

R DPLYR Cummean Test, Cumulative Mean within Group

Fan Wang

2021-01-08

Contents

1 Cumulative Statistics within Group	1
1.1 Cumulative Mean	1

1 Cumulative Statistics within Group

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 Cumulative Mean

There is a dataset where there are different types of individuals, perhaps household size, that is the grouping variable. Within each group, we compute the incremental marginal propensity to consume for each additional check. We now also want to know the average propensity to consume up to each check considering all allocated checks. We needed to calculate this for [Nygaard, Sørensen and Wang \(2021\)](#). This can be dealt with by using the [cumall](#) function.

Use the `df_hgt_wgt` as the testing dataset. In the example below, group by individual id, sort by survey month, and cumulative mean over the protein variable.

In the protein example

First select the testing dataset and variables.

```
# Load the REconTools Dataset df_hgt_wgt
data("df_hgt_wgt")
# str(df_hgt_wgt)

# Select several rows
df_hgt_wgt_sel <- df_hgt_wgt %>%
  filter(S.country == "Cebu") %>%
  select(indi.id, svymthRound, prot)
```

Second, arrange, groupby, and cumulative mean. The protein variable is protein for each survey month, from month 2 to higher as babies grow. The protein intake observed is increasing quickly, hence, the cumulative mean is lower than the observed value for the survey month of the baby.

```
# Group by indi.id and sort by protein
df_hgt_wgt_sel_cummean <- df_hgt_wgt_sel %>%
  arrange(indi.id, svymthRound) %>%
  group_by(indi.id) %>%
  mutate(prot_cummean = cummean(prot))
```

```
# display results
REconTools::ff_summ_percentiles(df_hgt_wgt_sel_cummean)
# display results
df_hgt_wgt_sel_cummean %>% filter(indi.id %in% c(17, 18)) %>%
  kable() %>% kable_styling_fc()
```

indi.id	svymthRound	prot	prot_cummean
17	0	0.5	0.5000000
17	2	0.7	0.6000000
17	4	0.5	0.5666667
17	6	0.5	0.5500000
17	8	6.1	1.6600000
17	10	5.0	2.2166667
17	12	6.4	2.8142857
17	14	20.1	4.9750000
17	16	20.1	6.6555556
17	18	23.0	8.2900000
17	20	24.9	9.8000000
17	22	20.1	10.6583333
17	24	10.1	10.6153846
17	102	NA	NA
17	138	NA	NA
17	187	NA	NA
17	224	NA	NA
17	258	NA	NA
18	0	1.2	1.2000000
18	2	4.7	2.9500000
18	4	17.2	7.7000000
18	6	18.6	10.4250000
18	8	NA	NA
18	10	16.8	NA
18	12	NA	NA
18	14	NA	NA
18	16	NA	NA
18	18	NA	NA
18	20	NA	NA
18	22	15.7	NA
18	24	22.5	NA
18	102	NA	NA
18	138	NA	NA
18	187	NA	NA
18	224	NA	NA
18	258	NA	NA

Third, in the basic implementation above, if an incremental month has NA, no values computed at that point or after. This is the case for individual 18 above. To ignore NA, we have, from [this](#). Note how results for individual 18 changes.

```
# https://stackoverflow.com/a/49906718/8280804
# Group by indi.id and sort by protein
df_hgt_wgt_sel_cummean_noNA <- df_hgt_wgt_sel %>%
  arrange(indi.id, svymthRound) %>%
  group_by(indi.id, isna = is.na(prot)) %>%
```

```
mutate(prot_cummean = ifelse(isna, NA, cummean(prot)))

# display results
df_hgt_wgt_sel_cummean_noNA %>% filter(indi.id %in% c(17, 18)) %>%
  kable() %>% kable_styling_fc()
```

indi.id	svymthRound	prot	isna	prot_cummean
17	0	0.5	FALSE	0.5000000
17	2	0.7	FALSE	0.6000000
17	4	0.5	FALSE	0.5666667
17	6	0.5	FALSE	0.5500000
17	8	6.1	FALSE	1.6600000
17	10	5.0	FALSE	2.2166667
17	12	6.4	FALSE	2.8142857
17	14	20.1	FALSE	4.9750000
17	16	20.1	FALSE	6.6555556
17	18	23.0	FALSE	8.2900000
17	20	24.9	FALSE	9.8000000
17	22	20.1	FALSE	10.6583333
17	24	10.1	FALSE	10.6153846
17	102	NA	TRUE	NA
17	138	NA	TRUE	NA
17	187	NA	TRUE	NA
17	224	NA	TRUE	NA
17	258	NA	TRUE	NA
18	0	1.2	FALSE	1.2000000
18	2	4.7	FALSE	2.9500000
18	4	17.2	FALSE	7.7000000
18	6	18.6	FALSE	10.4250000
18	8	NA	TRUE	NA
18	10	16.8	FALSE	11.7000000
18	12	NA	TRUE	NA
18	14	NA	TRUE	NA
18	16	NA	TRUE	NA
18	18	NA	TRUE	NA
18	20	NA	TRUE	NA
18	22	15.7	FALSE	12.3666667
18	24	22.5	FALSE	13.8142857
18	102	NA	TRUE	NA
18	138	NA	TRUE	NA
18	187	NA	TRUE	NA
18	224	NA	TRUE	NA
18	258	NA	TRUE	NA