ICPSR | INTER-UNIVERSITY CONSORTIUM FOR POLITICAL AND SOCIAL RESEARCH

ICPSR 37692

Survey of Prison Inmates, United States, 2016

United States. Bureau of Justice Statistics

User's Guide

Inter-university Consortium for
Political and Social Research
P.O. Box 1248
Ann Arbor, Michigan 48106
www.icpsr.umich.edu

Terms of Use

Information about Copyrighted Content

Survey of Prison Inmates, 2016
User's Guide for Variance Estimation

Note: *This User's Guide applies to both the Public Use Files and Restricted Use File; users may have to employ different methods depending on which file they are using. Users should be aware that using different variance methods and different statistical packages could produce different results depending on the statistic(s) of interest.*

I. Introduction

An important aspect of analyzing survey data is incorporating weights and design features to obtain appropriate point and variance estimates. There are several methods for generating point and variance estimates. The goal of this User's Guide is to provide syntax for applying two of those methods – replicate weights and Taylor series linearization (TSL) – to the data from the 2016 Survey of Prison Inmates (SPI).

TSL is recommended for most analyses with the Restricted Use File (RUF), including any prospective analysis and statistical modeling. However, using replicate weights (i.e., the jackknife method) is recommended for RUF analyses that involve finite population estimation[1] by geography/jurisdiction (e.g., comparing state and federal prisoners) and any analyses of sex, sexual orientation, and gender identity[2]. For proportions and means, this may not be as important because the use of replicate weights may result in variance estimates that are not substantially different (i.e., less than a 10% difference) from variance estimates generated by TSL. However, the replication method is preferable over TSL to generate variances estimates of totals/counts because the TSL approach, as implemented in most statistical software, fails to fully account for variance contributions of the range of weighting adjustments, such as nonresponse adjustments and calibration to the known population control totals (Heeringa et al. 2017; Chowdhury 2013; Valliant 2004). In particular, TSL's inability to account for the precision gained from poststratification may result in overestimation of variance estimates of totals/counts when analyses involve the poststrata variables (i.e., geography/jurisdiction, including state and federal, and sex) or variables closely related to them, such as sexual orientation and gender identity. Both methods may produce standards errors of totals/counts that are not substantially different (i.e.,

[1] BJS has assumed a finite prison population because the 2016 SPI sample was adjusted to a specific point in time (i.e., one-day stock population) using control totals from the 2015 National Prison Statistics (NPS) program. BJS uses the NPS to annually enumerate the nation's prison population, including by jurisdiction (i.e., state and federal) and by sex. Therefore, BJS's SPI products do not typically include standard errors of totals/counts by jurisdiction and sex. Other methodologists may not make the same assumption of a finite population, in which case, replicate weights would be the preferred option and will likely produce different standard errors than if TSL was used.

[2] Although replicate weights are preferred in these latter analyses, the user can expect them to be more computationally intensive.

less than a 10% difference) when analyses do not involve the poststrata variables or variables closely related to them.

The Public Use File (PUF) does not contain the necessary identifiers to support the use of TSL; as such, the use of replicate weights is required for direct variance estimation using the PUF.

Provided below are examples for the use of replicate weights[3] in SAS, R, SUDAAN, and Stata, and the use of TSL in SAS, R, SUDAAN, SPSS and Stata[4].

II. Guide to Analyzing 2016 SPI Data

The approach to analyzing 2016 SPI data is presented by way of an example. Code is presented showing how to estimate the proportion of prisoners who were ever homeless while growing up. The example demonstrates cross-tabulating the "ever homeless" variable by sex and selecting only those observations in the state prison subpopulation. The sample code used throughout this guide estimates both proportions and standard errors of those proportions in each subgroup. These estimates have been highlighted in yellow in the output below.

Key variables used in these analyses are included in Table 1.

Table 1. Key variables required for example

Variable	Description
V1571	Geo_Stratum: Geographic Stratum for Variance Estimation
V1572	Sex_Stratum: Sex Stratum for Variance Estimation (facility roster)
V1573	FACILITYSORT: Facility Sequential ID
V1585	WT_FINAL: Final Analysis Weight
V1586-V1949	Replicate Weights 1-364
RV0005	Sex for Analysis (1=Male, 2=Female)
RV0009	Jurisdiction (1=Federal, 2=State)
RV0082[5]	While growing up - Ever homeless (1=Yes, 2=No, 98=DK/REF, 99=Missing)

The replicate weights need to be accompanied by replicate multipliers with one multiplier for each replicate weight. They are included in a spreadsheet (see SPI2016_JackknifeMult.xlsx.) separate from the survey data and are also specified in each example in purple.

[3] To obtain correct statistical estimates, standard errors, significance tests, and confidence intervals, data users need to specify both the analysis weight and replicate weights, as the example syntax that follows will demonstrate.

[4] To learn more about the analysis of complex survey data, see Lumley (2010) for details with R, Kolenikov et al. (2014) and Heeringa et al. (2017) for details with Stata; the latter book also has an online appendix with R code.

[5] RV0082 is temporarily suppressed in the public use and restricted use files but is expected to be made available when the remaining temporarily suppressed variables are completed by BJS.

The following font conventions are used. Input syntax is presented in `fixed width Courier font`. The code comments are additionally **`bold and colored in dark green`**. The expected output is presented in **`bold fixed width Consolas font`**. In copying and pasting the code, please be mindful of software that may replace the straight single or double quotes with the opening and closing ones [i.e., `"full/path/to/Stata/data/spi_data.dta"` with the standard quotes may get replaced with a different format (e.g., `“full/path/to/Stata/data/spi_data.dta”`) that the software will not recognize].

III. Using Replicate Weights in Statistical Software

Using the 2016 SPI data, an analyst can obtain direct variance estimates using the replicate weights provided in the dataset; this requires the use of sophisticated statistical software, including SUDAAN, SAS, R, and Stata, all reflected below.[6] Note that design weights are not provided in the PUF so replicate weights must be used in that file to produce direct variance estimates.

A. SUDAAN

Input syntax:

```
proc crosstab data= SPI_data_RUF design=jackknife ; *a;
     weight V1585; *b;
     jackwgts V1586-V1949; *c;
     jackmult 18*.944 46*.978 12*.917 3*.667 70*.986 160*.994 29*.966
4*.75 20*.95 2*.5; *d;
     subpopn RV0009=2; *e;
     subgroup RV0005 RV0082; *f;
     levels 2 2; *g;
     tables RV0005*RV0082 ; *h;
     print rowper serow /rowperFMT=f8.4 serowFMT=f8.4; *i;
     output rowper serow / filename=SUDAANEsts_REP replace; *j;
run;
```

Code Comments:

a) Specifies dataset, which is RUF in this case, and jackknife as the variance estimation method.
b) Specifies the main weight variable.
c) Specifies the jackknife replicate weights.

Subpopulation estimation

Each of the packages reviewed in this User Guide has specific methods to restrict a population of interest (subpopulation). In SUDAAN, one uses the `SUBPOP` or `SUBPOPN` statement. In SAS, for SURVEYFREQ, one uses the variable as a level in the table, but for other procedures (e.g. SURVEYMEANS, SURVEYLOGISTIC), one uses a DOMAIN statement. In R, `filter` or `subset` a survey design object after specifying with the entire dataset. In Stata, use the `subpop` option of `svy:` estimation prefix. In SPSS, specify a subpopulation variable when completing an analysis.

[6] SPSS users cannot use replicate weights, as of version 25.

d) Specifies the jackknife replicate weight multipliers.
e) Restricts analysis to prisoners in the state facilities only.
f) Specifies variables, which are categorical.
g) Specifies the levels of each variable, which are assumed to be 1 and 2 if 2 is specified. This ignores the levels of 98/99. Alternatively, one could set RV0082 to missing when it is 98/99 and then use a class statement only.
h) Specifies the table to be produced.
i) Prints output (shown below) and specifies a display of row percent and standard error of row percent. Each will be printed with 4 decimals.
j) Outputs row percent and standard error of row percent to a dataset. Replace option will replace any dataset existing that is called SUDAANEsts_REP.

The core survey estimation syntax is provided by the lines commented (a) through (d). The syntax specific to the particular table of interest is contained in the lines commented (e) through (h). Lines commented (i) and (j) control the presentation of the output of the analysis.

Expected output:

RV0005: Sex		RV0082: While growing up - Ever homeless		
		Total	1=Yes	2=No
Total	Row Percent	100.0000	11.7950	88.2050
	SE Row Percent	0.0000	0.3767	0.3767
1=Male	Row Percent	100.0000	11.7243	88.2757
	SE Row Percent	0.0000	0.4037	0.4037
2=Female	Row Percent	100.0000	12.7131	87.2869
	SE Row Percent	0.0000	0.6104	0.6104

B. SAS

For the SAS analysis, some pre-processing is required. In terms of the core survey estimation syntax, the dataset of jackknife multipliers needs to be created for PROC SURVEYFREQ to use. In terms of the analysis of interest, RV0082 is set to missing when it is equal to 98 or 99. In addition, dummy variable _one_ is created, which sets the value for all records to 1. SAS does not enable domain estimation when using the SURVEYFREQ procedure, so the jurisdiction variable must be used as a variable in the crosstab. The results of cross-tabulation and domain estimation are algebraically identical.

Input syntax:

```
data SPIDatforSAS;
      set SPI_data_RUF;
      if RV0082 in (98,99) then call missing(RV0082); *a;
      _one_=1; *b;
run;

data JkCoefs; *c;
      array times{10} _temporary_ (18 46 12 3 70 160 29 4 20 2); *d;
      array mults{10} _temporary_ (.944 .978 .917 .667 .986 .994 .966
.75 .95 .5); *e;
      do i = 1 to dim(times); *f;
            JKCoefficient=mults{i};
            do j=1 to times{i};
                  output;
            end;
      end;
      drop i j;
run;

proc surveyfreq data=SPIDatforSAS varmethod=jackknife ; *g;
     weight V1585; *h;
     repweights V1586-V1949 /jkcoefs=JkCoefs; *i;
     tables RV0009*RV0005*_one_*RV0082 /nofreq nototal nowt; *j;
     ods output CrossTabs=SASEsts_Rep; *k;
run;
```

Code Comments:

a) Changes RV0082 to missing when it is 98 or 99.
b) Creates a dummy variable for use later that is always 1.
c) Makes a temporary dataset for replicate coefficients.
d) Designates the number of times each replicate coefficient is used.
e) These are the replicate multipliers.
f) This loop writes each coefficient the number of times specified.
g) Specifies dataset and variance method (jackknife) which is a replicate method.
h) Specifies the main weight (V1585).
i) Specifies the replicate weights (V1586-V1949) and replicate coefficients (to be picked up from the temporary dataset created in step c).
j) Specifies table – first the state/federal, then the sex, then dummy variable _one_, then the ever homeless variable. (The variable _one_ is created for presentation purposes only, so that this specific table is easier to read.)
k) Outputs table to a dataset.

The core survey estimation syntax is provided by the lines commented (c) through (f) to create a dataset of multipliers, and in lines commented (g) through (i). The syntax specific to the particular table of

interest is contained on line commented (j), as well as in the preprocessing in lines (a) and (b). Lines commented (j) and (k) control the presentation of the output of the analysis.

Expected output (in SAS output window):

Table of _one_ by RV0082

Controlling for RV0009=2=State RV0005=1=Male

one	RV0082	Percent	Std Err of Percent
1	1=Yes	11.7243	0.4037
	2=No	88.2757	0.4037

Table of _one_ by RV0082

Controlling for RV0009=2=State RV0005=2=Female

one	RV0082	Percent	Std Err of Percent
1	1=Yes	12.7131	0.6104
	2=No	87.2869	0.6104

C. R

For analysis in the R statistical package, there are various ways to load the data into R from the formats used by other programs, as demonstrated below. The libraries used are demonstrated by explicitly using the double-colon syntax of `package::function()` where applicable.

1. Import SAS file:

```
library(haven)
spi_data <- haven::read_sas(
   data_file = "full/path/to/SAS/data/spi_data_RUF.sas7bdat",
   catalog_file = NULL) #a
```

The second argument specifies the formats file, if applicable. If there is no formats file provided, `NULL` is used. The option names are optional.

2. Import Stata file, `library(foreign)`:

```
library(foreign)
spi_data <- foreign::read.dta("full/path/to/Stata/data/spi_data_RUF.dta")
#a
```

Limitation of this library is that it is only able to read files up to Stata version 12 (as of package version 0.8.72). To prepare data in that format, one should load data in Stata and use saveold command:

```
saveold spi_data12, version(12)
```

3. Import Stata file, `library(haven)`:

```
library(haven)
spi_data <- haven::read_dta("full/path/to/Stata/dataset/spi_data_RUF.dta")
#a
```

Input syntax:

Somewhat similar to Stata, analysis of complex survey data follows a specification step and an analysis step. Specification of the replicate weights can be done as follows. Using the replicate weight type as `type="other"` leads to identical results.

```
repwts <- paste0("V", 1586:1949)  #b
spi_svy <- svrepdesign(
    data       = spi_data,        #c
    weights    = ~V1585,          #d
    repweights = spi_data[, repwts], #e
    type       = "JK1",
    scale      = 1,
    rscales = rep(c(.944, .978, .917, .667, .986, .994, .966, .75, .95, .5),
            times = c(18,   46,   12,    3,   70,  160,   29,   4,  20,  2)
)
)
```

This is a special R object of class `svyrep.design` provided by `survey` package. Strictly speaking, it is not an R dataset, so it cannot be saved as .Rdata. However, as any R object, it can be saved and opened back with `saveRDS()` and `readRDS()` functions, which can simplify the analysis workflow:

```
saveRDS(spi_svy, file="full/path/to/file/spi_svy.Rds")
spi_svy <- readRDS(file="full/path/to/file/spi_svy.Rds")
```

Control for the missing data can be done at the analysis stage by subsetting the dataset:

```
svyby(
    ~as.factor(RV0082),      #f
    by=~RV0005,              #g
    design=subset(spi_svy,   #h
          RV0082 < 90
          & RV0009==2),      #i
    FUN = svymean            #j )
```

Code Comments:

a) spi_data is the RUF dataset read into R.
b) Specify replicate weights (V1586:V1949).
c) Specify dataset for survey design (spi_data).
d) Specify analysis weight (V1585).
e) Set data frame with replicate weights.
f) Set the "column" variable (RV0082).
g) Set the "row" variable (RV0005).
h) Pass the survey design object with baked-in survey design information.
i) Restrict analysis to state prisons (RV0009=2) and when RV0082 (ever homeless) is not missing.
j) Specify the estimation command. For R factor variables as specified in line (f), svymean() produces proportions.

The core survey estimation syntax is provided by the lines commented (b) through (e). The syntax specific to the particular table of interest is contained on lines commented (f) through (j).

Expected output:

```
  RV0005 as.factor(RV0082)1 as.factor(RV0082)2         se1         se2
1      1          0.1172432          0.8827568 0.004037068 0.004037068
2      2          0.1271312          0.8728688 0.006104513 0.006104513
```

D. Stata

Input syntax:

```
use "full/path/to/Stata/dataset/spi_data_RUF.dta" // a

svyset [pweight=V1585], vce(jackknife) mse /// b
       jkrw(V1586-V1949, ///
              multiplier( /// multipliers
              /// stratum 1:
              0.944 0.944 0.944 0.944 0.944 0.944 0.944 0.944 0.944 ///
              0.944 0.944 0.944 0.944 0.944 0.944 0.944 0.944 0.944 ///
              /// stratum 2:
              0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 ///
              0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 ///
              0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 ///
              0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 ///
              0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 0.978 ///
              /// stratum 3:
              0.917 0.917 0.917 0.917 0.917 0.917 0.917 0.917 0.917 ///
              0.917 0.917 0.917 ///
              /// stratum 4:
              0.667 0.667 0.667 ///
              /// stratum 5:
              0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 ///
              0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 ///
              0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 ///
              0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 ///
              0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 ///
```

```
0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 ///
0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 0.986 ///
0.986 0.986 0.986 0.986 0.986 0.986 0.986 ///
/// stratum 6:
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
0.994 0.994 0.994 0.994 0.994 0.994 0.994 ///
/// stratum 7:
0.966 0.966 0.966 0.966 0.966 0.966 0.966 0.966 0.966 ///
0.966 0.966 0.966 0.966 0.966 0.966 0.966 0.966 0.966 ///
0.966 0.966 0.966 0.966 0.966 0.966 0.966 0.966 0.966 0.966 0.966 ///
/// stratum 8:
0.750 0.750 0.750 0.750 ///
/// stratum 9:
0.950 0.950 0.950 0.950 0.950 0.950 0.950 0.950 0.950 ///
0.950 0.950 0.950 0.950 0.950 0.950 0.950 0.950 0.950 ///
0.950 0.950 ///
/// stratum 10
0.500 0.500) ///
///
stratum( /// strata ID
/// stratum 1:
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ///
/// stratum 2:
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ///
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ///
/// stratum 3:
3 3 3 3 3 3 3 3 3 3 3 3 3 ///
/// stratum 4:
4 4 4 ///
/// stratum 5:
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 ///
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 ///
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 ///
/// stratum 6:
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 ///
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 ///
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 ///
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 ///
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 ///
```

```
            6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 ///
            /// stratum 7:
            7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 ///
            /// stratum 8:
            8 8 8 8 ///
            /// stratum 9:
            9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 ///
            /// stratum 10
            10 10) ///
            /// end of jackknife options)
            )
```

This syntax only needs to be used once, and can be saved with the analysis dataset:

```
use SPI_2016_ruf.dta, clear
svyset …
save, replace
```

If the user saves the file with the survey settings, then the next time this file is loaded, the survey settings will already be available, which can be checked by running svyset command by itself without any additional options.

The code specific to the analysis problem is as follows.

```
svy , /// c
    subpop(if RV0009==2 & RV0082 < 90): /// d
    tab RV0005 RV0082, /// e
    row se             /// f
```

Code Comments:

a) Specify the file path to where the data is.
b) Specify sampling design for SPI including all the replicate multipliers.
c) Invoke estimation that is aware of the complex survey structure of the data.
d) Subset to the population in state facilities with nonmissing RV0082 (works independently of whether RV0082 would have been recorded to have missing values in place of 98/99 since missing values in Stata are larger than any real number).
e) Create table of RV0082 by RV0005.
f) Report proportions by row, and include standard error.

The core survey estimation syntax is provided by the lines commented (b) and (c) (the svy: estimation prefix). Note that the code that contains the continuation comments `///` needs to be inserted into a do-file and run as a do-file; Stata does not recognize the continuation comments in the interactive mode. The syntax specific to the particular table of interest is contained on line commented (e), and line (f) controls the presentation of the output in the table.

Expected output:

```
------------------------------------
          |RV0082: While growing up -
```

```
RV0005:    |       Ever homeless
Sex        |       1        2    Total
-----------+--------------------------
         1 |   .1172    .8828        1
           | (.0041)  (.0041)
           |
         2 |   .1271    .8729        1
           | (.0069)  (.0069)
           |
     Total |    .118     .882        1
           | (.0038)  (.0038)
-----------------------------------
```

IV. Using Taylor Series Linearization in Statistical Software (RUF only)

TSL estimation requires the software to know the entire sampling design to calculate the appropriate standard errors and statistical tests. Thus, one should not subset data prior to analyzing in any of the software packages. See description of subpopulation estimation on page 2.

On the dataset, there are no instances of only one observation per PSU (which is the prison facility). If the software produces an error that the variance cannot be computed because there is only one observation per PSU, this may have been caused by subsetting data prior to analyzing. Errors in SUDAAN will be phrased as "Cannot compute variance contribution for WR design when sample size is 1". Warnings in SAS will be phrased as "NOTE: There is at least one stratum that contains only a single cluster for the table of RV0082. Single-cluster strata are not included in the variance estimates." Note that SAS will still produce estimates, but the standard errors may not be correct. R will give an error such as "Stratum (2.2) has only one PSU at stage 1." Errors in Stata will be phrased as "Note: missing standard error because of stratum with single sampling unit."

A. SUDAAN

Input syntax:

```
proc crosstab data=SPI_data_RUF design=wr notsorted; *a;
     weight V1585; *b;
     nest V1571 V1572 V1573 / psulev=3; *c;
     subpopn RV0009=2; *d;
     subgroup RV0005 RV0082; *e;
     levels 2 2; *f;
     tables RV0005*RV0082 ; *g;
     print rowper serow /rowperFMT=f8.4 serowFMT=f8.4; *h;
     output rowper serow / filename=SUDAANEsts_TSL replace; *i;
run;
```

a) Specifies dataset, which is the RUF, as well as the details of how it is organized (e.g. not being sorted by the design variables).
b) Specifies the main weight variable.
c) Specifies strata and cluster, identifies V1573 (the third variable in the list) as the PSU.

d) Restricts analysis to state prisoners only.
e) Specifies variables, which are categorical.
f) Specifies the levels of each variable, which are assumed to be 1 and 2 if 2 is specified. This ignores the levels of 98/99. Alternatively, could set RV0082 to missing when it is 98/99 and then use a class statement only.
g) Specifies table to be produced.
h) Prints output (shown below) and specifies to display row percent and standard error of row percent. Each will be printed with 4 decimal points.
i) Outputs row percent and standard error of row percent to a dataset. Replace option will replace any dataset existing that is called SUDAANEsts_TSL.

The core survey estimation syntax is provided by the lines commented (a) through (c). The syntax specific to the particular table of interest is contained on line commented (d) through (g). Lines commented (h) and (i) control the presentation of the output of the analysis. Note that the point estimates are identical to those produced with the replicate weights (as they should be, as they are only affected by the main weight), but the standard errors differ slightly (e.g., 0.004037 vs. 0.004015).

Expected output:

RV0005: Sex		RV0082: While growing up - Ever homeless		
		Total	1=Yes	2=No
Total	Row Percent	100.0000	11.7950	88.2050
	SE Row Percent	0.0000	0.3746	0.3746
1=Male	Row Percent	100.0000	11.7243	88.2757
	SE Row Percent	0.0000	0.4015	0.4015
2=Female	Row Percent	100.0000	12.7131	87.2869
	SE Row Percent	0.0000	0.5897	0.5897

B. SAS

Input syntax:

```
data SPIDatforSAS;
     set SPI_data_RUF;
     if RV0082 in (98,99) then call missing(RV0082); *a;
     _one_=1; *b;
run;

proc surveyfreq data=SPIDatforSAS varmethod=TAYLOR NOSUMMARY ; *c;
     strata V1571 V1572; *d;
     cluster V1573; *e;
     weight V1585; *f;
     tables RV0009*RV0005*_one_*RV0082 /NOFREQ NOTOTAL NOWT; *g;
     ods output CrossTabs=SASEsts_TSL; *h;
run;
```

Code Comments:

a) Changes RV0082 to missing when it is 98 or 99.
b) Creates a dummy variable for use later that is always 1.
c) Specifies dataset and variance method (TAYLOR) which is the TSL method.
d) Specifies strata (V1571, V1572).
e) Specifies cluster (V1573).
f) Specify analysis weight (V1585).
g) Specifies table – first state/federal, then sex, then dummy variable, then ever homeless variable.
h) Outputs table to a dataset.

The core survey estimation syntax is provided by the lines commented (c) through (f). The syntax specific to the particular table of interest is contained on line commented (g), as well as in preparation steps (a) and (b). Lines commented (g) and (h) control the presentation of the output of the analysis. The output to a dataset is optional; users may only be interested in viewing the output on the screen.

Expected output:

Table of _one_ by RV0082

Controlling for RV0009=2=State RV0005=1=Male

one	RV0082	Percent	Std Err of Percent
1	1=Yes	11.7243	0.4015
	2=No	88.2757	0.4015

Table of _one_ by RV0082			
Controlling for RV0009=2=State RV0005=2=Female			
one	RV0082	Percent	Std Err of Percent
1	1=Yes	12.7131	0.5897
	2=No	87.2869	0.5897

C. R

Input syntax:

The steps to read the SAS or Stata data into R are provided in the instructions concerning replicate weights (see Section III.C above). After the data are read into R, to create a design object for linearization variance estimation, the following syntax can be used:

```
spi_data$stratra = 10*spi_data$V1571 + spi_data$V1572 # a
spi_svy_lin <- svydesign( # b
  data = spi_data,  # c
  weights=~V1585,   # d
  id = ~V1573,      # e
  strata = ~strata  # f
)
```

Code Comments:

a) Create a variable to contain the strata identifiers. (Any other way to create interaction of these two variables will do).
b) The command to create a survey design object for variance estimation with Taylor series linearization.
c) Specifies the input dataset.
d) Specifies analysis weight (V1585), as a formula.
e) Specifies the variable containing PSU (V1573), as a formula.
f) Specifies the variable containing strata (created in step (a) above), as a formula.

The estimation command itself is identical to the one used with replicate weights, with the appropriate survey design object being used instead.

```
svyby( ~as.factor(RV0082), by=~RV0005, design=subset(spi_svy_lin, RV0082 <
90 & RV0009==2), FUN=svymean)
```

Expected output:

```
RV0005 as.factor(RV0082)1 as.factor(RV0082)2 se.as.factor(RV0082)1 se.as.factor(RV0082)2
```

```
1      1          0.1172432          0.8827568          0.004015198          0.004015198
2      2          0.1271312          0.8728688          0.005897294          0.005897294
```

A different flavor of R syntax, the tidyverse syntax, can be specified with the interface between the tidyverse and the survey library as follows.

```
# a
library(tidyverse) #b
library(srvyr)     #c
library(naniar)    #d

SpiDat <- SPI_data_RUF %>% #e
  replace_with_na(replace=list(RV0082=c(98,99))) #f
  mutate(strata=interaction(V1571, V1572))       #g

spiTSLDes <- SpiDat %>%    #h
  as_survey(weights=V1585, #i
            strata=strata, #j
            id=V1573)      #k

spiTSLDes %>%
  filter(RV0009==2 & !is.na(RV0082)) %>% #l
  group_by(RV0005, RV0082) %>%           #m
  summarize(Proportion=survey_mean())    #n
```

Code Comments:

a) Load necessary packages
b) tidyverse allows for easy to read code
c) srvyr is tidyverse interface to library(survey)
d) naniar works with missing data
e) SPI_data_RUF is the RUF dataset read into R.
f) Changes RV0082 to NA when it is 98 or 99.
g) Make a variable for strata which is a combination of the geography stratum (V1571) and sex stratum (V1572).
h) Specify dataset for survey design (SpiDat).
i) Specify analysis weight (V1585). Note the quoted format of the variable assignment rather than the formula syntax used by library(survey).
j) Specify strata variable.
k) Specify cluster variable which is the facility (V1573).
l) Restrict analysis to state prisons (RV0009==2) and when RV0082 (ever homeless) is not missing.
m) Specify the specific analysis by RV0005 (sex) and RV0082 (ever homeless).
n) Specify to provide the mean for each group. When using survey_mean with no argument, it gives the proportion within that group.

Expected output:

```
# A tibble: 4 x 4
  RV0005 RV0082 Proportion Proportion_se
   <dbl> <chr>       <dbl>         <dbl>
1   1.00 1           0.117       0.00402
2   1.00 2           0.883       0.00402
3   2.00 1           0.127       0.00588
4   2.00 2           0.873       0.00588
```

D. Stata

Input syntax:

```
use "full/path/to/Stata/dataset/SPI_data_RUF.dta" // a
egen strat=group(V1571 V1572)                     // b
svyset V1573 [pweight=V1585], strata(strat) vce(linearized) // c
svy, subpop(if RV0009==2 & RV0082<90) : tabulate RV0005 RV0082, row se // d
```

Code comments:

a) Read in SPI RUF.
b) Create a strata code by combining V1571 and V1572.
c) Specify sampling design for SPI and variance method.
d) Create table of RV0082 by RV0005: compute the estimates and the standard errors accounting for the sampling design, restrict analysis to the subpopulation of inmates in state prisons with nonmissing target analysis variable RV0082, output the row proportions, include standard error. (See detailed comments in Section III.D on the components of Stata survey estimation syntax.)

Expected output:

```
------------------------------------
          |RV0082: While growing up -
RV0005:   |       Ever homeless
Sex       |       1        2    Total
----------+-------------------------
        1 |   .1172    .8828        1
          |  (.004)   (.004)
          |
        2 |   .1271    .8729        1
          | (.0059)  (.0059)
          |
    Total |    .118     .882        1
          | (.0037)  (.0037)
------------------------------------
```

E. SPSS

Input syntax:

In SPSS, one must first specify the sampling design. The script to make the plan is as follows where the user needs to specify a file path for their computer:

```
* Analysis Preparation Wizard.
CSPLAN ANALYSIS
  /PLAN FILE="SpecifyFilePath"
  /PLANVARS ANALYSISWEIGHT=V1585
  /SRSESTIMATOR TYPE=WOR
  /PRINT PLAN
  /DESIGN STRATA=V1571 V1572 CLUSTER=V1573
  /ESTIMATOR TYPE=WR.
```

To set RV0082 to missing when the value is 98 or 99, use the following steps:

1. From the menu, select Data -> Define Variable Properties
2. Select RV0082 as a variable to scan and Continue
3. Check the box of "Missing" next to 98 and 99
4. Press OK

Run this code in SPSS and then check the sampling plan was created where you specified. To do the specified analysis of the proportion of males and females in state prisons who were homeless while growing up, use the following steps:

1. From the menu select Analyze -> Complex Samples -> Crosstabs
2. Select the plan you created. For Joint Probabilities, Use Default file and click Next.
3. Use the SPI dataset
4. Select RV0009 (Jurisdiction) as the Subpopulation
5. Select RV0005 (Sex) as the Column
6. Select RV0082 (While growing up – ever homeless) as the Rows
7. Select Statistics and check Column percent and Standard error then select Continue
8. Press OK once you have rows, columns, subpopulations, and statistics set

Expected output:

RV0082: While growing up - Ever homeless * RV0005: Sex

				RV0005: Sex		
RV0009: Jurisdiction	RV0082: While growing up - Ever homeless			1	2	Total
2	1	% within RV0005: Sex	Estimate	11.7%	12.7%	11.8%
			Standard Error	0.4%	0.6%	0.4%
	2	% within RV0005: Sex	Estimate	88.3%	87.3%	88.2%
			Standard Error	0.4%	0.6%	0.4%
	Total	% within RV0005: Sex	Estimate	100.0%	100.0%	100.0%
			Standard Error	0.0%	0.0%	0.0%

References

Chowdhury, S. (2013). *A Comparison of Taylor Series Linearization and Balanced Repeated Replication Methods for Variance Estimation in Medical Expenditure Panel Survey*. Agency for Healthcare Research and Quality Working Paper No. 13004.

Heeringa, S. G., West, B. T., and Berglund, P. A. (2017). *Applied Survey Data Analysis*. 2nd edition. Boca Raton, FL: Chapman & Hall/CRC.

Kolenikov, S., and Pitblado, J. (2014). Analysis of Complex Health Survey Data. Chapter 29 in: Johnson, T. (ed.), *Handbook of Health Survey Methods*. New Hoboken, NJ: Wiley.

Lumley, T. (2010). *Complex Surveys: A Guide to Analysis Using R.* New Hoboken, NJ: Wiley.

Valliant, R. (2004). *The Effect of Multiple Weighting Steps on Variance Estimation*. Journal of Official Statistics, 20 (1), 1–18.