R Example Counting, Tabulation, and Cross Tabulation

Fan Wang

2020-04-01

Contents

1 Counting and Tabulations

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.

1.1 Tabulate Two Categorial Variables

First, we tabulate a dataset, and show categories as rows, and display frequencies.

```
# We use the mtcars dataset
tb_tab_joint <- mtcars %>%
    group_by(gear, am) %>%
    tally()
# Display
tb_tab_joint %>%
    kable(caption = "cross tabulation, stacked") %>%
    kable_styling_fc()
```

cross tabulation, stacked

gear	am	n
3	0	15
4	0	4
4	1	8
5	1	5

We can present this as cross tabs.

```
# We use the mtcars dataset
tb_cross_tab <- mtcars %>%
    group_by(gear, am) %>%
    tally() %>%
    spread(am, n)
# Display
tb_cross_tab %>%
```

```
    kable(caption = "cross tabulation") %>%
    kable_styling_fc()
```

cross tabulation

gear	0	1
3	15	NA
4	4	8
5	NA	5

1.2 Tabulate Once Each Distinct Subgroup

We have two variables variables, am and mpg, the mpg values are not unique. We want to know how many unique mpg levels are there for each am group. We use the dplyr::distinct function to achieve this.

```
tb_dist_tab <- mtcars %>%
    # .keep_all to keep all variables
    distinct(am, mpg, .keep_all = TRUE) %>%
    group_by(am) %>%
    tally()
# Display
tb_dist_tab %>%
    kable(caption = "Tabulate distinct groups") %>%
    kable_styling_fc()
```

Tabulate distinct groups

am	n
0	16
1	11

1.3 Expanding to Panel

There are N individuals, each observed for Y_i years. We start with a dataframe where individuals are the unit of observation, we expand this to a panel with a row for each of the years that the individual is in the survey for.

Algorithm:

1. generate testing frame, the individual attribute dataset with invariant information over panel
2. uncount, duplicate rows by years in survey
3. group and generate sorted index
4. add indiviual specific stat year to index

First, we construct the dataframe where each row is an individual.

```
# 1. Array of Years in the Survey
ar_years_in_survey <- c(2, 3, 1, 10, 2, 5)
ar_start_yaer <- c(1, 2, 3, 1, 1, 1)
ar_end_year <- c(2, 4, 3, 10, 2, 5)
mt_combine <- cbind(ar_years_in_survey, ar_start_yaer, ar_end_year)

# This is the individual attribute dataset, attributes that are invariant acrosss years
tb_indi_attributes <- as_tibble(mt_combine) %>% rowid_to_column(var = "ID")
```

```
# Display
tb_indi_attributes %>%
    head(10) %>%
    kable() %>%
    kable_styling_fc()
```

ID	ar_years_in_survey	ar_start_yaer	ar_end_year
1	2	1	2
2	3	2	4
3	1	3	3
4	10	1	10
5	2	1	2
6	5	1	5

Second, we change the dataframe so that each unit of observation is an individual in an year. This means we will duplicate the information in the prior table, so if an individual appears for 4 years in the survey, we will now have four rows for this individual. We generate a new variable that is the calendar year. This is now a panel dataset.

```
# 2. Sort and generate variable equal to sorted index
tb_indi_panel <- tb_indi_attributes %>% uncount(ar_years_in_survey)

# 3. Panel now construct exactly which year in survey, note that all needed is sort index
# Note sorting not needed, all rows identical now
tb_indi_panel <- tb_indi_panel %>%
    group_by(ID) %>%
    mutate(yr_in_survey = row_number())

tb_indi_panel <- tb_indi_panel %>%
    mutate(calendar_year = yr_in_survey + ar_start_yaer - 1)

# Show results Head 10
tb_indi_panel %>%
    head(10) %>%
    kable() %>%
    kable_styling_fc()
```

ID	ar_start_yaer	ar_end_year	yr_in_survey	calendar_year
1	1	2	1	1
1	1	2	2	2
2	2	4	1	2
2	2	4	2	3
2	2	4	3	4
3	3	3	1	3
4	1	10	1	1
4	1	10	2	2
4	1	10	3	3
4	1	10	4	4