncol=1)) %>%
  mutate(!!sym(svr_gender) := rbinom(n(), 1, 0.5),
         !!sym(svr_asian) := rbinom(n(), 1, 0.5),
         !!sym(svr_brthgt) := rnorm(n(), mean=60,sd=3)) %>%
  uncount(V1) %>%
  group_by(!!sym(svr_gender), !!sym(svr_asian), !!sym(svr_brthgt)) %>%
  mutate(!!sym(svr_age) := row_number(),
         !!sym(svr_hgtgrow) := runif(n(), min=5, max=15),
         !!sym(svr_hgtgrow_cumu) := cumsum(!!sym(svr_hgtgrow)),
         !!sym(svr_height) := !!sym(svr_brthgt) + !!sym(svr_hgtgrow_cumu))  %>%
  ungroup()

# Add Height Index
kable(df_panel_indiage) %>% kable_styling_fc()
```

1.3 Create Group IDs

Given the dataframe just created, generate group IDs for each Gender and Race Groups. Given that both are binary, there can only be 4 unique groups.

female	asian	birth_height	year_of_age	hgt_growth	hgt_growcumu	height
0	0	65.14520	1	13.895393	13.895393	79.04059
0	0	65.14520	2	11.928034	25.823427	90.96862
0	0	65.14520	3	11.405068	37.228495	102.37369
1	1	61.38275	1	11.907053	11.907053	73.28980
1	1	61.38275	2	12.954674	24.861727	86.24448
1	1	61.38275	3	5.246137	30.107864	91.49061
0	1	56.20482	1	14.942698	14.942698	71.14751
0	1	56.20482	2	11.557058	26.499756	82.70457
0	1	56.20482	3	12.085305	38.585060	94.78988
1	1	57.93944	1	6.471137	6.471137	64.41058
1	1	57.93944	2	14.630242	21.101379	79.04082
1	1	57.93944	3	14.022991	35.124369	93.06381
1	0	58.66301	1	10.440660	10.440660	69.10367
1	0	58.66301	2	10.941420	21.382081	80.04509
1	0	58.66301	3	7.891597	29.273678	87.93669

```r
# group id
svr_group_id <- 'female_asian_id'
# Define
ls_svr_group_vars <- c('female', 'asian')

# panel dataframe following
df_panel_indiage_id <- df_panel_indiage %>%
  arrange(!!!syms(ls_svr_group_vars)) %>%
  group_by(!!!syms(ls_svr_group_vars)) %>%
  mutate(!!sym(svr_group_id) := (row_number()==1)*1) %>%
  ungroup() %>%
  mutate(!!sym(svr_group_id) := cumsum(!!sym(svr_group_id))) %>%
  select(one_of(svr_group_id, ls_svr_group_vars), everything())

# Add Height Index
kable(df_panel_indiage_id) %>%
  kable_styling_fc_wide()
```

female_asian_id	female	asian	birth_height	year_of_age	hgt_growth	hgt_growcumu	height
1	0	0	65.14520	1	13.895393	13.895393	79.04059
1	0	0	65.14520	2	11.928034	25.823427	90.96862
1	0	0	65.14520	3	11.405068	37.228495	102.37369
2	0	1	56.20482	1	14.942698	14.942698	71.14751
2	0	1	56.20482	2	11.557058	26.499756	82.70457
2	0	1	56.20482	3	12.085305	38.585060	94.78988
3	1	0	58.66301	1	10.440660	10.440660	69.10367
3	1	0	58.66301	2	10.941420	21.382081	80.04509
3	1	0	58.66301	3	7.891597	29.273678	87.93669
4	1	1	61.38275	1	11.907053	11.907053	73.28980
4	1	1	61.38275	2	12.954674	24.861727	86.24448
4	1	1	61.38275	3	5.246137	30.107864	91.49061
4	1	1	57.93944	1	6.471137	6.471137	64.41058
4	1	1	57.93944	2	14.630242	21.101379	79.04082
4	1	1	57.93944	3	14.022991	35.124369	93.06381