

Simulate country-specific wage draws and compute country wage GINIs: Dataframe (Mx1 by N) to (MxQ by N+1) to (Mx1 by N)

Fan Wang

2022-07-16

Contents

1	(MxP by N) to (Mx1 by 1)	1
1.1	Wages from Many Countries and Country-specific GINI	1

1 (MxP by N) to (Mx1 by 1)

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 Wages from Many Countries and Country-specific GINI

There is a Panel with M individuals and each individual has Q records/rows. A function generate an individual specific outcome given the Q individual specific inputs, along with shared parameters/values stored as variables that contain common values for each of the M individuals.

For example, suppose we have a dataframe of individual wage information from different countries (the number of countries is M). Each row is an individual from one country, giving us $Q \cdot M$ observations of wages.

We want to generate country specific gini based on the individual wage data for each country in the dataframe. Additionally, perhaps the gini formula requires not just individual wages but some additional parameters or shared dataframes as inputs. We will use the [ff_dist_gini_vector_pos.html](#) function from [REconTools](#).

First, we simulate a dataframe with M countries, and up to Q people in each country. The countries share the same mean income, but have different standard deviations.

```
# Parameter Setups
it_M <- 10
it_Q_max <- 100
fl_rnorm_mu <- 1
ar_rnorm_sd <- seq(0.01, 0.2, length.out=it_M)
set.seed('789')
ar_it_q <- sample.int(it_Q_max, it_M, replace=TRUE)

# N by Q varying parameters
mt_data <- cbind(ar_it_q, ar_rnorm_sd)
tb_M <- as_tibble(mt_data) %>% rowid_to_column(var = "ID") %>%
  rename(sd = ar_rnorm_sd,
         Q = ar_it_q) %>%
  mutate(mean = fl_rnorm_mu) %>%
```

```

      select(ID, Q,
             mean, sd)

# Show table
kable(tb_M, caption = paste0("M=", it_M,
  " countries (ID is country ID), observation per country (Q)",
  ", mean and s.d. of wages each country")) %>%
kable_styling_fc()

```

M=10 countries (ID is country ID), observation per country (Q), mean and s.d. of wages each country

ID	Q	mean	sd
1	45	1	0.0100000
2	12	1	0.0311111
3	42	1	0.0522222
4	26	1	0.0733333
5	99	1	0.0944444
6	37	1	0.1155556
7	100	1	0.1366667
8	43	1	0.1577778
9	67	1	0.1788889
10	70	1	0.2000000

Second, we now expand the dataframe so that each country has not just one row, but Q_i of observations (i is country), or randomly drawn income based on the country-specific income distribution. Note that there are three ways of referring to variable names with dot, which are all shown below:

1. We can explicitly refer to names
2. We can use the [dollar dot structure](#) to use string variable names in do anything.
3. We can use dot bracket, this is the only option that works with string variable names

```

# A. Normal Draw Expansion, Explicitly Name
set.seed('123')
tb_income_norm_dot_dollar <- tb_M %>% group_by(ID) %>%
  do(income = rnorm(.$Q, mean=.$mean, sd=.$sd)) %>%
  unnest(c(income)) %>%
  left_join(tb_M, by="ID")

# Normal Draw Expansion again, dot dollar differently with string variable name
set.seed('123')
tb_income_norm_dollar_dot <- tb_M %>% group_by(ID) %>%
  do(income = rnorm(`.$`(. , 'Q'), mean = `.$`(. , 'mean'), sd = `.$`(. , 'sd')))) %>%
  unnest(c(income)) %>%
  left_join(tb_M, by="ID")

# Normal Draw Expansion again, dot double bracket
set.seed('123')
svr_mean <- 'mean'
svr_sd <- 'sd'
svr_Q <- 'Q'
tb_income_norm_dot_bracket_db <- tb_M %>% group_by(ID) %>%
  do(income = rnorm(.[[svr_Q]], mean = .[[svr_mean]], sd = .[[svr_sd]])) %>%
  unnest(c(income)) %>%
  left_join(tb_M, by="ID")

```

Third, we print the first set of rows of the dataframe, and also summarize income by country groups.

```
# Show dataframe dimension
print(dim(tb_income_norm_dot_bracket_db))
```

```
## [1] 541    5
```

```
# Show first 20 rows
kable(head(tb_income_norm_dot_bracket_db, 20),
  caption = "ID = country ID, wage draws"
) %>% kable_styling_fc()
```

ID = country ID, wage draws

ID	income	Q	mean	sd
1	0.9943952	45	1	0.01
1	0.9976982	45	1	0.01
1	1.0155871	45	1	0.01
1	1.0007051	45	1	0.01
1	1.0012929	45	1	0.01
1	1.0171506	45	1	0.01
1	1.0046092	45	1	0.01
1	0.9873494	45	1	0.01
1	0.9931315	45	1	0.01
1	0.9955434	45	1	0.01
1	1.0122408	45	1	0.01
1	1.0035981	45	1	0.01
1	1.0040077	45	1	0.01
1	1.0011068	45	1	0.01
1	0.9944416	45	1	0.01
1	1.0178691	45	1	0.01
1	1.0049785	45	1	0.01
1	0.9803338	45	1	0.01
1	1.0070136	45	1	0.01
1	0.9952721	45	1	0.01

```
# Display country-specific summaries
```

```
REconTools::ff_summ_bygroup(tb_income_norm_dot_bracket_db, c("ID"), "income")$df_table_grp_stats
```

Fourth, there is only one input for the gini function *ar_pos*. Note that the gini are not very large even with large SD, because these are normal distributions. By Construction, most people are in the middle. So with almost zero standard deviation, we have perfect equality, as standard deviation increases, inequality increases, but still pretty equal overall, there is no fat upper tail.

```
# Gini by Group
```

```
tb_gini_norm <- tb_income_norm_dot_bracket_db %>% group_by(ID) %>%
  do(inc_gini_norm = REconTools::ff_dist_gini_vector_pos(.$income)) %>%
  unnest(c(inc_gini_norm)) %>%
  left_join(tb_M, by="ID")
```

```
# display
```

```
kable(tb_gini_norm,
  caption = paste0(
    "Country-specific wage GINI based on income draws",
    ", ID=country-ID, Q=sample-size-per-country",
    ", mean=true-income-mean, sd=true-income-sd"
```

```
)) %>%
kable_styling_fc()
```

Country-specific wage GINI based on income draws, ID=country-ID, Q=sample-size-per-country, mean=true-income-mean, sd=true-income-sd

ID	inc_gini_norm	Q	mean	sd
1	0.0052111	45	1	0.0100000
2	0.0137174	12	1	0.0311111
3	0.0245939	42	1	0.0522222
4	0.0303468	26	1	0.0733333
5	0.0527628	99	1	0.0944444
6	0.0544053	37	1	0.1155556
7	0.0786986	100	1	0.1366667
8	0.0818873	43	1	0.1577778
9	0.1014639	67	1	0.1788889
10	0.0903825	70	1	0.2000000