

R Summary By Groups, One Variable All Statistics

Fan Wang

2020-04-01

Contents

| | |
|--------------------------------------|---|
| One Variable Group Summary | 1 |
|--------------------------------------|---|

One Variable Group Summary

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](#)).

There is a categorical variable (based on one or the interaction of multiple variables), there is a continuous variable, obtain statistics for the continuous variable conditional on the categorical variable, but also unconditionally.

Store results in a matrix, but also flatten results wide to row with appropriate keys/variable-names for all group statistics.

Pick which statistics to be included in final wide row

```
# Single Variable Group Statistics (also generate overall statistics)
ff_summ_by_group_summ_one <- function(
  df, vars.group, var.numeric, str.stats.group = 'main',
  str.stats.specify = NULL, boo.overall.stats = TRUE){

  # List of statistics
  # https://rdrr.io/cran/dplyr/man/summarise.html
  str.center <- c('mean', 'median')
  str.spread <- c('sd', 'IQR', 'mad')
  str.range <- c('min', 'max')
  str.pos <- c('first', 'last')
  str.count <- c('n_distinct')

  # Grouping of Statistics
  if (missing(str.stats.specify)) {
    if (str.stats.group == 'main') {
      str.all <- c('mean', 'min', 'max', 'sd')
    }
    if (str.stats.group == 'all') {
      str.all <- c(str.center, str.spread, str.range, str.pos, str.count)
    }
  } else {
    str.all <- str.stats.specify
  }
}
```

```

# Start Transform
df <- df %>% drop_na() %>%
  mutate(!!(var.numeric) := as.numeric(!sym(var.numeric)))

# Overall Statistics
if (boo.overall.stats) {
  df.overall.stats <- df %>%
    summarize_at(vars(var.numeric), funs(!!!strs.all))
  if (length(strs.all) == 1) {
    # give it a name, otherwise if only one stat, name of stat not saved
    df.overall.stats <- df.overall.stats %>%
      rename(!strs.all := !sym(var.numeric))
  }
  names(df.overall.stats) <-
    paste0(var.numeric, '.', names(df.overall.stats))
}

# Group Sort
df.select <- df %>%
  group_by(!!!syms(vars.group)) %>%
  arrange(!!!syms(c(vars.group, var.numeric)))

# Table of Statistics
df.table.grp.stats <- df.select %>%
  summarize_at(vars(var.numeric), funs(!!!strs.all))

# Add Stat Name
if (length(strs.all) == 1) {
  # give it a name, otherwise if only one stat, name of stat not saved
  df.table.grp.stats <- df.table.grp.stats %>%
    rename(!strs.all := !sym(var.numeric))
}

# Row of Statistics
str.vars.group.combine <- paste0(vars.group, collapse='_')
if (length(vars.group) == 1) {
  df.row.grp.stats <- df.table.grp.stats %>%
    mutate(!!(str.vars.group.combine) :=
      paste0(var.numeric, '.',
              vars.group, '.g',
              (!!!syms(vars.group)))) %>%
    gather(variable, value, -one_of(vars.group)) %>%
    unite(str.vars.group.combine, c(str.vars.group.combine, 'variable')) %>%
    spread(str.vars.group.combine, value)
} else {
  df.row.grp.stats <- df.table.grp.stats %>%
    mutate(vars.groups.combine :=
      paste0(paste0(vars.group, collapse='.'),
              !!(str.vars.group.combine) :=
                paste0(interaction(!!!syms(vars.group)))) %>%
    mutate(!!(str.vars.group.combine) :=
      paste0(var.numeric, '.', vars.groups.combine, '.',

```

```

        (!!sym(str.vars.group.combine))) %>%
    ungroup() %>%
    select(-vars.groups.combine, -one_of(vars.group)) %>%
    gather(variable, value, -one_of(str.vars.group.combine)) %>%
    unite(str.vars.group.combine, c(str.vars.group.combine, 'variable')) %>%
    spread(str.vars.group.combine, value)
}

# Clean up name strings
names(df.table.grp.stats) <-
  gsub(x = names(df.table.grp.stats), pattern = "_", replacement = "\\.")
names(df.row.grp.stats) <-
  gsub(x = names(df.row.grp.stats), pattern = "_", replacement = "\\.")

# Return
list.return <-
  list(df_table_grp_stats = df.table.grp.stats,
        df_row_grp_stats = df.row.grp.stats)

# Overall Statistics, without grouping
if (boo.overall.stats) {
  df.row.stats.all <- c(df.row.grp.stats, df.overall.stats)
  list.return <- append(list.return,
                        list(df_overall_stats = df.overall.stats,
                              df_row_stats_all = df.row.stats.all))
}

# Return
return(list.return)
}

```

Build Program

Test Load data and test

```

# Library
library(tidyverse)

# Load Sample Data
setwd('C:/Users/fan/R4Econ/_data/')
df <- read_csv('height_weight.csv')

```

Function Testing By Gender Groups Need two variables, a group variable that is a factor, and a numeric

```

vars.group <- 'sex'
var.numeric <- 'hgt'

df.select <- df %>% select(one_of(vars.group, var.numeric)) %>% drop_na()

```

Main Statistics:

```

# Single Variable Group Statistics
ff_summ_by_group_summ_one(

```

```
df.select, vars.group = vars.group, var.numeric = var.numeric,
str.stats.group = 'main')$df_table_grp_stats
```

Specify Two Specific Statistics:

```
ff_summ_by_group_summ_one(
  df.select, vars.group = vars.group, var.numeric = var.numeric,
  str.stats.specify = c('mean', 'sd'))$df_table_grp_stats
```

Specify One Specific Statistics:

```
ff_summ_by_group_summ_one(
  df.select, vars.group = vars.group, var.numeric = var.numeric,
  str.stats.specify = c('mean'))$df_table_grp_stats
```

Function Testing By Country and Gender Groups Need two variables, a group variable that is a factor, and a numeric. Now joint grouping variables.

```
vars.group <- c('S.country', 'sex')
var.numeric <- 'hgt'
```

```
df.select <- df %>% select(one_of(vars.group, var.numeric)) %>% drop_na()
```

Main Statistics:

```
ff_summ_by_group_summ_one(
  df.select, vars.group = vars.group, var.numeric = var.numeric,
  str.stats.group = 'main')$df_table_grp_stats
```

Specify Two Specific Statistics:

```
ff_summ_by_group_summ_one(
  df.select, vars.group = vars.group, var.numeric = var.numeric,
  str.stats.specify = c('mean', 'sd'))$df_table_grp_stats
```

Specify One Specific Statistics:

```
ff_summ_by_group_summ_one(
  df.select, vars.group = vars.group, var.numeric = var.numeric,
  str.stats.specify = c('mean'))$df_table_grp_stats
```