

Cross-group Within-time and Cross-time Within-group Statistics

Fan Wang

2023-03-21

Contents

1	Within Panel Cross-time and Cross-group Statistics	1
1.1	Comparing Three Countries over Time	1

1 Within Panel Cross-time and Cross-group Statistics

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 Comparing Three Countries over Time

Given three time series for three “countries”, we compute percentage change from initial year for each country, and compare relative values within each timer period versus one country.

First, we generate the core data inputs. We assume that the output here would be the data structure we face prior to generating the figures we are interested in. We use data from the attitude dataset, but re-interpret what the columns are. We work with data from three “countries” at the same time, which generalizes also to the two countries case.

```
# Load data, and treat index as "year"
# pretend data to be country-data
df_attitude <- as_tibble(attitude) %>%
  rowid_to_column(var = "year") %>%
  select(year, rating, complaints, learning) %>%
  rename(stats_usa = rating,
         stats_canada = complaints,
         stats_uk = learning)

# Wide to Long
df_attitude <- df_attitude %>%
  pivot_longer(cols = starts_with('stats_'),
              names_to = c('country'),
              names_pattern = paste0("stats_(.*)"),
              values_to = "rating")

# Print
kable(df_attitude[1:10,]) %>% kable_styling_fc()
```

Second, we generate additional data inputs. Specifically, we also generate ratios of values with respect to the “first” country, within each year.

year	country	rating
1	usa	43
1	canada	51
1	uk	39
2	usa	63
2	canada	64
2	uk	54
3	usa	71
3	canada	70
3	uk	69
4	usa	61

year	country	rating	ratings_ratio_vs_countrycanada
1	canada	51	1.0000000
1	uk	39	0.7647059
1	usa	43	0.8431373
2	canada	64	1.0000000
2	uk	54	0.8437500
2	usa	63	0.9843750
3	canada	70	1.0000000
3	uk	69	0.9857143
3	usa	71	1.0142857
4	canada	63	1.0000000

```

# Sort and get list of countries
ar_countries_sorted <- df_attitude %>%
  ungroup() %>% distinct(country) %>% arrange(country) %>%
  pull(country)
st_ratio_var <- paste0('ratings_ratio_vs_country', ar_countries_sorted[1])

# Generate ratio over the first location
df_attitude <- df_attitude %>%
  arrange(year, country) %>% group_by(year) %>%
  mutate(!sym(st_ratio_var) := rating/first(rating))

# Print
kable(df_attitude[1:10,]) %>% kable_styling_fc()

```

Third, we now generate ratios of values with respect to the first year, within each country.

```

# Sort and get list of countries
ar_years_sorted <- df_attitude %>%
  ungroup() %>% distinct(year) %>% arrange(year) %>%
  pull(year)
st_ratio_var <- paste0('ratings_ratio_vs_year', ar_years_sorted[1])

# Generate ratio over the first location
df_attitude <- df_attitude %>%
  arrange(country, year) %>% group_by(country) %>%
  mutate(!sym(st_ratio_var) := rating/first(rating))

# Print
# Within each country, we show the first 3 years

```

year	country	rating	ratings_ratio_vs_countrycanada	ratings_ratio_vs_year1
1	canada	51	1.0000000	1.000000
2	canada	64	1.0000000	1.254902
3	canada	70	1.0000000	1.372549
1	uk	39	0.7647059	1.000000
2	uk	54	0.8437500	1.384615
3	uk	69	0.9857143	1.769231
1	usa	43	0.8431373	1.000000
2	usa	63	0.9843750	1.465116
3	usa	71	1.0142857	1.651163

```
kable(df_attitude %>%
  group_by(country) %>%
  slice_min(order_by = year, n = 3)
) %>% kable_styling_fc()
```