Analyzing the Survey of Consumer Finances: Examining Analytical Methods

Joseph Cohen
Department of Sociology
City University of New York
February 27, 2024

This post replicates and assesses *R-based* methods for analyzing data from the *Survey of Consumer Finances*. It assesses population estimates of household net worth quartiles using methods to account for the data's complex survey design and multiple imputation.[1] The note is intended to walk reader's through the process of obtaining population percentile score estimates from this data using officially-sanctioned scripts. In addition to the R code that is conventionally used, I generate estimates using code that I would conventionally apply in such a context to see if it replicates results in published reports or those obtained using recommended scripts.

I compare reported values from Federal Reserve-published reports (*Aladangady et al. (2023)* with those obtained trhough two methods (1) the data *documentation's recommended R scripts* by *Anthony Damico*[2] and those that implemented based on my own reading of best practices. Although the estimates obtained using Damico's scripts do differ from Federal Reserve-published figures, they do replicate the results that I obtained by my own efforts to employ standard best practice.

My own conclusion is that, although there can be discrepancies between the Federal Reserve-published figures and those obtained using Fed-endorsed analytical techniques, the discrepancies are small to the point of non-substantive and are readily explainable by the subjective judgment element of statistical analysis that non-practitioners often do not understand is part of good practice. My assessment concludes that the Damico scripts follow best practice and render quality results.

Background

The *Survey of Consumer Finances (SCF)* is a high quality, nationally-representative survey of U.S. household finances published by the United States Federal Reserve. The survey has been

[1] *Net worth* is the money value of one's personal assets, less that of their debts.

[2] Mr. Damico's repository of R scripts to analyze major surveys has been a boon to our graduate students at Queens College, and has been of great help to me as well. It is one of those projects in which someone makes a very big contribution to the research community, but does so in a way that is not registered in the academy's formal bookkeeping mechanisms. Thank you, Mr. Damico.

collected for decades, but its modern incarnation has run triennially since 1989. Its data can be an invaluable information source for learning about the income, expenditures, assets, debts, and global financial situation of U.S. households. Such information can help inform planning and decision-making in fields where households' financial situation is germane to decision-making, such as contemplating government policies (e.g., as in Cohen 2017), assessing marketing or human resource strategies, and a range of other applications.

The data is delivered in a complex structure that defies a simple and direct application of data analysis methods. The data was collected using a complex sampling scheme and deployed using a missing data imputation scheme for which the analyst must account in management and processing. Below, I detail the specifics of these considerations. I recommend Heeringa *et al.* (2017) for a general introduction to the analysis of complex survey data, and Lumley (2011) for the implementation of these methods in R. I recommend Allison (2010) for an accessible introduction to the basics of analyzing data with missing values, Carpenter *et al.* (2023)for a more advanced one, and Little and Rubin (2019) as a canonical text in this field.

Data and Methods

This section notes the practicalities of accessing, preparing, and analyzing these data.

```
# Clear the memory
rm(list=ls())
gc()

# Set the directory
root_directory <- "D:/Dropbox/Research/Household Finance"
directory <- paste0(root_directory, "/Net Worth Replication")
setwd(directory)

# Set seed
set.seed(123)

# Packages for this script:
packages <- c("httr", "jsonlite", "haven", "stringr", "survey", "dplyr", "scales",
              "ggplot2", "knitr", "kableExtra", "scales", "mitools", "rstanarm")
lapply(packages, library, character.only = TRUE)
rm(packages)

#Turn off scientific notation
options(sciepen = 999)
```

Accessing the Data

Each year's data is distributed publicly via the Internet. It is delivered as three tables. The first "main" table with the data collected in the survey. The second "summary" table has variables derived form the main table. For example, the table contains a *net worth* estimate that is calculated from balance sheet items in the main data table. The third set is a set of replicate weights designed to reweight observations to mitigate the effects of sample bias while retaining respondent anonymity.

```
# PART ONE: DOWNLOAD DATA
# Download Main 2022 Files
response_0 <- GET("https://www.federalreserve.gov/econres/files/scf2022s.zip",
write_disk("scf2022s.zip", overwrite = TRUE))
response_1 <- GET("https://www.federalreserve.gov/econres/files/scfp2022s.zip",
write_disk("scfp2022s.zip", overwrite = TRUE))
response_2 <- GET("https://www.federalreserve.gov/econres/files/scf2022rw1s.zip",
write_disk("scf2022rw.zip", overwrite = TRUE))

rm(list=ls(pattern = "response")) # Clean up objects


# PART TWO: UNZIP DATA
unzip("scf2022s.zip")
unzip("scfp2022s.zip")
unzip("scf2022rw.zip")

file.remove("scf2022s.zip")  # Erase the zip files because I don't need them and they
take up space.
file.remove("scfp2022s.zip")
file.remove("scf2022rw.zip")


# PART THREE:  CONVERT DATA TO R FORMAT
# The data are distributed in Stata format.  Below, I import data from its Stata
format.
scf2022 <- read_dta("p22i6.dta")
scf2022s <- read_dta("rscfp2022.dta")
scf2022rw <- read_dta("p22_rw1.dta")
```

Structure of the Data

The structure of this data is complex due to sample correction, anonymization, and missing data imputation methods. The main and summary data tables in the 2022 data include five imputations for each of the 4,595 households represented. It also contains a table of replicate weights that correspond to each of the households studied in this set. A *household* is a living arrangement in which an economically dominant individual or couple coalesce into a single economic unit acting under the general direction of the household head(s). It can be a nuclear or extended family that is financed by a working age couple, a single individual living alone, a pair that finances a joint livelihood, or any other number of configurations.

Replicate Weights: Anonymized Weighting

Respondents were chosen through one of two sampling mechanisms. Roughly two-thirds were selected through a geography-based clustering system in which allocations of respondents are randomly distributed across progressively narrower geographic regions until individual households are chosen. Another third was randomly sampled from IRS tax records to which the samplers were given special access, in part to oversample wealthy families (Bricker et al. 2016).

Our analysis of this data must account for the fact that it was built on a stratified and clustered sampling mechanism (see Heeringa et al. 2017). These mechanisms violate a basic assumption that our analytical units had an equal chance of being chosen and are independent from one another. Dependencies among units can lead to underestimated standard errors and anticonservative significance estimates. The distortion to respondents' probability of inclusion generates parameter estimates that are not properly calibrated to their true representation in the target population. This is certainly true in this case because the SCF deliberately oversamples the wealthy.

One problem is that crafting a correction requires detailed data on the respondents, such that we can make guesses about whether or not a particular respondent is in fact over- or under-sampled relative to their prevalnce in the population. However, the more data we offer, the greater the chance that people can be identified from the data, especially if they are have uncommon blends of demographic, geographic, and personal financial characteristics. The response is to craft *replicate weights*, which can be understood as a process that conducts multiple resamples of the data whose combination will ultimately adjust each individual observation's weight to their representation in the population. The method replaces detailed information that could make observations identifiable with (presumably) impossibly complicated permutations of the sample.

The weights corresponding to the households represented in the data are given in the replicate weights table distributed with this data. Each row corresponds to a distinct household, and to each of the five rows of imputed data pertaining to them on the main and summary data tables.

```r
# PART FOUR: DATA QUALITY CHECK
# Data quality check: Ensure all three files have corresponding rows
stopifnot( nrow( scf2022 ) == nrow( scf2022rw ) * 5 )  # One RW score per household
stopifnot( nrow( scf2022 ) == nrow( scf2022s ) )

#Confirm only the primary economic unit and the five implicate identifiers overlap:
stopifnot( all( sort( intersect( names( scf2022 ) , names( scf2022s ) ) ) == c( 'y1'
, 'yy1' ) ) )
stopifnot( all( sort( intersect( names( scf2022 ) , names( scf2022rw ) ) ) == c( 'y1'
, 'yy1' ) ) )
stopifnot( all( sort( intersect( names( scf2022s ) , names( scf2022rw ) ) ) == c(
'y1' , 'yy1' ) ) )

# Convert column names to lower case in all sets, per Damico script
names(scf2022) <- tolower(names(scf2022))
names(scf2022rw) <- tolower(names(scf2022rw))
names(scf2022s) <- tolower(names(scf2022s))

# Per Damico script
# Remove implicate identifier from RW table, and add column of fives for weighting
scf2022rw[, 'y1'] <- NULL
scf2022[,'five'] <- 5

save(scf2022, scf2022s, scf2022rw, file = "SCF 2022 Raw Data Tables.RData") # Save
the data

# PART FIVE: MERGE MAIN AND SUMMARY DATA
# Merge Summary and Raw Data Tables by 'y1'
scf2022 <- merge(scf2022, scf2022s, by = "y1", sort = T)


# PART SIX: RECAST DATA AS FIVE SEPARATE IMPLICATES
# Splitting data set into five separate sets of individual implicates
scf_1 <- subset(scf2022, as.numeric(substr(scf2022$y1, nchar(scf2022$y1),
nchar(scf2022$y1))) == 1)
scf_2 <- subset(scf2022, as.numeric(substr(scf2022$y1, nchar(scf2022$y1),
nchar(scf2022$y1))) == 2)
scf_3 <- subset(scf2022, as.numeric(substr(scf2022$y1, nchar(scf2022$y1),
nchar(scf2022$y1))) == 3)
scf_4 <- subset(scf2022, as.numeric(substr(scf2022$y1, nchar(scf2022$y1),
nchar(scf2022$y1))) == 4)
scf_5 <- subset(scf2022, as.numeric(substr(scf2022$y1, nchar(scf2022$y1),
nchar(scf2022$y1))) == 5)

# Clean Up Subject Identifier in Individual Implicates
for (i in 1:5){
  temp <- get(paste0("scf_", i))
  temp$yy1 <- temp$yy1.x
  temp$yy1.x <- NULL
  temp$yy1.y <- NULL
  assign(paste0("scf_", i), temp)
}

# Compile Individual Implicates to a List
scf_data_list <- list(scf_1, scf_2, scf_3, scf_4, scf_5)

# Removing unnecessary objects to save memory and space in Environment window
rm(scf_1, scf_2, scf_3, scf_4, scf_5)
rm(i, temp)
gc()


# PART SIX: CLEANING DATA
# Replace missing replicate weights with zeros to prevent downstream bugs
scf2022rw[ is.na( scf2022rw ) ] <- 0

# Rescale weights, per documentation
scf2022rw[ , paste0( 'wgt' , 1:999 ) ] <-
    scf2022rw[ , paste0( 'wt1b' , 1:999 ) ] * scf2022rw[ , paste0( 'mm' , 1:999 ) ]

# Using Damico's strategy of storing as a data table with y1 and wgts*
scf2022rw <- scf2022rw[ , c( 'yy1' , paste0( 'wgt' , 1:999 ) ) ]

# Check if yy1 values match across the datasets and RW table
all(scf_data_list[[1]]$yy1 == scf2022rw$yy1)
all(scf_data_list[[2]]$yy1 == scf2022rw$yy1)
all(scf_data_list[[3]]$yy1 == scf2022rw$yy1)
all(scf_data_list[[4]]$yy1 == scf2022rw$yy1)
all(scf_data_list[[5]]$yy1 == scf2022rw$yy1)

# Saving Clean Data & Clearning Memory
save(scf_data_list, scf2022rw, file = "SCF 2022 Data and RW Tables.RData")
rm(scf2022, scf2022rw, scf2022s)
gc()
```

Missing Data

The SCF uses multiple imputation with randomness, a method in which the analyst simulates missing data based on relationships within observed data. The method estimates missing values using a multivariate model, but creates multiple versions of the imputed sets with randomness injected into missing data estimates to address concerns that imputation artifically strengths the relationships upon which the missing data imputation model was built. In this set, the analysts created five different imputations, each of which replicates observed values and different, randomness-infused imputed values.

When trying to estimate linear statistics, like sample means or many types of regression coefficients, the process to yield population estimates from the five sets are as follows. The parameter estimate is the mean estimate of the five (or however many) imputed sets:

Coefficient Estimates. The combined estimate of the mean ($\bar{Q}$) is calculated as the average of the imputed means:

$$\bar{Q} = \frac{1}{m}\sum_{i=1}^{m}\bar{Q}_i$$

where $\bar{Q}_i$ is the estimate of the mean from the i^{th} imputation and m is the total number of imputations.

Variance Estimates. The combined estimate of variance (T) for the estimate $\bar{Q}$ takes into account both the within-imputation variance (W) and the between-imputation variance (B):

$$T = W + \left(1 + \frac{1}{m}\right)B$$

$$W = \frac{1}{m}\sum_{i=1}^{m}S_i^2$$

$$B = \frac{1}{m-1}\sum_{i=1}^{m}(\bar{Q}_i - \bar{Q})^2$$

where:

- m are the number of imputated sets
- W is the average of the within-imputation variances:
- B is the variance of the imputed estimates:
- S_i^2 is the variance estimate from the i^{th} imputation.

These formulas allow for the calculation of combined estimates that reflect the uncertainty due to missing data by including both the within-imputation variability and the variability across different imputations.

For Percentile Estimates. This analysis focuses on estimating percentile scores, and Rubin's Rule is conventionally applied to linear statistics. I did not find much literature engaging the issue of estimating percentiles from multiply imputed data. My analysis finds that the point estimates obtained using Lumley's R package (and thus commonplace practical strategies when analyzing data with R) replicate the results obtained with a direct application of Rubin's Rule.

Comparing the Performance of Scripts

It was my goal in this analysis to ensure that standard practice in the analysis of the SCF in R rendered acceptable analytical results. The issue was prompted by my inability to replicate Federal Reserve-published figures. This post provides an account of this analysis. I then checked that the recommended scripts replicated the results obtained in the SAS script in the documentation. The results obtained in Damico's scripts replicate those obtained using my translation of the documentation-provided SAS script.

I then reanalyzed my data as I would a set of this type were a recommended script not provided. The main differences between Damico's scripts and mine in how I structured the data. I broke the individual imputations into five data tables, analyzed them individually using the Lumley (2011) 'survey' package, and recombined using Rubin's Rule. The Damico scripts process the data using the 'ImputationList()' operation in the package *mitools* and employed a customized function to estimate the five imputations built on the 'survey' package.

```
# Damico Script
# Load Data
load("SCF 2022 Data and RW Tables.RData")

# Replicating Damico Object
scf_design <-
  svrepdesign(
    weights = ~wgt,
    repweights = scf2022rw[ , -1 ] ,
        data = imputationList( scf_data_list ) ,
        scale = 1 ,
        rscales = rep( 1 / 998 , 999 ) ,
        mse = FALSE ,
        type = "other" ,
        combined.weights = TRUE
  )

# Damico's function to combine implicates to give summary estimate from each of the
five sets.
scf_MIcombine <-
    function (results, variances, call = sys.call(), df.complete = Inf, ...) {
        m <- length(results)
        oldcall <- attr(results, "call")
        if (missing(variances)) {
            variances <- suppressWarnings(lapply(results, vcov))
            results <- lapply(results, coef)
        }
        vbar <- variances[[1]]
        cbar <- results[[1]]
        for (i in 2:m) {
            cbar <- cbar + results[[i]]
            # MODIFICATION:
            # vbar <- vbar + variances[[i]]
        }
        cbar <- cbar/m
        # MODIFICATION:
        # vbar <- vbar/m
```

```r
        evar <- var(do.call("rbind", results))
        r <- (1 + 1/m) * evar/vbar
        df <- (m - 1) * (1 + 1/r)^2
        if (is.matrix(df)) df <- diag(df)
        if (is.finite(df.complete)) {
            dfobs <- ((df.complete + 1)/(df.complete + 3)) * df.complete *
            vbar/(vbar + evar)
            if (is.matrix(dfobs)) dfobs <- diag(dfobs)
            df <- 1/(1/dfobs + 1/df)
        }
        if (is.matrix(r)) r <- diag(r)
        rval <- list(coefficients = cbar, variance = vbar + evar *
        (m + 1)/m, call = c(oldcall, call), nimp = m, df = df,
        missinfo = (r + 2/(df + 3))/(r + 1))
        class(rval) <- "MIresult"
        rval
    }

# Calculate quartile values using his recommended method:
nwdec.Q25_damico <- scf_MIcombine( with( scf_design,
                                         svyquantile(~ networth ,   0.25 ,
                                                     se = TRUE ,
                                                     interval.type = 'quantile' ) ) )

nwdec.Q50_damico <- scf_MIcombine( with( scf_design,
                                         svyquantile(~ networth ,   0.5 ,
                                                     se = TRUE ,
                                                     interval.type = 'quantile' ) ) )

nwdec.Q75_damico <- scf_MIcombine( with( scf_design,
                                         svyquantile(~ networth ,   0.75 ,
                                                     se = TRUE ,
                                                     interval.type = 'quantile' ) ) )

nwdec.Q90_damico <- scf_MIcombine( with( scf_design,
                                         svyquantile(~ networth ,   0.9 ,
                                                     se = TRUE ,
                                                     interval.type = 'quantile' ) ) )

damico_ests <- as.vector(c(nwdec.Q25_damico$coefficients,
                           nwdec.Q50_damico$coefficients,
                           nwdec.Q75_damico$coefficients,
                           nwdec.Q90_damico$coefficients))

create_survey_designs <- function(data_list, rep_weights, weight_column = 'x42001') {
  survey_designs <- lapply(data_list, function(data) {
    svrepdesign(repweights = as.matrix(rep_weights[ , -1]),  # Excluding the first
column assuming it's an identifier
                weights = as.formula(paste0("~", weight_column)),
                data = data,
                ids = ~y1,
                nest = FALSE,
                scale = 1,
                type = "other",
                rscales = rep(1 / 998, 999),
                combined.weights = TRUE,
                mse = TRUE)
  })
  return(survey_designs)
}

# Execute the function to create survey design objects
scf_designs <- create_survey_designs(scf_data_list, scf2022rw)

scf_percentiles <- function(design_list, var_name, percentile) {
  # Calculate quantiles for each design object
  quantiles_list <- lapply(design_list, function(design) {
    svyquantile(as.formula(paste0("~", var_name)), c(percentile), design = design,
na.rm = TRUE)
  })

  # Initialize a data frame to store the compiled results
  compiled_results <- data.frame(
    Iteration = integer(),
    Point = numeric(),
    SE = numeric(),
    stringsAsFactors = FALSE
  )

  # Extract data from quantiles_list and populate the data frame
  for (i in seq_along(quantiles_list)) {
    compiled_results <- rbind(compiled_results, data.frame(
      Iteration = i,
      Point = quantiles_list[[i]][[var_name]][,"quantile"],
      SE = quantiles_list[[i]][[var_name]][,"se"],
      stringsAsFactors = FALSE
    ))
  }

  # Create a final row that compiles the results

  W <- mean(compiled_results$SE^2)
  B <- var(compiled_results$Point)
  IMPS <- nrow(compiled_results)
  T <- W + (1 + 1/IMPS) * B
  combined_SE <- sqrt(T)

  combined <- c("combined",
                mean(compiled_results[,"Point"], na.rm = F),
                combined_SE)

  results_table <- rbind(compiled_results, combined)
  coef <- mean(compiled_results[,"Point"], na.rm = F)
  se <- round(combined_SE, 2)

  compiled_results <- list(results = results_table,
                           coef = coef,
                           se = se)

  return(compiled_results)
```

```
}

joe_nw_25 <- scf_percentiles(scf_designs, "networth", 0.25)
joe_nw_50 <- scf_percentiles(scf_designs, "networth", 0.50)
joe_nw_75 <- scf_percentiles(scf_designs, "networth", 0.75)
joe_nw_90 <- scf_percentiles(scf_designs, "networth", 0.90)
joe_ests <- as.vector(c(joe_nw_25$coef, joe_nw_50$coef, joe_nw_75$coef,
joe_nw_90$coef))
```

Empirical Results

Our analysis begins by comparing the empirical results obtained for estimates of the 25th, 50th, 75th, and 90th percentile values of U.S. household net worth in 2022. Table 1 (below) compares results yielded using both Damico's scripts, my own implementation of standard practice, and to official estimate from Table B.2 in Aladangady *et al.* (2024).

```
aladangady_estimates <- c(27100, 192900, 658900, 1938000)
```

The estimates in Aladangady *et al.* do not match those obtained in the Damico script. This discrepancy was the initial impetus of this analysis. Damico's results those that I obtained as the mean of the five imputed set's individual population percentile estimates.

```
results <- rbind(aladangady_estimates, damico_ests, joe_ests)
# Assuming your original data frame is named 'results'
# Transpose the results and convert to a data frame
results_transposed <- as.data.frame(t(results))

# Relabel the rows and columns
rownames(results_transposed) <- c("25th Percentile", "50th Percentile", "75th
Percentile", "90th Percentile")
colnames(results_transposed) <- c("Aladangady et al.", "Damico Scripts",
"Reanalysis")

# Format the numbers with commas
results_transposed[] <- lapply(results_transposed, function(x) format(x, big.mark =
",", scientific = FALSE))

# Create the table with a title using kable
kable(results_transposed, caption = "SCF Net Worth Estimates", format = "html") #
Change format to "latex" if needed
```

SCF Net Worth Estimates

	Aladangady et al.	Damico Scripts	Reanalysis
25th Percentile	27,100	27,016	27,016
50th Percentile	192,900	192,084	192,084

75th Percentile	658,900	658,340	658,340
90th Percentile	1,938,000	1,920,758	1,920,758

The results obtained using the R scripts do not match the officially-published results. Although discrepant, they are very close to those obtained by Aladangady *et et.*. Estimates differed across a range of 0.08% and 0.9% acorss the four percentile estimates obtained here. Are these discrepancies a signal of something problematic? The official documentation argues that these discrepancies can occur even with rigorous estimates:

> Results users may obtain from using this release of the 2022 SCF data may differ from those reported in this article for several reasons. First, a small number of the analysis weights used in that article may have been altered somewhat to provide robust estimates of the detailed categories shown. In brief, the data were examined for extreme outliers, and where a given case was overly influential in determining an outcome, the weight was trimmed and other weights were inflated to maintain a constant population. Second, as noted below, the public version of the data has been systematically altered to minimize the likelihood that unusual individual cases could be identified. Our analysis of the public data set suggests that these changes should not alter the conclusions of reasonable analyses of the data. Finally, over time we correct errors that we find in the data set. In our past experience, the effects of such errors on the estimates have been quite small.

This is consistent with best practice, as analysts should watch and correct for outliers and similar sources of distortion. Without access to the confidential data that they used, there is no way to verify and reproduce their decisions, but the discrepancies are so small as to be immaterial.

Conclusion

In this reanalysis of the household net worth percentiles estimated from the Survey of Consumer Finances, I am left with confidence in the quality of Anthony Damico's scripts. Although they do not replicate official reports, they render effectively similar results, and they likely represent the best an analyst can do with the public release set.

Works Cited

Allison, Paul D. 2010. *Missing Data*. Vol. 200210. Thousand Oaks, CA: Sage.

Bricker, Jesse, Alice Henriques Volz, Jacob Krimmel, and John Sabelhaus. 2016. *Measuring Income and Wealth at the Top Using Administrative and Survey Data*. Brookings Institution.

Carpenter, James R., Jonathan W. Bartlett, Tim P. Morris, Angela M. Wood, Matteo Quartagno, and Michael G. Kenward. 2023. *Multiple Imputation and Its Application*. John Wiley & Sons.

Cohen, Joseph Nathan. 2017. *Financial Crisis in American Households: The Basic Expenses That Bankrupt the Middle Class*. Santa Barbara: Praeger.

Heeringa, Steven G., Brady T. West, and Patricia A. Berglund. 2017. *Applied Survey Data Analysis*. CRC Press.

Little, Roderick JA, and Donald B. Rubin. 2019. *Statistical Analysis with Missing Data*. Vol. 793. John Wiley & Sons.

Lumley, Thomas. 2011. *Complex Surveys: A Guide to Analysis Using R*. John Wiley & Sons.