Percentile-Based Grouping in SCF Analysis

Introducing the Percentile Cut Function

Joseph N. Cohen
Department of Sociology
Queens College, City University of New York
65-30 Kissena Blvd., Queens, NY 11367
joseph.cohen@qc.cuny.edu
https://jncohen.commons.gc.cuny.edu
ORCID: 0000-0002-6197-4453

May 27, 2026

Abstract

This note examines the task of summarizing statistics by percentile-defined groups in Survey of Consumer Finances (SCF) microdata. Percentile-based grouping is a commonplace analytical operation, but its use in SCF analysis is complicated by the data's complex design. This note reviews three methods for making such comparisons: a naive threshold approach, implicate-wise estimation using Rubin's Rules, or the Federal Reserve's official "stacking" method. The function `scf::scf_pctile_sum()` offers a structured and transparent means for comparisons by percentile-based grouping variables in SCF analysis. The note demonstrates the function by comparing top-decile mean net worth estimates across SCF survey years to Federal Reserve published estimates and to the Federal Reserve-provided `nwcat` grouping variable.

Keywords: Survey of Consumer Finances, multiple imputation, missing data, complex survey design, percentiles, Rstats

Acknowledgements

Thank you to the correspondent whose feedback created the occasion for this new function.

Introduction

This technical note discusses the task of summarizing statistics by percentile groups in analyses of the *Survey of Consumer Finances* (SCF) public-use microdata (U.S. Federal Reserve Bank 2023b). Examples of such an operation might be discerning health insurance coverage rates in the bottom 25% of household income, or the mean financial investments among those in the top 1% of net worth. The Survey's complex design and use of multiple imputation to handle missing data complicate this task. An analyst can make different choices about how percentile-group membership is operationalized and estimated, each of which produces similar but not identical results.

Below, I describe three methods for calculating statistics across percentile-defined groups in SCF data: a "commonsense" approach, the Federal Reserve's official "stacking" method, and the implicate-wise method suggested by Rubin (1987). In this exercise, the three methods render substantively similar results. The commonsense approach renders slightly imprecise decile groupings, and Rubin's Rules will render slightly different results than the stacking method.

In prior iterations of the `scf` package, users had to write their own code to implement percentile-based grouping procedures that respected the SCF's multiple-imputation and weighting structure (Cohen 2026b). Here, we introduce the new function `scf_pctile_sum()`, which automates these operations. The function allows users to choose between an implicate-wise workflow for ordinary analysis and a stacked workflow for reproducing the Federal Reserve's percentile-category convention.

Background

A Federal Reserve-published chartbook estimates that, in 2022, the mean net worth among the top 10% was $7,771,300 (rounded to nearest hundreds) (U.S. Federal Reserve Bank 2023a). In 1989, this figure was $2,926,600[1]. Both figures represent statistical estimates based on percentile groupings.[2] How to reproduce this finding?

The SCF is a complex survey that uses multiple imputation to deal with missing values. As such, its wrangling and analysis requires an array of functions that correctly handle this nonstandard data (see Cohen 2026a). Standard R functions that do not account for the data's complicated structure and analytical demands will produce incorrect estimates. The R package `scf` provides a comprehensive set of tools that correctly handle data.

[1] in inflation-adjusted 2022 dollars

[2] Note that these cited figures were produced with the SCF's publicly-available microdata, a privacy-conscious version of its internal-use data. As such, they may differ slightly and non-substantively from official estimates derived from the internal set.

Commonsense Approach

A commonsense strategy for producing estimates is through three steps: (1) estimate the 90th percentile values of net worth for the population, (2) identify households in the top decile if their net worth surpasses that threshold, and (3) estimate the mean net worth of the so-identified top decile group. The chunk below exemplifies that approach:

```
# Load the data
scf2022 <- scf_load(2022)

# scf_percentile() to discern percentile values:
temp_90p <- scf_percentile(scf2022, ~networth, q = 0.9)

# The `scf` functions return lists with multiple points of information.
# The final results are stored under the list item `results`:
threshold <- temp_90p$results$estimate

# Create a variable demarcating a household as having a
# net worth that surpasses threshold
scf2022 <- scf_update(scf2022, P90_networth = networth > threshold)

# Calculate mean by group
scf_mean(scf2022, ~networth, ~P90_networth)
```

```
## Multiply-Imputed, Replicate-Weighted Mean Estimate
##
##  group variable  estimate         se     min       max
##  FALSE networth  312092.7   9102.745  306631  317064.5
##   TRUE networth 7733378.0 385909.468 7335566 7940556.7
```

This gives us a result of $7,733,378. This is close to the rounded official estimate of $7,771,300, but it does not replicate it. This failure to replicate is the product of two issues. First, the commonsense method is not best practice for estimating parameters across percentile-defined groups when using multiply imputed data. Second, the method does not match SCF protocol for estimating percentile scores, which slightly diverges from the approach recommended in canonical missing data literature (Little and Rubin 1987; Rubin 1987).

Within-Implicate Method

One issue is that we used our global estimate of the 90th percentile net worth value to set our cutoffs. This is our estimate of the population-level 90th percentile net worth value after having processed each of the imputed sets and combined them to generate a global estimate. Each individual implicate has its own 90th percentile estimate, which you can see in the object created by the function `scf_percentile()`:

```
summary(temp_90p)
```

```
## Summary of SCF Percentile Estimate
##
## Pooled Estimates:
##  variable quantile   estimate        se        min        max
##  networth     0.90 1920758.00 138689.44 1809170.00 2079000.00
##
## Implicate-Level Estimates:
##  implicate group quantile   estimate        se
##          1   All     0.90 1877000.00  75589.97
##          2   All     0.90 2079000.00 203738.15
##          3   All     0.90 1866120.00  70992.55
##          4   All     0.90 1972500.00  81655.12
##          5   All     0.90 1809170.00 127701.60
```

The population-level 90th percentile estimate ultimately settled on $1,920,758, but that cutoff will not get you the top 10% of each implicate. For example, implicate #1 has an estimated 90th percentile wealth value of $1,877,000, so you will exclude observations that are above that number but not greater than the global estimate. The differences are extremely narrow, such that between 9.74% and 10.81% of individual implicates are scored as "top 10%" in our estimates using this approach. It will likely not lead to substantive differences, but will lead to estimates that do not fully replicate those generated by best practice.

Calculating Percentiles by Implicate

A standard imputation-aware procedure is to perform the complete-data analysis separately on each implicate and then pool the resulting estimates. So we are supposed to calculate the mean net worth for the top 10 percent of each implicate separately, and then pool those five estimates using Rubin's (1987) rules. The threshold is treated as an intermediate quantity internal to each implicate's analysis, not as a single threshold value applied to each individual implicate. The earlier versions (<=1.0.6) of the `scf` package have the function `scf_update_by_implicate()` which allows users to generate variables on an implicate-by-

implicate basis, such that we can create a variable that properly demarcates the top 10% scores for each implicate.

```r
# Function to demarcate the top 10% scores within each implicate
scf2022 <- scf_update_by_implicate(scf2022, function(df) {

  # Create array of replicate weight column names
  rep_cols <- grep("^wt1b", names(df), value = TRUE)

  # Create an implicate-specific survey design object
  design <- survey::svrepdesign(
    weights = ~wgt,
    repweights = as.matrix(df[, rep_cols]),
    data = df,
    type = "other",
    scale = 1,
    rscales = rep(1 / (length(rep_cols) - 1), length(rep_cols)),
    mse = TRUE,
    combined.weights = TRUE
  )

  # User `survey::svyquantile()` to calculate the 90th percentile
  # value for this implicate, and capture its value as a numeric object
  p90 <- as.numeric(coef(
    survey::svyquantile(~networth, design, quantiles = 0.9, se = FALSE)
  ))

  # Create a variable that demarcates the top 10% of this implicate
  df$top10nw <- df$networth >= p90
  df
})

# scf_mean will apply to each implicate with the correctly-identified
# top decile
scf_mean(scf2022, ~networth, ~top10nw)
```

```
## Multiply-Imputed, Replicate-Weighted Mean Estimate
##
##  group variable  estimate        se       min       max
##  FALSE networth  313585.3  15044.92  302993.7  332144.9
##   TRUE networth 7766387.1 289339.74 7680548.0 7852333.2
```

The implicate-wise method results in an estimate of $7,766,387. This is close to, but distinct from, the published estimate because it targets a different workflow than the Federal Reserve's stacked percentile-category convention. The reason is that the SCF's official estimates are not generated by calculating the 90th percentile value for each implicate and then pooling those values. Instead of using Rubin's Rules, the SCF's macro stacks all five implicates, divides each row's weight by five, sorts the stacked file by the target variable and household identifier, computes the cumulative weighted population share, and assigns percentile categories directly from that cumulative share. Published percentile-group summaries are then calculated over these assigned categories. This procedure produces estimates that are substantively very similar to those generated by the implicate-wise workflow.

Introducing `scf_pctile_sum()`

The above operations require functional programming. The function `scf_pctile_sum()` automates percentile-based grouping in SCF data. With `method = "implicate"`, it computes survey-weighted quantile thresholds within each implicate using `svyquantile()` from Lumley's `survey` package (Lumley et al. 2026), classifies households within implicates, and pools summary estimates using the package's multiple-imputation workflow. With `method = "stack"`, it follows the Federal Reserve convention: the five implicates are stacked, weights are divided by five, observations are sorted by the percentile variable and household identifier, percentile groups are assigned from the cumulative weighted population share, and flat weighted statistics are computed on the stacked data.

```
scf2022 <- scf_load(2022)

scf_pctile_sum(scf2022, ~networth,
               probs    = c(0, 0.9, 1),
               labels   = c("bottom90", "top10"),
               method   = "implicate",
               stat     = "mean",
               stat_var = ~networth)
```

```
## Multiply-Imputed, Replicate-Weighted Mean Estimate
##
##     group variable  estimate        se       min       max
##  bottom90 networth  313585.3  15044.92  302993.7  332144.9
##     top10 networth 7766387.1 289339.74 7680548.0 7852333.2
```

The resulting $7,766,387 replicates the estimate using the implicate-wise method (above), but not the official estimate. As noted, these do not replicate due to differences between how

percentile values are estimated by the SCF. To reproduce official estimates, use the "stacked" method:

```
scf_pctile_sum(scf2022, ~networth,
               probs    = c(0, 0.9, 1),
               labels   = c("bottom90", "top10"),
               method   = "stack",
               stat     = "mean",
               stat_var = ~networth)
```

```
##      group variable  estimate
## 1 bottom90 networth  313517.2
## 2    top10 networth 7771163.1
```

The result is approximately $7,771,160, or $7,771.16K in the table below. This is within about $0.13K of the published value. I repeat the exercise with the 1989 data, where the estimate matches the published value to the displayed precision used here:

```
scf1989 <- scf_load(1989)

scf_pctile_sum(scf1989, ~networth,
               probs    = c(0, 0.9, 1),
               labels   = c("bottom90", "top10"),
               method   = "stack",
               stat     = "mean",
               stat_var = ~networth)
```

```
##      group variable  estimate
## 1 bottom90 networth  159372.9
## 2    top10 networth 2926547.5
```

The result is $2,926,547.50, or $2,926.55K, matching the published value to the displayed precision used here. Table 1 compares the `scf_pctile_sum()` stacked estimates to published Federal Reserve values across SCF survey years. It also compares those estimates to means computed directly from the Federal Reserve-provided `nwcat` variable. This separates percentile-category assignment from upstream differences in public-use data values, source precision, or official-public processing.

[Insert Table 1 here]

The revised stack implementation reproduces the Federal Reserve percentile-category assignment procedure in the public-use data. In years with small residual differences from published values, the same discrepancies appear when using the Fed-provided `nwcat` variable directly. This indicates that the remaining differences do not arise from `scf_pctile_sum()`'s percentile-grouping algorithm. They more likely reflect upstream differences in public-use data values, source precision, rounding, or official-public processing.

Summary

This technical note discusses the task of summarizing statistics by percentile-defined groups in multiply imputed *Survey of Consumer Finances* microdata. It describes three approaches: a commonsense threshold method, implicate-wise estimation, and the Federal Reserve's official stacking method.

The function `scf_pctile_sum()` provides two clearly distinguished workflows: an implicate-wise, imputation-aware approach for ordinary analysis, and a stacked approach for reproducing Federal Reserve percentile-group point estimates. Tests show that the stacked method reproduces the Federal Reserve percentile-category assignment procedure in the public-use data and closely approximates published top-decile mean net worth estimates across survey years.

References

Cohen, Joseph Nathan. 2026a. "Analyzing the Survey of Consumer Finances with Scf." Accessed March 20, 2026. https://academicworks.cuny.edu/qc_pubs/695/.

Cohen, Joseph Nathan. 2026b. *Scf: Analyzing the Survey of Consumer Finances*. Version 1.0.9. [Computer software]. https://cran.r-project.org/web/packages/scf/index.html.

Little, Roderick JA, and Donald B. Rubin. 1987. *Statistical Analysis with Missing Data*. Vol. 793. John Wiley & Sons.

Lumley, Thomas, Peter Gao, Ben Schneider, and Stas Kolenkikov. 2026. *Survey: Analysis of Complex Survey Samples*. Version 4.5. [Computer software]. https://cran.r-project.org/web/packages/survey/index.html.

Rubin, Donald B. 1987. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons.

U.S. Federal Reserve Bank. 2023a. *Estimates in Nominal Dollars (3 MB Excel)*. [Data set]. https://www.federalreserve.gov/econres/files/scf2022_tables_public_nominal_historical.xlsx.

U.S. Federal Reserve Bank. 2023b. *Survey of Consumer Finances*. [Survey Data]. https://doi.org/10.17016/8799.

Table 1: Comparison of Federal Reserve Published Estimates, scf Percentile-Group Estimates, and Fed-Provided nwcat Estimates for Mean Net Worth Among the Top 10 Percent of Households

Year	Official	‘scf_pctile_sum()‘	‘nwcat‘	‘scf‘ difference	‘nwcat‘ difference
1989	$2,926.55K	$2,926.55K	$53.37K	$0.00K	$2,873.18K
1992	$2,595.59K	$2,595.59K	$52.77K	$0.00K	$2,542.82K
1995	$2,801.32K	$2,801.32K	$59.28K	$0.00K	$2,742.04K
1998	$3,545.41K	$3,545.41K	$65.57K	$0.00K	$3,479.84K
2001	$4,629.55K	$4,629.55K	$74.43K	$0.00K	$4,555.12K
2004	$4,900.37K	$4,900.29K	$74.11K	$0.08K	$4,826.26K
2007	$5,688.07K	$5,687.94K	$83.18K	$0.13K	$5,604.89K
2010	$5,028.69K	$5,026.44K	$48.66K	$2.25K	$4,980.03K
2013	$5,043.52K	$5,043.50K	$45.65K	$0.02K	$4,997.87K
2016	$6,550.28K	$6,550.24K	$55.09K	$0.04K	$6,495.19K
2019	$6,619.46K	$6,619.35K	$67.44K	$0.11K	$6,552.02K
2022	$7,771.29K	$7,771.16K	$98.84K	$0.13K	$7,672.45K