New York City
Housing and Vacancy Survey

Guide to Estimating Variances

7/19/2019

U.S. Census Bureau, Department of Commerce
New York City Department of Housing Preservation and Development

Table of Contents

1. Introduction....1

2. General Variance Estimation for NYCHVS....4

Example 2.1. Estimating the Variance of the Total Number of Housing Units in a Domain....5

3. Estimating Variances with SAS PROC SURVEY Procedures....11

Example 3.1. Estimating the Variance of a Mean with SAS PROC SURVEYMEANS....12

Example 3.2. Estimating the Variance for a Contingency Table with SAS PROC SURVEYFREQ....13

Example 3.3. Estimating the Variance of a Regression Coefficients with SAS PROC SURVEYREG....14

4. Examples of Calculating Variances with Base SAS....18

Example 4.1. Estimating the Variance of a Total....18

Example 4.2. Estimating the Variance of a Difference....20

Example 4.3. Estimating the Variance of a Mean....22

Example 4.4. Estimating the Variance of a Median....24

Example 4.5. Estimating the Variance of a Regression Parameter....26

Example 4.6. Estimating the Variance of a Longitudinal Change....27

Example 4.7. Estimating the Variance of a Percent Change in Two Cycles of NYCHVS....30

5. Examples of Calculating Variances with Stata....33

Example 5.1. Estimating the Variance of a Total....33

Example 5.2. Estimating the Variance of a Difference....35

Example 5.3. Estimating the Variance of a Mean....36

Example 5.4. Estimating the Variance of a Median....37

Example 5.5. Estimating the Variance of a Regression Parameter....38

Example 5.6. Estimating the Variance of a Longitudinal Change....39

Example 5.7. Estimating the Variance of a Percent Change in Two Cycles of NYCHVS....42

6. Examples of Calculating Variances with R....45

Example 6.1. Estimating the Variance of a Total....45

Example 6.2. Estimating the Variance of a Difference....47

Example 6.3. Estimating the Variance of a Mean....48

Example 6.4. Estimating the Variance of a Median....49

Example 6.5. Estimating the Variance of Regression Parameter....49

Example 6.6. Estimating the Variance of a Longitudinal Change....50

Example 6.7. Estimating the Variance of a Percent Change in Two Cycles of NYCHVS....52

7. How to Calculate Confidence Intervals (CIs).....54

Example 7.1. Estimating CIs of a Total with Base SAS.....55

Example 7.2. Estimating CIs of a regression parameter with SAS PROC SURVEYREG.....56

Example 7.3. Estimating CIs of a Total with Stata Manually.....58

Example 7.4. Estimating the CI of a Total with R.....59

8. How to Calculate Confidence Intervals for the Odds Ratio.....61

Example 8.1. Estimating a CI of an Odds Ratio with Replicate Weights and SAS PROC SURVEYLOGISTIC.....65

Example 8.2. Estimating a CI of an Odds Ratio with Replicate Weights and SAS PROC SURVEYFREQ.....67

Example 8.3. Estimating a CI of an Odds Ratio with Replicate Weights and Base SAS.....69

Example 8.4. Estimating a CI of an Odds Ratio with Replicate Weights and Stata.....72

Example 8.5. Estimating a CI of an Odds Ratio with Replicate Weights and R.....73

Example 8.6. Estimating a CI of an Odds Ratio with Final weights and SAS PROC SURVEYFREQ.....75

Example 8.7. Estimating a CI of an Odds Ratio with Normalized Weights and SAS PROC FREQ.....77

Example 8.8. Estimating a CI of an Odds Ratio with Final weights and SAS PROC FREQ.....79

Example 8.9. Estimating a CI of an Odds Ratio with No Weights and SAS PROC FREQ.....80

9. How the Replicate Weights Are Calculated.....83

Variance Estimates with Replication.....83

Replicate Weights.....83

Replicate Factors.....83

Example 9.1. Successive Difference Replication.....84

Other Weighting Adjustments for Replicate Weights.....86

The Factor of 4 in Equation (2.1).....87

10. References.....88

Appendix A: Glossary of Terms

1. Introduction

The New York City (NYC) Housing and Vacancy Survey (NYCHVS) provides estimates of rental and homeowner vacancy rates, as well as various household and person characteristics. In 2017, the NYCHVS Public Use Files (PUFs) include the addition of replicate weights. The replicate weights allow data users an additional tool to calculate estimates of variance. Using the information provided in this guide and the replicate weights on the PUFs, data users will have the necessary tools to compute estimates of variance using the replicate weights.

Only Sample Design Variances Estimated

By variance, we refer to the sample design variance or simply design variance or the variance from a finite population sample. The variance measured by the replicate weights represents the variance of the estimated statistic if we repeated the sample selection many times and estimated the statistic of interest with each sample. See textbooks by Cochran (1977), Wolter (2007), and Särndal, Swensson, and Wretman (1992) for detailed discussions on sample design and sample variances.

How the Guide is Organized

Section 2 of the guide describes "how to estimate variances yourself using replicate weights." This part of the guide explains how the new variance estimation tools provided by the NYCHVS can be used to estimate variances.

Replicate weights can be used to estimate sampling variance for any complex statistic from the survey design. The bulk of the guide – sections 3 to 6 – reviews several examples that show how to use replicate weights to estimate variances of several types of statistics. The examples are repeated in dedicated sections for the following software packages: SAS, STATA and R. We include two separate sections on SAS: one section describes the PROC SURVEY procedures and another section describes how to use base SAS to calculate the variances.

Table 1 summarizes the examples and the four main sections of the guide.

Table 1: Summary Table of Examples

Type of Statistics	Description of Specific Example	Example Number/Section			
		SAS PROC SURVEY	Base SAS	STATA	R
Total	Number of 2017 occupied housing units in NYC	2.1	4.1	5.1	6.1
Difference	Difference of the number of 2017 rent-stabilized housing units in Manhattan and the Bronx		4.2	5.2	6.2
Mean	2017 average gross rent for renter-occupied housing units in NYC	3.1	4.3	5.3	6.3
Median	2017 Median gross rent for renter-occupied housing units in NYC		4.4	5.4	6.4
Odds Ratio	Odds of rent stabilization status (pre-1947 or post-1947) with maintenance deficiencies	8.1, 8.2	8.3	8.4	8.5
Regression Parameter	The regression parameter representing the year the householder moved into the housing unit for renters.	3.3	4.5	5.5	6.5
Longitudinal Change	The difference of the 2017 and 2014 vacancy rates		4.6	5.6	6.6
Percent Change	Percent change in the median gross rent from 2014 to 2017		4.7	5.7	6.7

To further demonstrate the importance of using the replicate weights in variance estimation, Section 7 provides a general example of confidence interval calculations and then Section 8 provides a detailed example using the odds ratio and confidence interval calculations.

Section 9 explains how the replicate weights are calculated. We provide this section for transparency, context, and background for sophisticated data users.

Scope of the Guide

The scope of this guide primarily includes the data provided in the 2017 NYCHVS. Two examples demonstrate longitudinal estimates which incorporates 2014 NYCHVS data. Sample Year 2017 is the first year that replicate weights have been released for NYCHVS. Replicate weights for prior years of 2011 and 2014 have been retroactively created and are available upon request.

The methods provided in this guide can be used for both housing unit (HU) estimates as well as person estimates. Even though examples of person estimates are not provided, the same methods can be applied with the substitution of the person replicate weights.

The results provided in this guide, including the estimations, standard errors, and confidence intervals are approximations, and are subject to errors. We do not provide a review of the sample design for the NYCHVS, but refer the reader to the documentation of the survey for a

more thorough discussion of the design and errors: 2017 New York City Housing and Vacancy Survey Sample Design, Weighting, and Error Estimation (U.S. Census Bureau 2018).

Within the guide, we provide several examples using SAS. The examples use SAS because SAS has a set of excellent procedures for working with survey data and we have great familiarity with the software. Additionally, we provide some of the same examples using the statistical packages of STATA and R and identify any limitations that we have found with this software. The examples use the publicly available data sets provided by NYCHVS and therefore, you should be able to replicate the results.

2. General Variance Estimation for NYCHVS

The variance of any survey estimate based on a probability sample may be estimated by the method of replication. This method requires that the sample selection, the collection of data, and the estimation procedures be independently carried through (replicated) several times. Each time the sample is replicated, a different set of estimates is calculated. The dispersion of the resulting estimates then can be used to measure the variance of the sample.

However, we would not consider repeating any large survey, such as the NYCHVS, several times to obtain variance estimates. A practical alternative is to alter the sample several times by applying different weighting factors to the sample units. The alterations of the replicate weights allow the single sample to represent multiple replicate samples that can be used to estimate variances. We sometimes refer to the replicate samples as simply replicates. For the NYCHVS, we used a total of 80 replicates to calculate the NYCHVS variance estimates.

The replicate weights should only be used in creating variances and should not be used to create independent estimates. The final weights (FW) are provided to produce all estimates.

The user should also note that the replicate weights are calculated using information from the sample. Therefore, the 2017 NYCHVS replicate weights are applicable for use on only 2017 NYCHVS data. Replicate weights for prior years, for example, 2014 are applicable for use with the 2014 public use file only.

Use of Replicate Estimates in Variance Calculations

Calculate variance estimates using the replication variance estimator

$$\hat{v}(\hat{\theta}) = \frac{4}{80}\sum_{r=1}^{80}(\hat{\theta}_r - \hat{\theta}_0)^2 \qquad (2.1)$$

Where $\hat{\theta}$ is the weighted estimate of the statistic of interest; such as a total, median, mean, proportion, regression coefficient, or log-odds ratio, using the weight for the sample and $\hat{\theta}_r$ is the replicate estimate for replicate *r* of the same statistic using the replicate weights. The estimator $\hat{\theta}_0$ is the estimate of θ. With replicate variances, $\hat{\theta}_0$ can be chosen in two ways: either as the estimator of θ using the final weights or as the mean of the replicate estimators. Throughout this document, we suggest using the regular estimate that used the final weights since it produces a more conservative estimate of the variance.

The value of 80 in Equation (2.1) is the number of replicates used by NYCHVS and for more explanation about the factor of 4, see the end of Section 8. See also Fay and Train (1995), Ash (2014), and Opsomer, Breidt, White, and Li (2016) for more background about Successive Difference Replication (SDR).

To ensure confidentiality of the data, some characteristics have either been bottom coded or top coded. This procedure places a lower (or upper) boundary on the published value for the variable in question. Therefore, some estimates calculated from the Public Use File may differ from the estimates provided in the NYCHVS publication tables.

Using Replication to Estimate Variances

The following example illustrates how a statistic would be estimated, replicated, and combined to form a variance estimate. We are going to estimate the variance using the 80 replicate weights provided for the NYCHVS.

Note that in 2017 NYCHVS, the replicate weights of Replicate 1 are equal to the final weights. The same is not true with 2011 and 2014: the replicate weights of replicate 1 are different than the final weights.

The following example illustrates how a statistic would be estimated, replicated, and combined to estimate a variance.

Example 2.1. Estimating the Variance of the Total Number of Housing Units in a Domain

The goal of this example is to estimate the total number of occupied housing units in NYC for 2017 and its corresponding estimate of variance. In 2017, we have 13,266 completed interviews that are either owner or renter occupied housing units in NYC. Table 2.1 displays the first four and last interview responses for occupied housing units in NYC for 2017.

Table 2.1: Example of Estimating Variances with Replication

Sample Housing Unit	Tenure	Final Weight	Replicate Weights				
			Replicate 1	Replicate 2	Replicate 3		Replicate 80
1	Owner	340.405	340.405	95.036	287.495	…	137.400
2	Renter	245.771	245.771	240.316	70.290	…	289.403
3	Renter	245.771	245.771	240.316	70.290	….	289.403
4	Renter	370.417	370.417	365.079	366.310	…	667.296
…	…	…	…	…	…		…
N (13,266)	Renter	4.753	4.753	5.064	4.397	…	4.467

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey Public Use Files.

In the 2017 NYCHVS, the estimate from the final weight are sometimes referred to as replicate estimate 0 or replicate weight 0.

Step 1: Calculate the weighted survey estimate.

The statistic of interest is the total number of occupied housing units in NYC for 2017. Add the final weights of the sample cases that are either renter or owner occupied. Therefore, the estimate of total number of occupied housing units is calculated as follows:

Occupied HUs Estimate $\widehat{N}_0$= 340.405 + 245.771 + … + 4.753 = 3,109,954.63

Step 2: Calculate the weighted survey estimate for each of the replicate samples.

The replicate estimates of occupied HUs are:

Replicate estimate 1 $\widehat{N}_{r=1}$= 340.405 + 245.771 + … + 4.753 = 3,109,954.63
Replicate estimate 2 $\widehat{N}_{r=2}$= 95.036 + 240.316 +… + 5.064 = 3,109,920.64
Replicate estimate 3 $\widehat{N}_{r=3}$= 287.495 + 70.290 + … + 4.397 = 3,105,282.02
⋮ ⋮
Replicate estimate 80 $\widehat{N}_{r=80}$= 137.400 + 289.403 + … + 4.467 = 3,114,050.64

Step 3: Use these survey estimates in Equation (2.1) to calculate the variance estimate for the total occupied housing units.

$$\begin{aligned}\hat{v}(\widehat{N}) &= \frac{4}{80}\sum_{r=1}^{80}\left(\widehat{N}_r - \widehat{N}_0\right)^2 \\ &= 0.05 \times [(3{,}109{,}954.63 - 3{,}109{,}954.63)^2 + (3{,}109{,}920.64 - 3{,}109{,}954.63)^2 \\ &\qquad + (3{,}105{,}282.02 - 3{,}109{,}954.63)^2 + \cdots \\ &\qquad + (3{,}114{,}050.64 - 3{,}109{,}954.63)^2] \\ &= 0.05 \times [0 + 1155.32 + 21{,}833{,}284.21 + \cdots + 16{,}777{,}297.92] \\ &= 74{,}422{,}895.83\end{aligned}$$

The estimate of the variance of total occupied population is $\hat{v}(\widehat{N})$ = 74,422,896.

The survey estimate for occupied housing units in NYC is 3,109,954. This survey estimate has an estimated variance of 74,422,896, and its standard error, which is the square root of the estimated variance, is 8,627 housing units.

The three steps of Example 2.1 will be used throughout the guide to calculate variances. Sometimes Steps 1 and 2 will be combined since estimating the statistic with the final weight (or for replicate 0) can be done with estimating the statistic with the 80 replicate weights.

Cautions about Domain Analysis Don't Apply

Korn and Graubard (1999; Section 5.4), Lewis (2017; Section 8), Heeringa, West, and Berglund (2010; Section 4.5) and others have provided important cautions about the analysis of domains, sometimes also referred to as subdomains. These cautions are important but do not apply to the analysis described within this Guide since our variance estimation employs replicate weights in order to estimate variances. When using the Taylor series to estimate variances, the default method of most software packages, the issue of domains is important: SAS will generate incorrect estimates with the `where` statements and correct estimates with the `domain` statement. Similarly with STATA, the **`subpop()`** option is needed. However, with the replicate weights, using the `domain` or **`subpop()`** or `where` statements with SAS or STATA will produce the same results.

Confidence Intervals and Significance Tests

Once the standard error is calculated, it can be combined with the estimate to calculate confidence intervals. Section 7 provides further instructions on how this can be done.

NYCHVS Public Use File Description

To access the PUF, go to the Census Bureau's NYCHVS website at: https://www.census.gov/programs-surveys/nychvs/data/datasets.html . Once navigation to the site is completed, then select the "2017" tab and select the "2017 New York City Housing and Vacancy Survey Microdata". At the bottom of the resulting webpage, there are five ACSII text files and four PDF files.

The five text files:

1) SAS Import Program, which give the SAS code you need to read in each of the three record files (text files #3 – #5) into SAS,
2) STATA Import Program, which give the Stata code you need to read in each of the three record files (text files #3 – #5) into Stata,
3) Occupied Records, which contains the records for occupied housing units,
4) Person Records, which contains the records for persons, and
5) Vacant Records, which contains the records for vacant housing units.

The four PDF files:

1) 2017 Table of Contents,
2) 2017 Household Records – Occupied Units,
3) 2017 Person Records – Occupied Units, and
4) 2017 Household Records – Vacant Units.

These PDF files provide the record layout of each dataset listed above (text files #3 - #5).

The datasets Occupied Records, Person Records, and Vacant Records (text files #3, #4, and #5 respectively) contains the sample estimate with full sample weights as well as the 80 replicate weights for occupied housing units, persons, and vacant housing units respectively. NYCHVS data users please use the Occupied Records and Vacant Records dataset for any housing unit estimates, and use the Person Records dataset for any person estimates.

Data preparation for Running SAS

For the NYCHVS, we used a total of 80 replicates to calculate the NYCHVS variance estimates. When reading these files (text files #3 - #5) into SAS, first download and save these data files as txt files. Then download the SAS import program (text file #1), update the input-related items in the code (highlighted in yellow in Figure 2.1.1). Run the part of the code for Occupied Records, and/or Person Records, and/or Vacant Records, then local SAS datasets will be created. Figure 2.1.1 shows the top part of the code for reading in Occupied HUs data file (text file #2). For generating SAS datasets for Person Records or Vacant Records (text file #4 and #5 respectively), just follow the same steps for Occupied Records.

Figure 2.1.1: Partial SAS Code for Reading in the Public Use Files

```
* Import the Occupied data file. ;
data occupied17;
infile 'LOCATION OF TEXT FILE\uf_17_occ_web_b.txt' lrecl=1334 truncover;
input recid $1 @;
if(recid='1') then do;

***SEE THE REST OF THE SAS CODE IN SAS IMPORT PROGRAM FOUND IN THE CENSUS
BUREAU'S NYCHVS WEBSITE MENTIONED IN PREVIOUS PAGE*** ;
```

For housing unit estimates, users need to download both Occupied Records (text file #3) and Vacant Records (text file #5), and then combine the two files into one. Figure 2.1.2 shows the SAS code for importing vacant records, and then combining it with the occupied records.

Figure 2.1.2: SAS Code for Appending Occupied and Vacant Records

```
* Import the Vacant data file. ;
data vacant17;
infile 'LOCATION OF TEXT FILE\uf_17_vac_web_b.txt' lrecl=831 truncover;
input recid $1 @;
if(recid='3') then do;

***SEE THE REST OF THE SAS CODE IN SAS IMPORT PROGRAM FOUND IN THE CENSUS
BUREAU'S NYCHVS WEBSITE MENTIONED IN PREVIOUS PAGE *** ;

* Combined Occupied and Vacant records for 2017;
Data HU_all17;
      set occupied17
          vacant17;
run;
```

For prior year's NYCHVS data files, data users can go to the same website mentioned on page 7, click the "2014" or "2011" tab, and download the PUFs from there. We used the same procedure provided for 2017 Occupied Records, updated the statements for the 2014 files and locations. We saved the 2014 occupied HU SAS data file as occupied14, and the 2014 vacant HU SAS data file as vacant14. The 2014 combination file for all housing units is called HU_all14. Figure 2.1.3 shows the partial SAS code for reading in the public use files for 2014.

Figure 2.1.3: Partial SAS Code for Reading in the Public Use Files for 2014

```
* Import the Occupied data file for 2014. ;
data occupied14;
infile 'LOCATION OF TEXT FILE\uf_14_repwgt_occ_web.txt' lrecl=1321
truncover;
input recid $1 @;
if(recid='1') then do;

***SEE THE REST OF THE SAS CODE IN SAS IMPORT PROGRAM FOUND IN THE CENSUS
BUREAU'S NYCHVS WEBSITE MENTIONED IN PAGE 7*** ;

*Import VACANT DATA FILE;
data vacant14;
infile 'LOCATION OF TEXT FILE\uf_14_repwgt_vac_web.txt' lrecl=831 truncover;
input recid $1 @;
if(recid='3') then do;


***SEE THE REST OF THE SAS CODE IN SAS IMPORT PROGRAM FOUND IN THE CENSUS
BUREAU'S NYCHVS WEBSITE MENTIONED IN PAGE 7*** ;

*Combined Occupied and Vacant records for 2014;
Data HU_all14;
      set occupied14
          vacant14;
run;
```

For the housing unit data file, the final weight is stored in variable FW, and the replicate weights are stored in variables FW1-FW80. For the persons data file, the final weight is stored in variable PW, and the replicate weights for persons are stored in variables PW1- PW80. These weights are stored as character string with five implied decimal places, so before running any analysis, they need to be converted. Figure 2.1.4 shows the SAS code for converting the weights to numeric.

Figure 2.1.4: SAS Code for Converting Weights to Numeric

```
* convert weights to numeric with the correct decimal place. ;
%macro wgt(yr)  ;
Data HU&yr.;
      set HU_all&yr.;
      fw0=fw/100000;
  %do i=1 %to 80 ;
      fw&i=fw&i /100000 ;
      fw_&i=input(fw&i, 9.);
      drop fw&i;
      rename fw_&i=fw&i;
    %end ;
run;
%mend wgt ;

%wgt(17);
%wgt(14);
```

After running the code above, dataset HU17 and HU14 are created. These two datasets will be used throughout this document.

3. Estimating Variances with SAS PROC SURVEY Procedures

Within SAS, there are currently four specialized 'survey' procedures that use replicate weights directly. Table 3 reviews the SAS SURVEY procedures.

Table 3: Summary of SAS 'SURVEY' PROCs

PROC name	**Can be used to…**
SURVEYMEANS	Calculate basic statistics
SURVEYFREQ	Complete categorical data analysis
SURVEYREG	Complete regression analysis
SURVEYLOGISTIC	Complete logistic regression analysis

The equivalent procedures in other statistical software packages will be discussed at the end of the section and some examples using Base SAS, Stata and R are provided in Sections 4, 5 and 6.

Because the replicate weights for NYCHVS are calculated with Fay's SDR methods, the SAS SURVEY procedures need to use the `varmethod=brr(fay)` option in the PROC statement. If the fay option for BRR is not used, the variances will be off by a factor of four.

An unintended consequence of using our replicate weights with the `varmethod=brr(fay)` option is that all of the SAS PROC SURVEY procedures assume that BRR is being used and further that the number of replicates is the number of strata in the sample design. The replicate weights that are provided for NYCHVS use SDR methodology that is appropriate for the NYCHVS sample design: systematic random sample from an ordered list. As a result, the confidence intervals will use a critical value from a t-distribution with 80 degrees of freedom. We, however, suggest using a critical value from a normal distribution – see also "Normal Distribution versus the *t*-distribution for Confidence Intervals" in Section 7.

Mukhopadhyay, An, Tobias, and Watts (2008) provides an excellent review of replication-based variances methods and the "survey" procedures of SAS. Taylor (2016) also provides a comprehensive review of using SAS to analyze survey data.

Each of the procedures of Table 3 can use the replicate weights in its analysis. For example, the SURVEYMEANS procedure and the replicate weight file can be used together to generate a standard error for a population total estimate, as shown in Figure 3.1. Throughout this document, input related items are highlighted in yellow.

Figure 3.1: SAS Code Using PROC SURVEYMEANS

```
proc surveymeans data=HU17 sum std clsum cvsum varmethod=brr(fay);
  var variable ;
  weight fw0 ;
  repweights fw1-fw80 ;
run ;
```

Since we are most familiar with SAS, the rest of this section and the next will provide examples using SAS. In Sections 4 and 5 we repeat some of the examples using Stata and R. We do recommend the reader refer to their software documentation for further information on the use of replicate weights in variance estimation for any other statistical packages. Alternatively, readers can contact the software specific customer support.

Example 3.1. Estimating the Variance of a Mean with SAS PROC SURVEYMEANS

In this example, we estimate the average gross rent for renter-occupied housing units. The output for PROC SURVEYMEANS of SAS will also include the estimated standard error of the average gross rent of renter-occupied HUs.

Figure 3.1.1 provides the SAS code that can be used to estimate the average gross rent of the renter-occupied HUs from the 2017 NYCHVS.

Figure 3.1.1: Example of SAS Code Using PROC SURVEYMEANS

```
Data Data1 ;
  set HU17 ;
  * Convert invalid rent records, covert character variable to numeric. ;
  if uf26 eq '99999' then rent = . ;
  else rent=input(uf26,8.);
  * Define domains. ;
  if sc116 in ('2','3') then tenure='Renter' ;
  else if sc115='1' then tenure='Owner' ;
run ;

proc surveymeans data=Data1 mean std varmethod=brr(fay) ;
  domain tenure ;
  var rent ;
  weight fw0 ;
  repweights fw1-fw80 ;
run ;
```

The SAS code of Figure 3.1.1 generates the output of Figure 3.1.2.

Figure 3.1.2: SAS Output for PROC SURVEYMEANS

Statistics for tenure Domains

tenure	Variable	Mean	Std Error of Mean
Renter	rent	1693.538796	10.580511

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The estimate of the average gross rent in renter occupied HUs is $1,694 with a standard error of $11.

Example 3.2. Estimating the Variance for a Contingency Table with SAS PROC SURVEYFREQ

In this example, we estimate the frequency and proportion of rent stabilization status (pre-1947 or post-1947) with maintenance deficiencies. The output for PROC SURVEYFREQ will also include the standard errors of the estimated quantities. Figure 3.2.1 provides the SAS code for PROC SURVEYFREQ. See Section 7 for this same example on calculating odds ratios.

Figure 3.2.1 provides the SAS code that produces the two-way table of building deficiencies by rent stabilization status.

Figure 3.2.1: Example of SAS Code Using PROC SURVEYFREQ

```
Data Data2 ;
  set HU17 ;
  if rec53 in ('4','5','6','7','8') then def='1' ;
  * Redefine HUs that did not report on deficiencies. ;
  else if rec53 ='9' then def=' ';
  else def='0';
  if uf_csr='30' then stabilized='pre-1947' ;
  else if uf_csr ='31' then stabilized='post1947';
  * Redefine HUs w/ other CSR code. ;
  else stabilized =' ';

  * Define variable TENURE that identifies the domains Renter and Owner. ;
  if sc116 in ('2','3') then tenure='Renter';
  else if sc115='1' then tenure='Owner';
run;

proc surveyfreq data=data2  varmethod=brr(fay) NOSUMMARY ;
  tables tenure*def*stabilized ;
  weight fw0 ;
  repweights fw1-fw80 ;
run ;
```

The SAS code of Figure 3.2.1 generates the output seen in Figure 3.2.2.

Figure 3.2.2: SAS Output for PROC SURVEYFREQ

The SURVEYFREQ Procedure

Variance Estimation

Method	BRR
Replicate Weights	DATA2
Number of Replicates	80
Fay Coefficient	0.500

Table of Def by Stabilized

Controlling for tenure=Renter

Def	Stabilized	Frequency	Weighted Frequency	Std Err of Wgt Freq	Percent	Std Err of Percent
0	post1947	888	214719	8110	25.6280	0.8476
	pre-1947	2007	474343	13632	56.6158	1.0282
	Total	2895	689062	16064	82.2438	0.8313
1	post1947	111	26435	2487	3.1552	0.3082
	pre-1947	517	122331	5823	14.6010	0.7043
	Total	628	148767	6681	17.7562	0.8313
Total	post1947	999	241154	8181	28.7832	0.8797
	pre-1947	2524	596674	13774	71.2168	0.8797
	Total	3523	837829	15216	100.0000	

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Example 3.3. Estimating the Variance of a Regression Coefficients with SAS PROC SURVEYREG

Correctly estimating the variance of a regression coefficient can be complicated depending on which procedure is used. This example will show a regression model modeling the 2017 gross rent of a renter-occupied housing unit using the year householder moved in to the housing unit. The regression model will first be shown using PROC SURVEYREG with correct parameter estimates and variances. Next, the regression model will be shown using PROC REG with the correct parameter estimates but will generate incorrect variances. Last, we demonstrate how not using any weighting produces incorrect results. Moreover, in Section 4 Example 4.5 we calculate the regression parameter directly.

First, PROC SURVEYREG will be used to generate the correct parameter estimates and variances. Figure 3.3.1 shows how this can be done.

Figure 3.3.1: SAS SURVEYREG Code for Estimating Variance of a Regression Parameter

```
Data Data3 ;
  set HU17 ;
  * Convert invalid rent records, covert character variable to numeric. ;
  if uf26 eq '99999' then rent = . ;
  else rent=input(uf26,8.);
  yr_movein=input(uf66,8.);
  * Define domains. ;
  if sc116 in ('2','3') then tenure='Renter';
  else if sc115='1' then tenure='Owner';
run;

proc surveyreg data=Data3 varmethod=brr(fay) ;
  domain tenure;
  model rent = yr_movein / solution ;
  weight fw0 ;
  repweights fw1-fw80 ;
  ods output parameterEstimates = MyParmEst ;
run;

data MyParmEstfin ;
  set MyParmEst ;
  * Keep the second observation, which is renters. ;
  if _n_ = 2 ;
  se = stderr;
  var = se**2 ;
  drop parameter dendf tvalue probt stderr ;
run ;

proc print data=MyParmEstFin ( keep = estimate se ) noobs ;
  format estimate se 8.4 ;
run ;
```

The SAS code of Figure 3.3.1 produces the output of Figure 3.3.2.

Figure 3.3.2: SAS SURVEYREG Output for Estimating Variance of a Regression Parameter

Estimate	se
28.7885	0.8122

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Using the `weight` statement in SAS is not a shortcut. We now show how PROC REG and the final weights estimate the correct parameter estimates but overstate the variances because it does not correctly account for the two-stage sample design of NYCHVS. Figure 3.3.3 shows how this can be done.

Figure 3.3.3: SAS REG Code for Estimating Variance of a Regression Parameter

```
* Weighted -- Using the sample design weights.;
proc reg data=Data3 ;
  where tenure='Renter' ;
  model rent = yr_movein ;
  weight fw0 ;
  ods output parameterEstimates = MyParmEstw ;
run ;

data MyParmEstwFin ;
  set MyParmEstw ;
  if _n_ = 2 ;
  drop model dependent variable df tvalue probt ;
run;

proc print data=MyParmEstwFin ; run ;
```

The SAS code of 3.3.3 generates the output of Figure 3.3.4.

Figure 3.3.4: SAS REG Output for Estimating Regression Parameters

Estimate	StdErr
28.78853	0.90248

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

As expected, the estimate of the regression parameter in Figures 3.3.4 and 3.3.2 agree, but the standard error using PROC REG and the `weight` statement overestimates the standard error by 11 percent or $\left(1 - \frac{0.9025}{0.8122}\right)$.

Next, we show the worst possible case. We show what happens when PROC REG is used without final sample weights to generate an unweighted modeled parameter estimate with its corresponding standard error. Figure 3.3.5 shows how this can be done.

Figure 3.3.5: SAS Code for Estimating Variance of a Regression Parameter

```
* Unweighted – Not using the sample design weights. ;

proc reg data=Data3 ;
  where tenure='Renter' ;
  model rent = yr_movein ;
  ods output parameterEstimates = MyParmEstu ;
run ;

data MyParmEstuFin ;
  set MyParmEstu ;
  if _n_ = 2 ;
  drop model dependent variable df tvalue probt ;
run;

proc print data = MyParmEstuFin ; run ;
```

The SAS code of Figure 3.3.5 generates the output of Figure 3.3.6.

Figure 3.3.6: SAS Output for Estimating Variance of a Regression Parameter

Estimate	StdErr
29.67853	0.87447

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Estimating the regression parameter and its variance without the weights produces an incorrect estimate of the parameter and an overestimate of the variance.

Statistical Software for Calculating Variances

Several statistical software packages employ replicate weights and Equation (2.1). We now review a few of the software packages that we know about.

R Software: The '`survey`' package allows users to estimate variances by Taylor Series linearization or replicate weights.

Stata: A good review of the use of STATA is given by Kreuter and Valliant (2007).

Wesvar: This software is developed and distributed by Westat and can use replicate weights directly to estimate variances of a complex survey design. This document does not provide guidance for using Wesvar.

4. Examples of Calculating Variances with Base SAS

This section includes examples of calculating the variance of a total, difference, mean, median, regression parameter, longitudinal change and percent change within a complex survey design by using replicate weights. Unlike the previous section where we used the SAS PROC SURVEY procedures, here we calculate all values directly using programming within base SAS (the DATA step, PROC SORT, and PROC MEANS). This section is provided for data users who would rather code the variance estimation methods themselves rather than use the PROC SURVEY procedures provided by SAS. In almost all cases, the estimates calculated using SAS PROC SURVEY and the estimates from base SAS are the same and when they differ, we explain those differences.

Example 4.1. Estimating the Variance of a Total

In Example 2.1, we generally demonstrated how to estimate the variance for the occupied housing units in NYC using replicate weights. We now show how to estimate the same total directly with the replicate weights.

For our example, the domain of interest is all occupied housing units in NYC for 2017. In Figure 4.1.1, we start by keeping only those sample HUs of interest; the 13,266 completed interviews who are either owners or renters.

Figure 4.1.1: SAS Code for Flagging Desired Records

```
* Flag the data records desired after creating the SAS data sets.
  This example flags both owner- or renter-occupied housing units. ;

data Data4 ;
  set HU17 ;
  where recid="1" ;
run ;
```

Next, Figure 4.1.2 shows how we can calculate the sample estimate and replicate estimates of the number of occupied housing units in the NYC in 2017.

Figure 4.1.2: SAS Code for Estimating Variance of a Total (Step 1 &2)

```
* Steps 1 & 2: Sum the sample and the 80
  replicate weights and writes them out to a file ;

proc means data=Data4 sum noprint ;
  * The sample and the replicates. ;
  var fw0 fw1-fw80 ;
  output out=Data5 sum=est rw1-rw80 ;
run ;
```

The final step, represented by Figure 4.1.3, is to apply Equation (2.1) with the sample estimate and replicate estimates.

Figure 4.1.3: SAS Code for Estimating Variance of a Total (Step 3)

```
* Step 3: Use the sample estimate and the 80 replicate
  estimates to compute the estimated replicate variance(s)
  using the Equation 2.1 for 80 replicates. ;

data Data6 (keep = est var se) ;
  set Data5 end=eof ;
  * Fill array with the replicate sums. ;
  array repwts{80} rw1-rw80 ;
  * Fill array with the squared diffs. ;
  array sdiffsq{80} sdiffsq1-sdiffsq80 ;
  do j = 1 to 80 ;
    sdiffsq{j} = (repwts{j} - est)**2 ;
    end ;
  * Sum the squared diffs. ;
  totdiff = sum(of sdiffsq1-sdiffsq80) ;
  var = (4/80) * totdiff ;
  se = (var)**(0.5) ;
  output ;
run ;

proc print data=Data6 noobs ;
  var est  se ;
  format est se comma12.0 ;
run ;
```

The SAS code of Figures 4.1.1 - 4.1.3 generates the output of Figure 4.1.4.

Figure 4.1.4: SAS Output for Estimating Variance of a Total

est	se
3,109,955	8,627

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

In this example, the estimate of the number of occupied HUs in NYC for 2017 is 3,109,955 with an estimated standard error of 8,627. These results from Example 4.1 match exactly with the results we got from Example 2.1.

Example 4.2. Estimating the Variance of a Difference

This example demonstrates how to estimate the variance of a difference. The specific statistic of interest is the difference between the number of rent stabilized units in Manhattan and the number of rent stabilized units in the Bronx for 2017.

First, input the data file and keep those sample units in the domain of interest. For our example, the domains of interest are 1) the rent stabilized units in Manhattan, and 2) the rent stabilized units in the Bronx. In Figure 4.2.1, we start by keeping only those sample HUs in the domains of interest. For 2017, there are 1,058 completed interviews that are rent stabilized units in Manhattan, and 952 completed interviews that are rent stabilized in the Bronx.

Figure 4.2.1: SAS Code for Flagging Domain of Interest

```
* Subset to only rent stabilized units in Manhattan. ;
data data7 ;
  set HU17 ;
  where boro='3' and uf_csr in ('30', '31') ;
run ;

* Subset to only rent stabilized units in the Bronx. ;
data Data8 ;
  set HU17;
  where boro='1' and uf_csr in ('30', '31') ;
run;
```

Next, Figure 4.2.2 shows how to calculate the sample estimate and replicate estimates of the number of rent stabilized housing units in Manhattan, as well as the Bronx. Then, we merge those files to get the differences.

Figure 4.2.2: SAS Code for Estimating Variance of a Difference

```
* Manhattan;
proc means data=data7 sum noprint ;
  * The sample estimate and the replicate estimates. ;
  var fw0 fw1-fw80 ;
  output out=data9 sum=est1 rw1-rw80 ;
run ;

* The Bronx. ;
proc means data=data8 sum noprint ;
  var fw0 fw1-fw80 ;
  output out=Data10 sum=est2 rw2_1-rw2_80 ;
run ;

* Merge the two files and get their differences. ;
Data Data11 (keep = diff0 diff1-diff80) ;
  merge data9 Data10 ;
  array diff(80) diff1-diff80 ;
  array rw2_(80) rw2_1-rw2_80 ;
  array rw(80) rw1 - rw80 ;
  diff0=est1-est2 ;
  do i=1 to 80 ;
    diff(i)=rw(i)-rw2_(i) ;
    end;
run ;
```

Last, Figure 4.2.3 shows to apply Equation (2.1) to calculate the sample estimate and the replicate estimates of the difference.

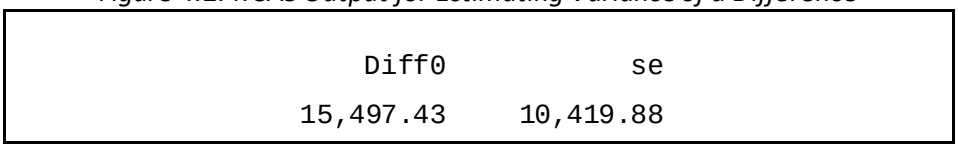
Figure 4.2.3: SAS Code for Estimating Variance of a Difference

```
data Data12 (keep=diff0 var se) ;
  set Data11 end=eof ;
  * Fill array with the replicate means ;
  array diff{80} diff1-diff80 ;
  * Fill array with the squared diffs. ;
  array sdiffsq{80} sdiffsq1-sdiffsq80 ;
  do j = 1 to 80 ;
    sdiffsq{j} = (diff{j} - diff0)**2 ;
    end ;
  * Sum the squared diffs. ;
  totdiff = sum(of sdiffsq1-sdiffsq80) ;
  var = (4/80) * totdiff ;
  se = var**(0.5) ;
  output ;
run ;

proc print data=Data12 noobs ;
  var diff0  se ;
  format diff0  se comma15.2 ;
run ;
```

The SAS code of Figures 4.2.1 -4.2.3 generates the output of Figure 4.2.4.

Figure 4.2.4: SAS Output for Estimating Variance of a Difference



Diff0	se
15,497.43	10,419.88

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The difference between the number of Manhattan and the Bronx housing units in 2017 that are renter-occupied is 15,497, with an estimated standard error of 10,420.

In Example 4.2, we show how to estimate the variance of a difference. Next, in Example 4.3, we will show how to estimate the variance of a mean. The variable of interest will be the average gross rent of renter-occupied housing units.

Example 4.3. Estimating the Variance of a Mean

An estimated ratio from a survey, defined as $\hat{R} = \hat{Y}/\hat{X}$, is a nonlinear statistic of two estimated totals $\hat{Y}$ and $\hat{X}$. BRR is especially suited for just this. To estimate the variance of a mean $\hat{\bar{Y}} = \sum \hat{Y}/\hat{N}$, we note that a mean is a special case of the ratio estimator. Users interested in estimating the variance of a proportion, please refer to Example 4.6 for more detail.

This example demonstrates how to estimate the variance of a mean. The specific statistic of interest is the average gross rent for renter-occupied housing units in the NYC for 2017. The first step of calculating the variance of a mean with replication is to calculate the mean for each replicate. Figure 4.3.1 shows how this can be done.

Figure 4.3.1: SAS Code for Estimating Variance of a Mean

```
* Subset the dataset to only renter occupied HUs
  with valid gross rent. ;
data Data13 ;
  set HU17 ;
  where uf26 ne '99999' and sc116 in ('2','3') ;
  rent=input(uf26,8.) ;
run;

* This empty data set is produced for the merge later. ;
data Data14 ;
  length mean0-mean80 8. ;
run;

* Estimate the replicate estimates for 80 replicates
;
%macro repss(rep) ;
proc means data=Data13 mean noprint ;
  weight fw&rep. ;
  var rent ;
  output out=datal&rep. mean=mean&rep. ;
run;

data Data14 ;
  merge Data14 datal&rep. ;
run ;
%mend repss ;

%macro doit;
  %do i=0 %to 80 ;
    %repss(&i.) ;
    %end;
%mend doit ;

%doit ;
```

Finally, we apply Equation (2.1) to calculate the replicate variance. Figure 4.3.2 shows how this can be done with SAS.

Figure 4.3.2: SAS Code for Estimating Variance of a Mean

```
* Apply Step 3 to the replicate estimates and
  estimate the variance. ;
data Data15 (keep=mean0 var se) ;
  set Data14 end=eof ;
  * Fill array with the replicate means ;
  array mean{80} mean1-mean80 ;
  * Fill array with the squared diffs. ;
  array sdiffsq{80} sdiffsq1-sdiffsq80;
  do j = 1 to 80 ;
    sdiffsq{j} = (mean{j} - mean0)**2 ;
    end;
  * Sum the squared diffs. ;
  totdiff = sum(of sdiffsq1-sdiffsq80) ;
  var = (4/80) * totdiff ;
  se = var**(0.5) ;
  output ;
run ;

proc print data=Data15 noobs ;
  var mean0 se ;
  format mean0 se 8.2 ;
run ;
```

The SAS code of Figures 4.3.1-4.3.2 generates the output of Figure 4.3.3.

Figure 4.3.3: SAS Output for Estimating Variance of a Mean

mean0	se
1,693.54	10.58

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The average gross rent for renter-occupied housing units in NYC for 2017 is $1,694 with a standard error of $11.

Example 4.4. Estimating the Variance of a Median

This example demonstrates how to estimate the variance of a median. The specific statistic of interest is the median gross rent for renter-occupied housing units in NYC for 2017.

Estimating the variance of a median is generally the same as a mean in that we need to calculate the median for every replicate and then apply Equation (2.1). The first and second steps are to calculate the sample estimate and the replicate estimates of the median. The third step is to apply Equation (2.1) to the sample estimate and replicate estimates. Figure 4.4.1 shows how this can be done with SAS.

Since the domain of interest is same as previous example, we can use dataset Data13 from Example 4.3.

Figure 4.4.1: SAS Code for Estimating Variance of a Median

```
* This empty data set is produced for the merge later. ;
data Data16 ;
  length med0-med80 8. ;
run;

* Steps 1 & 2: Estimate the replicate estimates for replicates 0 to 80. ;
%macro repss(rep) ;
proc means data=Data13 median noprint ;
  weight fw&rep. ;
  var rent ;
  output out=datal&rep. median=med&rep. ;
run ;

data Data16 ;
  merge Data16 datal&rep. ;
run ;

%mend repss ;
%macro doit ;
  %do i=0 %to 80 ;
    %repss(&i) ;
    %end ;
%mend doit ;
%doit ;

* Apply Step 3 to the replicate estimates and estimate the variance. ;
data Data17 (keep=med0 var se) ;
  set Data16 end=eof ;
  * Fill array with the replicate means ;
  array med{80} med1-med80 ;
  * Fill array with the squared diffs. ;
  array sdiffsq{80} sdiffsq1-sdiffsq80 ;
  do j = 1 to 80 ;
    sdiffsq{j} = (med{j} - med0)**2 ;
    end ;
  * Sum the squared diffs. ;
  totdiff = sum(of sdiffsq1-sdiffsq80) ;
  var = (4/80) * totdiff ;
  se = var**(0.5) ;
  output ;
run ;

proc print data=Data17 noobs ;
  var med0 se ;
run ;
```

The SAS code of Figure 4.4.1 generates the output of Figure 4.4.2.

Figure 4.4.2: SAS Output for Estimating Variance of a Median

med0	se
1450	7.48331

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The estimated median gross rent for renter-occupied housing units in NYC for 2017 is $1,450 with a standard error of $7.

Example 4.5. Estimating the Variance of a Regression Parameter

Instead of using PROC SURVEY to correctly estimate the regression parameter estimates and their variances, this example shows how we can do it directly. We do this by calculating parameter estimates for all replicate weights with PROC REG and the sample weights. This produces the correct replicate estimates. Then we apply Step 3 and calculate the squared differences between the estimates. Figure 4.5.1 shows how this can be done.

Figure 4.5.1: SAS Code for Estimating Variance of a Regression Parameter

```
data Data18 ;
  set HU17 ;
  * Subset dataset to renter occupied units with valid rent. ;
  where sc116 in ('2','3')  and uf26 not in ('99999') ;
  * Convert variables to numeric. ;
  rent=input(uf26,8.) ;
  yr_movein=input(uf66,8.)  ;
run ;

* This empty data set is produced for the merge later. ;
data Data19 ;
  length parmest0-parmest80 8. ;
run ;

* Steps 1 & 2: Estimate the replicate estimates for replicates 0 to 80;
%macro repss(rep) ;
proc reg data=Data18  noprint;
  model rent = yr_movein ;
  weight fw&rep. ;
  ods output parameterEstimates = MyParmEst&rep. ;
run ;

data myparmest&rep.a ;
  set myparmest&rep. ;
  rename estimate=parmest&rep. StdErr=se&rep. ;
  drop model dependent df tvalue probt ;
  if _n_=2 then output ;
run ;

data Data19 ;
  merge Data19 myparmest&rep.a ;
```

Figure 4.5.1: SAS Code for Estimating Variance of a Regression Parameter

```
run ;

%mend repss ;
%macro doit ;
  %do i=0 %to 80 ;
    %repss(&i.) ;
    %end ;
%mend doit ;
%doit ;

* Apply Step 3 to the replicate estimates and estimate the variance.;
data data20 (keep=parmest0 var se) ;
  set Data19 end=eof ;
  * Fill array with the replicate means ;
  array parmest{80} parmest1-parmest80 ;
  * Fill array with the squared diffs. ;
  array sdiffsq{80} sdiffsq1-sdiffsq80 ;
  do j = 1 to 80 ;
    sdiffsq{j} = (parmest{j} - parmest0)**2 ;
    end ;
  * Sum the squared diffs. ;
  totdiff = sum(of sdiffsq1-sdiffsq80) ;
  var = (4/80) * totdiff ;
  se = var**(0.5) ;
  output ;
run;

proc print data=data20 noobs ;
  var parmest0  se;
  format parmest0  se comma8.4 ;
run ;
```

The SAS code of Figure 4.5.1 generates the output of Figure 4.5.2.

Figure 4.5.2: SAS Output for Estimating Variance of a Regression Parameters

parmest0	se
28.7885	0.8122

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The standard error for the regression estimator in Figure 4.5.2 is the same as estimate of Example 3.3.

Example 4.6. Estimating the Variance of a Longitudinal Change

Both this example and the next consider statistics that measure longitudinal change. This example will consider a statistic that measures the change of a rate between two cycles of the estimates. Example 4.7 will consider an estimate of percent of change that is calculated at the

HU level. We present these two longitudinal statistics as examples thinking these calculations are the most likely of calculations that data users would be interested in.

In this example, we show how to estimate the variance of a difference in proportion. Specifically, we are interested in the estimated vacancy rate or $\hat{p}_t = \hat{X}_t / \widehat{N}_t * 100$. Where $\hat{X}_t$ is the estimator of the total number of vacant HUs at time *t*, and $\widehat{N}_t$ is the estimator of the total number of HUs at time *t*.

Further, we're really interested in the difference in the vacancy rate between 2017 and 2014. So the statistic of interest is:

$$\hat{\Delta}_{t=2017} = \frac{\hat{X}_{t=2017}}{\widehat{N}_{t=2017}} * 100 - \frac{\hat{X}_{t=2014}}{\widehat{N}_{t=2014}} * 100$$

To estimate the variance of $\hat{\Delta}_t$ with replicate weights, we first calculate all the pieces of the $\hat{\Delta}_{t=2017}$ for each of the 80 replicates, making 80 replicate estimates of $\hat{\Delta}_{t=2017}$, repeat those steps for 2014, and then we apply Equation (2.1). Figure 4.6.1 shows how we do this with SAS.

Figure 4.6.1: SAS Code for Estimating Variance of a Longitudinal Change

```
*Define Vacant and Occupied HUs. ;
data temp1 ;
  set HU14 ;
  if recid="1"  then type='1'; * Occupied HUs;
  else if recid="3"   then type='2'; * X-hat 2014* vacant HUs ;
  * Type 1 & 2 makes total number HUs, N-hat ;
run;

data temp2 ;
  set HU17 ;
  if recid="1" then type='1' ; * Occupied HUs ;
  else if recid="3"  then type='2' ; * X-hat 2017* vacant HUs ;
  * Type 1 & 2 makes total number HUs, N-hat;
run;

* This empty data set is produced for the merge later. ;
data data21 ;
  length diff0-diff80 8. ;
run ;

* Estimate the replicate estimates for replicates 0-80. ;
%macro repssss(rep) ;
* Get denominator N-hat 2014. ;
proc means data= temp1 noprint ;
  where type in ('2','1') ;
  var fw&rep.;
  output out=den14rep&rep. sum=den14rep&rep. ;
run ;
```

Figure 4.6.1: SAS Code for Estimating Variance of a Longitudinal Change

```
* Get numerator X-hat 2014. ;
proc means data= temp1 noprint ;
  where type in ('2') ;
  var fw&rep.;
output out=num14rep&rep. sum=num14rep&rep. ;
run ;

* Get denominator N-hat 2017. ;
proc means data= temp2 noprint ;
  where type in ('2','1') ;
   var fw&rep.;
  output out=den17rep&rep. sum=den17rep&rep. ;
run ;

* Get numerator X-hat 2017. ;
proc means data= temp2 noprint ;
  where type in ('2') ;
  var fw&rep.;
  output out=num17rep&rep.  sum=num17rep&rep.;
run ;

* Merge the replicate estimates of each year, and get difference
  in vacancy rate by replicate. ;
data datamrep&rep.;
  merge num17rep&rep. num14rep&rep. den17rep&rep. den14rep&rep. ;
  diff&rep. = (num17rep&rep./den17rep&rep.)*100 -
(num14rep&rep./den14rep&rep.)*100 ;
  keep diff&rep.;
run ;

data data21 ;
  merge data21 datamrep&rep. ;
run ;

%mend repssss ;

%macro doit ;
  %do i=0 %to 80 ;
    %repssss(&i.) ;
    %end;
%mend doit ;
%doit ;

* Apply Step 3 to the replicate estimates and estimate the variance. ;
data data22 (keep=diff0 var se) ;
  set data21 end=eof ;
  * Fill array with the replicate means. ;
  array diff{80} diff1-diff80 ;
  * Fill array with the squared diffs. ;
  array sdiffsq{80} sdiffsq1-sdiffsq80 ;
  do j = 1 to 80 ;
    sdiffsq{j} = (diff{j} - diff0)**2 ;
    end ;
  * Sum the squared diffs. ;
  totdiff = sum(of sdiffsq1-sdiffsq80) ;
```

Figure 4.6.1: SAS Code for Estimating Variance of a Longitudinal Change

```
  var = (4/80) * totdiff ;
  se = (var)**(0.5) ;
  output ;
run ;

proc print data=data22 noobs ;
  var diff0  se ;
  format diff0 se 8.4 ;
run ;
```

The SAS code of Figure 4.6.1 produces the output in Figure 4.6.2.

Figure 4.6.2: SAS Output for Estimating Variance of a Longitudinal Change

diff0	se
2.2402	0.2588

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Thus, the total vacancy rate in NYC for 2017 has increased 2.2 percent from 2014 with a standard error of 0.3 percent.

Example 4.7. Estimating the Variance of a Percent Change in Two Cycles of NYCHVS

There are many different statistics that measure the change between two cycles of NYCHVS. Differences, percent change, and ratios can be used to measure how much a given statistic changed from one cycle of NYCHVS to another. This example considers the housing characteristic of the median gross rent and how it can change over time. Both the rate of change and resulting variance are demonstrated.

The statistic of interest is the percent change in the median gross rent from 2014 to 2017. Let $\widehat{M}_t$ the estimator of the median gross rent at time *t*. The statistic of interest is

$$\%\hat{\Delta}_t = \frac{\widehat{M}_t - \widehat{M}_{t-1}}{\widehat{M}_{t-1}}$$

To estimate the variance of $(\Delta\%)_t$, we use the 2014 replicate weights and calculate 80 replicate estimates of $\widehat{\mathrm{M}}_{\mathrm{t=2014}}$ and similarly use the 2017 replicate weights and calculate 80 replicate estimates of $\widehat{\mathrm{M}}_{\mathrm{t=2017}}$. Next, we merge the replicate estimates of $\widehat{\mathrm{M}}_{\mathrm{t=2014}}$ and $\widehat{\mathrm{M}}_{\mathrm{t=2017}}$ by replicate and calculate 80 replicate estimates of $(\Delta\%)_t$. The final step is to apply Equation (2.1) to the replicate estimates of $(\Delta\%)_t$.

Figure 4.7.1 shows how this can be done with SAS.

Figure 4.7.1: SAS Code for Estimating Variance of a Rate of Change

```
* Subset to renter-occupied units with valid rent. ;
Data D2014;
  set HU14;
  where sc116 in ('2','3') and uf26 not in ('99999') ;
  * Convert rent to numeric variable. ;
  rent14=input(uf26,8.);
run;

Data D2017 ;
  set HU17 ;
  where sc116 in ('2','3') and uf26 not in ('99999') ;
  rent17=input(uf26,8.) ;
run;

* This empty data set is produced for the merge later. ;
data data23 ;
  length diff0-diff80 8. ;
run ;

* Steps 1 & 2: Estimate the replicate estimates for replicates 0-80. ;
%macro repssss(rep) ;
* Estimate theta for each year. ;
proc means data= d2014 median noprint ;
  weight fw&rep. ;
  var rent14 ;
  output out=datam14rep&rep. median=median14rep&rep. ;
run ;

proc means data= d2017 median noprint ;
  weight fw&rep. ;
  var rent17 ;
  output out=datam17rep&rep. median=median17rep&rep. ;
run ;

* Merge the replicate estimates of each year by replicate. ;
data datamrep&rep. ;
  merge datam17rep&rep. datam14rep&rep. ;
  diff&rep. = (median17rep&rep.-median14rep&rep. )/median14rep&rep.;
  keep diff&rep. ;
run ;

data data23 ;
  merge data23 datamrep&rep.;
run ;

%mend repssss ;

%macro doit ;
  %do i=0 %to 80 ;
    %repssss(&i.) ;
    %end ;
%mend doit ;
%doit ;
```

Figure 4.7.1: SAS Code for Estimating Variance of a Rate of Change

```
* Apply Step 3 to the replicate estimates and estimate the variance. ;
data data24 (keep=diff0 var se) ;
  set data23 end=eof ;
  * Fill array with the replicate means. ;
  array diff{80} diff1-diff80 ;
  * Fill array with the squared diffs. ;
  array sdiffsq{80} sdiffsq1-sdiffsq80 ;
  do j = 1 to 80 ;
    sdiffsq{j} = (diff{j} - diff0)**2 ;
    end ;
  * Sum the squared diffs. ;
  totdiff = sum(of sdiffsq1-sdiffsq80) ;
  var = (4/80) * totdiff ;
  se = (var)**(0.5) ;
  output ;
run ;

proc print data=data24 noobs ;
  var diff0 se ;
  format diff0 se 8.4 ;
run ;
```

The SAS code of Figure 4.7.1 produces the output in Figure 4.7.2.

Figure 4.7.2: SAS Output for Estimating Variance of a Rate of Change

diff0	se
0.0943	0.0074

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

So the median gross rent in NYC for 2017 has increased 9.4 percent from 2014, with a standard error of 0. 7 percent.

5. Examples of Calculating Variances with Stata

In addition to SAS, data users can also use Stata to calculate variances for NYCHVS. Similar to how Section 4 is structured, we review the same examples using the Stata software.

Overall, Stata produces the exact same estimates and, for the most part, close approximations to standard errors as the SAS and by-hand methods.

Example 5.1. Estimating the Variance of a Total

When reading NYCHVS PUFs into Stata, first download and save these data files as txt files. Then download the STATA import program (text file #2 mentioned in page 7) provided on the Census Bureau's NYCHVS website at: https://www.census.gov/programs-surveys/nychvs/data/datasets.html, update the input-related items in the code (highlighted in yellow in Figure 5.1.1). Run the part of the code for Occupied Records, and/or Person Records, and/or Vacant Records, then local Stata datasets will be created. Figure 5.1.1 shows the top part of the code for reading in Occupied HUs data file (text file #3). For generating Stata datasets for Person Records or Vacant Records (text file #4 and #5 respectively), just follow the same steps for Occupied Records.

Figure 5.1.1: Stata Code (Partial) for Reading in the Public Use Files

```
*2017 NYCHVS Stata IMPORT PRORGRAM

cd "LOCATION OF TEXT FILES"

*OCCUPIED DATA FILE
clear
infix          ///
recid  1       ///

***SEE THE REST OF THE STATA CODE IN STATA IMPORT PROGRAM FOUND IN THE CENSUS BUREAU'S NYCHVS WEBSITE MENTIONED ABOVE***
```

For housing units estimates, users need to download both Occupied and Vacant Records (text files #3 and #5), and then combine both files. In our examples, we saved the occupied record file as nychvs_17_occ, and saved the vacant record file as nychvs_17_vac, and their combined file as HU17, which will be used in the Stata examples. Figure 5.1.2 shows the Stata code for appending the Occupied Records after saving the Vacant Records.

Figure 5.1.2: Stata Code for Append Two Files Together for Housing Unit Estimates

```
* append Occupied Records with Vacant Records

Append using "nychvs_17_occ.dta"
```

```
Save "HU17.dta", replace
```

Since the final weights and replicate weights have five implied decimal places, users need to convert them. Figure 5.1.3 shows the Stata code on how to convert the weights to numeric.

Figure 5.1.3: Stata Code for Converting Weights with Implied Decimal Places

```
* Dividing weights by 100,000

Replace fw=fw/10000

foreach x of var fw1-fw80 {
      replace `x' = `x' / 100000
}
```

For generating Stata datasets for 2014 or 2011 NYCHVS PUFs, go to the same website mentioned in the previous page, click the "2014" or "2011" tab, and repeat the same steps in Figure 5.1.1 through Figure 5.1.3 as for 2017. The 2014 dataset is saved as HU14, and will be used in later examples.

After local Stata datasets are created, run the code in Figure 5.1.4 before running any example analysis at the household level.

Figure 5.1.4: Reading in the Data File into Stata and Setting Survey Design Parameters

```
cd "DIRECTORY PATH OF NYCHVS DATA"

* First, household level data:
use "HU17.dta", clear

* Setting survey design parameters:
svyset[pweight=fw], vce(sdr) sdrweight(fw1-fw80) fay(.5)mse
```

In this example, we estimate the total occupied housing units in NYC and use the replicate weights to calculate the variance. This is done by first identifying the domain of interest and then running the total and variance calculation by using the Stata code in Figure 5.1.5. The resulting output is provided in Figure 5.1.6.

Note that both `vce(brr)` and `vce(sdr)` options generate the same estimate and standard errors. The only difference between those two options is the first one uses t-distribution and the second uses a z-distribution, thus, their CIs are slightly different. NYCHVS uses SDR, so users should use `vce(sdr)` option. For a more detailed explanation of z-distribution and t-distribution, please refer to Section 7 on page 54.

In Figure 5.1.5, we used the `subpop` instead of the `if` option. Either would produce the same variance estimates – see also "Cautions about Domain Analysis Don't Apply" in Section 2.

Figure 5.1.5: Stata Code for Estimating Variance of a Total

```
* Generate occupied housing unit dummy var:
gen occ_final = (recid == 1)

* Now estimating total:
svy, subpop(occ_final): total occ_final
```

The Stata code of Figure 5.1.5 generates the output of Figure 5.1.6.

Figure 5.1.6: Stata Output for Estimating Variance of a Total

```
Survey: Total estimation          Number of obs   =     15,135
                                  Population size =  3,469,240
                                  Subpop. no. obs =     13,266
                                  Subpop. size    =  3,109,955
                                  Replications    =         80

--------------------------------------------------------------
             |                SDR *
             |      Total   Std. Err.     [95% Conf. Interval]
-------------+------------------------------------------------
   occ_final |    3109955   8626.835       3093046     3126863
--------------------------------------------------------------
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

From Figure 5.1.6, the estimate of total occupied housing units in NYC is 3,109,955 with an estimated standard error of 8,627. The estimates from Stata are the same as the estimates produced by SAS in Example 4.1. However, the standard error is slightly different before rounding. Unlike SAS, the `[svy] sdr` command in Stata displays the 95 percent confidence intervals by default – for more information on the confidence intervals, see Section 7.

Example 5.2. Estimating the Variance of a Difference

In this example, we are interested in calculating the difference between the number of rent stabilized units in Manhattan and the number of rent stabilized units in the Bronx for 2017.

First, we define the domains of interest individually and then define the difference. After we do this, we are able to calculate the variance using the replicate weights in the Stata code defined in Figure 5.2.1.

Figure 5.2.1: Stata Code for Estimating Variance of a Difference

```
* Bronx:
gen rent_bronx = ((uf_csr == 30 | uf_csr == 31) & (boro == 1))

* Manhattan:
gen rent_man = ((uf_csr == 30 | uf_csr == 31) & boro == 3)

* Creating difference between these:
gen stable_diff = rent_man - rent_bronx

* Create proper subdomain:
gen rent_boro = (((uf_csr == 30 | uf_csr == 31) & boro == 3) | ((uf_csr
== 30 | uf_csr == 31) & (boro == 1)))

* Now estimating differences between these:
svy, subpop(rent_boro): total stable_diff
```

The Stata code of Figure 5.2.1 generates the output of Figure 5.2.2.

Figure 5.2.2: Stata Output for Estimating Variance of a Difference

```
Survey: Total estimation          Number of obs   =     15,135
                                  Population size =  3,469,240
                                  Subpop. no. obs =      2,010
                                  Subpop. size    = 482,501.87
                                  Replications    =         80

--------------------------------------------------------------
             |                SDR *
             |      Total   Std. Err.     [95% Conf. Interval]
-------------+------------------------------------------------
 stable_diff |   15497.43   10419.88     -4925.149    35920.02
--------------------------------------------------------------
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

From Figure 5.2.2, we can see the estimated difference of rent stabilized units in Manhattan and the Bronx is 15,497 and the standard error is 10,420. These results are the same as produced with SAS – see Example 4.2.

Example 5.3. Estimating the Variance of a Mean

For this example, we are estimating the average gross rent for renter-occupied housing units in NYC for 2017 and its variance. This can be easily done in Stata by first identifying the domains of interest and using the Stata code to calculate the variance using the replicate weights. The Stata code is provided in Figure 5.3.1.

Figure 5.3.1: Stata Code for Estimating Variance of a Mean

```
* Need proper subdomain var: Renters only - note STATA automatically
removes missing values from analysis

gen renters = (sc116 == 2 | sc116 == 3)

gen gross_rent = uf26
replace gross_rent = . if uf26 == 99999

* Now estimating mean:
svy, subpop(renters): mean gross_rent
```

The Stata code of Figure 5.3.1 generates the output of Figure 5.3.2.

Figure 5.3.2: Stata Output for Estimating Variance of a Mean

```
Survey: Mean estimation           Number of obs   =     14,858
                                  Population size =  3,404,017
                                  Subpop. no. obs =      8,902
                                  Subpop. size    =  2,038,651
                                  Replications    =         80

--------------------------------------------------------------
             |                SDR *
             |       Mean   Std. Err.     [95% Conf. Interval]
-------------+------------------------------------------------
  gross_rent |   1693.539   10.58051      1672.801    1714.276
--------------------------------------------------------------
```

Source: U.S. Census Bureau, 2017, New York City Housing and Vacancy Survey.

From Figure 5.3.2, we can see that the average NYC gross rent calculated using Stata is $1,694 and the corresponding standard error is $11, and again, the same as produced with SAS – see Examples 3.1 and 4.3.

Example 5.4. Estimating the Variance of a Median

In working with Stata on calculating the variance of a median, users first need to download a program, `epctile` written by Kolenikov (2019). Depending on one's Stata settings and internet permissions, `epctile` can be installed using the code provided in Figure 5.4.1.

Figure 5.4.1: Stata Code for Installing epctile Program

```
net describe epctile, from(http://staskolenikov.net/stata)
net install epctile
```

Using `epctile`, we are able to calculate the median gross rent and its standard error for renter-occupied housing units in NYC for 2017. Figure 5.4.2 shows the Stata code for calculating variance of a median, the domain of interest is the same as the previous example and the code for the domain identification can be found in Example 5.3.

Figure 5.4.2: Stata Code for Estimating Variance of a Median

```
* Median Value:
Svy, subpop(renters): epctile gross_rent, p(50)
return list
```

The Stata code of Figure 5.4.2 generates the output of Figure 5.4.3.

Figure 5.4.3: Stata Output for Estimating Variance of a Median

```
SDR results                                     Number of obs      =     15,135
                                                Population size    =  3,469,240
                                                Subpop. no. of obs =      9,179
                                                Subpop. size       =  2,103,873
                                                Replications       =         80

------------------------------------------------------------------------------
             |                SDR *
  gross_rent |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         p50 |        1450   7.483315   193.76   0.000     1435.34    1464.66
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The estimate for median gross rent for renter-occupied housing units in NYC is $1,450, with standard error of $7. The results are the same as produced by SAS – see Example 4.4.

Example 5.5. Estimating the Variance of a Regression Parameter

For this example, we are interested in modeling the 2017 gross rent of renter-occupied housing units using the year the householder moved into the housing unit.

In Stata, we first need to convert strings to numeric values and then provide the Stata code to create a linear regression model using the replicate weights. The code is provided in Figure 5.5.1.

Figure 5.5.1: Stata Code for Estimating Variance of a Regression Parameter

```
* Linear regression model:
svy, subpop(renters): reg gross_rent uf66
```

The Stata code of Figure 5.5.1 generates the output of Figure 5.5.2.

Figure 5.5.2: Stata Output for Estimating Variance of a Regression Parameter

```
Survey: Linear regression                       Number of obs     =     14,858
                                                Population size   =  3,404,017
                                                Subpop. no. obs   =      8,902
                                                Subpop. size      =  2,038,651
                                                Replications      =         80
                                                Wald chi2(1)      =    1256.30
                                                Prob > chi2       =     0.0000
                                                R-squared         =     0.1026

------------------------------------------------------------------------------
             |                SDR *
  gross_rent |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        uf66 |   28.78853   .8122175    35.44   0.000     27.19662    30.38045
       _cons |  -56053.71   1625.767   -34.48   0.000    -59240.16   -52867.27
------------------------------------------------------------------------------
```

Source: U.S. Census Bureau, 2017, New York City Housing and Vacancy Survey.

From Figure 5.5.2, we can see that the calculated regression parameter is 28.79 and the standard error is 0.81. The estimates are the same produced by SAS – see Examples 3.3 and 4.5.

Example 5.6. Estimating the Variance of a Longitudinal Change

Both this example and the next consider statistics that measure longitudinal change. This example will consider a statistic that measures the change of a rate between two cycles of the estimates. Example 5.7 will consider an estimate of percent of change that is calculated at the HU level. We present these two longitudinal statistics as examples thinking these calculations are the most likely types of calculations that data users would be interested in.

In this example, we show how to estimate the variance of a difference in proportion. Specifically, we are interested in the estimated the vacancy rate or $\hat{p}_t = \hat{X}_t / \hat{N}_t * 100$. Where $\hat{X}_t$ is the estimator of the total number of vacant HUs at time *t*, and $\hat{N}_t$ is the estimator of the total number of HUs at time *t.*

Further, we are really interested in the difference in the vacancy rate between 2017 and 2014. So the statistic of interest is:

$$\hat{\Delta}_{t=2017} = \frac{\hat{X}_{t=2017}}{\hat{N}_{t=2017}} * 100 - \frac{\hat{X}_{t=2014}}{\hat{N}_{t=2014}} * 100$$

To estimate the variance of $\hat{\Delta}_t$ with replicate weights, we first calculate all the pieces of the $\hat{\Delta}_{t=2017}$ for each of the 80 replicates, making 80 replicate estimates of $\hat{\Delta}_{t=2017}$, repeat those

steps for 2014, and then we apply Equation (2.1). Figure 5.6.1 shows how we do this with Stata.

Figure 5.6.1: Stata Code for Estimating Variance of a Longitudinal Change

```
* Read in 2017 data:
use "HU17.dta", clear

* Generate vacancy variable:
gen vac = (recid == 3 )

* Calculating 2017 totals:
quietly summarize fw if vac == 1
scalar vac_est17 = r(sum)

quietly summarize fw
scalar tot_est17 = r(sum)

scalar est17 = (vac_est17 / tot_est17) * 100

* Creating matrix that stores 2017 estimates:
matrix all_ests_17 = (est17)

* Now going through replicates:
foreach x of var fw1-fw80 {

      quietly summarize `x' if vac == 1
      scalar vac_est17_i = r(sum)

      quietly summarize `x'
      scalar tot_est17_i = r(sum)

      * i_th replicate estimate for 2017 portion:
      scalar est17_i = (vac_est17_i / tot_est17_i) * 100

      * Adding to matrix that stores 2017 estimates:
      matrix all_ests_17 = (all_ests_17 \ est17_i)
}

* Now reading in 2014 data:
use "HU14.dta", clear

* Generate vacancy variable:
gen vac = (recid == 3)

* Calculating 2014 totals:
quietly summarize fw if vac == 1
scalar vac_est14 = r(sum)

quietly summarize fw
scalar tot_est14 = r(sum)

scalar est14 = (vac_est14 / tot_est14) * 100

* Creating matrix that stores 2014 estimates:
matrix all_ests_14 = (est14)
```

Figure 5.6.1: Stata Code for Estimating Variance of a Longitudinal Change

```
* Now going through replicates:
foreach x of var fw1-fw80 {

      quietly summarize `x' if vac == 1
      scalar vac_est14_i = r(sum)

      quietly summarize `x'
      scalar tot_est14_i = r(sum)

      * i_th replicate estimate for 2014 portion:
      scalar est14_i = (vac_est14_i / tot_est14_i) * 100

      * Adding to matrix that stores 2014 estimates:
      matrix all_ests_14 = (all_ests_14 \ est14_i)
}

* Matrix will point estimates for the sample estimate and all replicates:
matrix diffs = all_ests_17 - all_ests_14

* Now getting squared differences for variance estimation:
scalar var = 0
quietly forvalues i = 2/81 {

      scalar sq_pt_est_diffs = (diffs[`i',1] - diffs[1,1])^2
      scalar var = var + sq_pt_est_diffs
}

scalar var = (4/80) * var
scalar se_est = (var)^(0.5)
```

The Stata code of Figure 5.6.1 generates the output of Figure 5.6.2.

Figure 5.6.2: Stata Output for Estimating Variance of a Longitudinal Change

```
. * Final Point Estimate:
. display diffs[1,1]
2.2402014

. * Final Estimated Standard Error:
. display se_est
.25878038
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

So, the change in the vacancy rate from 2014 to 2017 is 2.2 percent with a standard error of 0.3 percent.

Example 5.7. Estimating the Variance of a Percent Change in Two Cycles of NYCHVS

There are many different statistics that measure the change between two cycles of NYCHVS. Differences, percent change, and ratios can be used to measure how much a given statistic changed from one cycle of NYCHVS to another. This example considers the housing characteristic of the median gross rent and how it can change over time. Both the rate of change and resulting variance are demonstrated.

The statistic of interest is the percent change in the median gross rent from 2014 to 2017. Let $\widehat{M}_t$ be the estimator of the median gross rent at time *t*. The statistic of interest is

$$\%\hat{\Delta}_t = \frac{\widehat{M}_t - \widehat{M}_{t-1}}{\widehat{M}_{t-1}}$$

To estimate the variance of $(\Delta\%)_t$, we use the 2017 replicate weights and store 80 replicate estimates of $\widehat{\mathrm{M}}_{\mathrm{t=2017}}$ within a matrix, and similarly use the 2014 replicate weights to store 80 replicate estimates of $\widehat{\mathrm{M}}_{\mathrm{t=2014}}$ in another matrix. Next, we take the difference of these two matrices $\widehat{\mathrm{M}}_{\mathrm{t=2017}}$ and $\widehat{\mathrm{M}}_{\mathrm{t=2014}}$ by replicate and then divide each matrix entry by $\widehat{\mathrm{M}}_{\mathrm{t=2014}}$ to get our 80 replicate estimates of $(\Delta\%)_t$. The final step is to apply Equation (2.1) to the replicate estimates of $(\Delta\%)_t$.

Figure 5.7.1 shows how this can be done with Stata.

Figure 5.7.1: Stata Code for Estimating Variance of a Rate of Change

```
* Read in 2017 data:
use " HU17.dta", clear

* Generating needed variables - renters and gross rent:
gen renters = (sc116 == 2 | sc116 == 3) & uf26 != 99999
gen gross_rent = uf26

* Calculating 2017 median gross rent:
quietly summarize gross_rent [w=fw] if renters == 1, detail
scalar rent17 = r(p50)

* Matrix to store 2017 estimates:
matrix rent_ests_17 = (rent17)

* Now looping through all repweights for 2017:
foreach x of var fw1-fw80 {

      quietly summarize gross_rent [w=`x'] if renters == 1, detail
      scalar rent17_ith = r(p50)

      * Adding to matrix that stores 2017 estimates:
      matrix rent_ests_17 = (rent_ests_17 \ rent17_ith)
}
* Now to 2014 data:
use "HU14.dta", clear
```

Figure 5.7.1: Stata Code for Estimating Variance of a Rate of Change

```
* Generating needed variables - renters and gross rent:
gen renters = (sc116 == 2 | sc116 == 3) & uf26 != 99999
gen gross_rent = uf26

* Calculating 2014 median gross rent:
quietly summarize gross_rent [w=fw] if renters == 1, detail
scalar rent14 = r(p50)

* Creating matrix that stores 2014 estimates:
matrix rent_ests_14 = (rent14)

* Now looping through all repweights for 2014:
foreach x of var fw1-fw80 {

      quietly summarize gross_rent [w=`x'] if renters == 1, detail
      scalar rent14_ith = r(p50)

      * Adding to matrix that stores 2014 estimates:
      matrix rent_ests_14 = (rent_ests_14 \ rent14_ith)
}

* Final Matrix - for now storing the numerator of the estimate for
sample and all replicates:
matrix med_rent_increase = (rent_ests_17 - rent_ests_14)

forvalues i = 1/81 {

      * Converting entries in final matrix to hold final estiamtes for
full sample and all replicates:
      matrix med_rent_increase[`i',1] = med_rent_increase [`i',1] /
rent_ests_14[`i',1]

}

* Now getting squared differences for variance estimation:
scalar var = 0
quietly forvalues i = 2/81 {

      scalar sq_pt_est_med_increase = (med_rent_increase[`i',1] -
med_rent_increase[1,1])^2
      scalar var = var + sq_pt_est_med_increase
}

scalar var = (4/80) * var
scalar se_est_med_rent = (var)^(0.5)

* Final Point Estimate:
display med_rent_increase[1,1]

* Final Estimated Standard Error:
display se_est_med_rent
```

The Stata code of Figure 5.7.1 generates the output of Figure 5.7.2.

Figure 5.7.2: Stata Output for Estimating Variance of a Rate of Change

```
. * Final Point Estimate:
. display med_rent_increase[1,1]
.09433962

. * Final Estimated Standard Error:
. display se_est_med_rent
.00739298
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Therefore, the median gross rent for NYC in 2017 has increased 9.4 percent from 2014, with a standard error of 0.7 percent.

6. Examples of Calculating Variances with R

R is another statistical package that can calculate variances using replicate weights for the NYCHVS. This section is structured similar to Sections 3, 4 and 5 and uses the same examples for calculating variances of a total, difference, mean, median, regression parameter, longitudinal change, and percent change.

Overall, users can use R to produce the exact same estimates as produced by SAS and Stata; however, the standard errors generated by R are generally slightly different. We know that differences of the standard errors in Examples 6.1-6.3 are due to R using the mean of the replicates as $\hat{\theta}_0$ in Equation (2.1) instead of using the estimate derived from the final weight. See also "Use of Replicate Estimates in Variance Calculations" in Section 2 for more about defining $\hat{\theta}_0$ in Equation (2.1).

Example 6.1. Estimating the Variance of a Total

Although the NYCHVS PUFs are ACSII text files, there are no column/variable names on these files, and data users still need to use the SAS Import Program (text file #1 mentioned in Section 2, Page 7 provided in the Census Bureau's NYCHVS website at https://www.census.gov/data/datasets/2017/demo/nychvs/microdata.html to read in the NYCHVS PUFs into SAS (please see Pages 7 and 8 for instructions on how to read PUFs into SAS). Once in SAS, data users need to run the code in Figure 2.1.1, Figure 2.1.2, Figure 2.1.3, and Figure 2.1.4, which generate the datasets HU17 and HU14. Then, users can export the data sets from SAS to CSV format in order to read the file into R. In SAS, run the code in Figure 6.1.1 to convert the SAS datasets HU17 and HU14 to an R / CSV dataset.

Figure 6.1.1: SAS Code to Output Files into R Readable Format

```
* Exporting NYCHVS datasets to R;

proc export data = HU17 dbms = csv outfile="PATH \ TO \ NEW R
DATASET\new_dataset_name.csv" replace;
run;

proc export data = HU14 dbms = csv outfile="PATH \ TO \ NEW R
DATASET\new_dataset_name2.csv" replace;
run;
```

Once data users are in R, some initial code needs to run in order to set things up for the rest of Section 6. This includes ensuring the proper R Packages are installed. Our examples use the "survey" and "spatstat" packages. The initial steps in Figure 6.1.2. show how to install the survey packages in R.

Figure 6.1.2: R Code to Install Survey Package and Read in Data File

```
# Install survey package with install.packages("survey") and
install.packages("spatstat") if not already # installed.
library(survey)
library(spatstat)

# Changing to relevant directory:
setwd("PATH TO NYCHVS DATA")

# Reading in HU dataset:
hu <- read.csv("NAME OF HU-LEVEL NYCHVS IN CSV FORMAT.csv", header=TRUE)
```

Also, prior to running any of the examples in Section 6, ensure that the following code is used to set up the domains found in the examples. See Figure 6.1.3 for the code to use for domain definitions.

Figure 6.1.3: R Code for Domain Definitions

```
## Total Example:
# Occupied housing units:
hu$occ_final = ifelse(hu$recid == 1, 1, 0)

## Difference Example:
# Renter occupied the Bronx:
hu$rent_bronx = ifelse(((hu$uf_csr == 30 | hu$uf_csr == 31) & (hu$boro
== 1)), 1, 0)

# Renter Occupied Manhattan:
hu$rent_man = ifelse((hu$uf_csr == 30 | hu$uf_csr == 31) & (hu$boro ==
3), 1, 0)

# Difference between Renters in Manhattan and the Bronx:
hu$stable_diff = hu$rent_man - hu$rent_bronx

# Proper subdomain for differences:
hu$rent_boro = ifelse((
      ((hu$uf_csr == 30 | hu$uf_csr == 31) & (hu$boro == 3)) |
      ((hu$uf_csr == 30 | hu$uf_csr == 31) & (hu$boro == 1))), 1, 0)

## Mean / Median Example:
# Subdomain: Renters only with non-missing rent values
hu$renters = ifelse(((hu$sc116 == 2 | hu$sc116 == 3) & hu$uf26 !=
99999), 1, 0)

# Gross Rent:
hu$gross_rent <- hu$uf26
hu$gross_rent[hu$uf26 == 99999] <- NA

## Odds Ratio Example:
# Binary variable for three or more deficiencies:
hu$defect = ifelse((hu$rec53 == 4 | hu$rec53 == 5 | hu$rec53 == 6 |
hu$rec53 == 7 | hu$rec53 == 8), 1, 0)

# Need binaries for time demarcation:
```

Figure 6.1.3: R Code for Domain Definitions

```
hu$time_stable <- ifelse(hu$uf_csr == 30, 0, NA)
hu$time_stable[hu$uf_csr == 31] <- 1

## First Longitudinal Example:
# Adding vacancy dummy:
hu$vac <- ifelse((hu$recid == 3 ), 1, 0)

# Setting survey design parameters:
## Note: make sure column numbers align with your datasets variables /
repweights / final weight locations:
hu_design <- svrepdesign(variables=hu[,c(1:946, 1028:1257)],
      repweights=hu[,947:1026],
      weights=hu[,1027], combined.weights=TRUE,
      type = "other", scale = 4/80, rscales = 1)
```

For our example of estimating the variance of a total, we will estimate the total occupied housing units in NYC and use the replicate weights to calculate the variance. Use the R code in Figure 6.1.4 in order to do this calculation.

Figure 6.1.4: R Code for Estimating Variance of a Total

```
# Now estimating total:
occ_hus <- subset(hu_design, occ_final == 1)
svytotal(~occ_final, design = occ_hus)
```

The R code provided in Figure 6.1.4 produces the following output in R shown in Figure 6.1.5.

Figure 6.1.5: R Output for Estimating Variance of a Total

	total	SE
occ_final	3,109,955	8,611.5

Source: U.S. Census Bureau, 2017, New York City Housing and Vacancy Survey.

From the output in Figure 6.1.5, we see that the estimated total occupied housing units in NYC is 3,109,955 and its standard error is 8,612.

Example 6.2. Estimating the Variance of a Difference

In this example, we are interested in calculating the difference between the number of rent stabilized units in Manhattan and the number of rent stabilized units in the Bronx for 2017.

The R code provided in Figure 6.2.1 can be used to calculate the estimated difference and its variance calculation.

Figure 6.2.1: R Code for Estimating Variance of a Difference

```
rent_boros <- subset(hu_design, rent_boro == 1)
svytotal(~stable_diff, design = rent_boros)
```

The output from the code in Figure 6.2.1 is provided in Figure 6.2.1.

Figure 6.2.2: R Output for Estimating Variance of a Difference

```
               total        SE
stable_diff   15,497    10,253
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

From the R output, we can see that the estimated difference between rent stabilized units in Manhattan versus the rent stabilized units in the Bronx is 15,497 and the standard error is 10,253.

Example 6.3. Estimating the Variance of a Mean

This example calculates the variance of a mean for gross rent. We are interested in the average gross rent for renter-occupied housing units in NYC for 2017.

The R code to make the point estimate and variance estimate for this is provided in Figure 6.3.1.

Figure 6.3.1: R Code for Estimating Variance of a Mean

```
svymean(~gross_rent, design = renter_hus)
```

The R output from using the code in 6.3.1 is provided in Figure 6.3.2.

Figure 6.3.2: R Output for Estimating Variance of a Mean

```
              mean        SE
gross_rent   1,693.5    10.562
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The estimated average gross rent for renter-occupied housing units in NYC for 2017 is $1,694 with a standard error of $11.

Example 6.4. Estimating the Variance of a Median

We will use R to compute the median gross rent for renter-occupied housing units in NYC for 2017.

Figure 6.4.1: R Code for Estimating Variance of a Median

```
renter_hus <- subset(hu_design, renters == 1)
svyquantile(~gross_rent, design = renter_hus, quantiles=c(.5))
```

The R output from using the code in Figure 6.4.1 is provided in Figure 6.4.2.

Figure 6.4.2: R Output for Estimating Variance of a Median

```
Statistic:
gross_rent
q0.5       1,450
SE:
gross_rent
q0.5   8.794271
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The estimated median gross rent for NYC in 2017 is $1,450, with a standard error of $9.

Example 6.5. Estimating the Variance of Regression Parameter

For this example, we are interested in modeling the 2017 gross rent of renter-occupied housing units using the year the householder moves into the housing unit.

In R, we use the `svyglm` command to create a linear regression model using the replicate weights. The R code is provided in Figure 6.5.1.

Figure 6.5.1: R Code for Estimating Variance of a Regression Parameter

```
lin_ex <- svyglm(gross_rent~uf66, design = renter_hus)
summary(lin_ex)
```

Figure 6.5.1 generates the output in Figure 6.5.2.

Figure 6.5.2: R Output for Estimating Variance of a Regression Parameter

```
Call:
svyglm(formula = gross_rent ~ uf66, design = renter_hus)

Survey design:
subset(hu_design, renters == 1)

Coefficients:
             Estimate      Std. Error     t value     Pr(>|t|)
(Intercept) -5.605e+04      1.619e+03    -34.63     <2e-16 ***
UF66         2.879e+01      8.086e-01     35.60     <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 1066817)

Number of Fisher Scoring iterations: 2
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

From the R output provided in Figure 6.5.2, we can see that the calculated regression parameter is 28.79 with the standard error of 0.81.

Example 6.6. Estimating the Variance of a Longitudinal Change

Both this example and the next consider statistics that measure longitudinal change. This example will consider a statistic that measures the change of a rate between two cycles of the estimates. Example 6.7 will consider an estimate of percent of change that is calculated at the HU level. We present these two longitudinal statistics as examples thinking these calculations are the most likely of calculations that data users would be interested in.

In this example, we show how to estimate the variance of a difference in proportion. Specifically, we interested in estimate the vacancy rate or $\hat{p}_t = \hat{X}_t / \hat{N}_t * 100$. Where $\hat{X}_t$ is the estimator of the total number of vacant HUs at time *t*, and $\hat{N}_t$ is the estimator of the total number of HUs at time *t.*

Further, we're really interested in the difference in the vacancy rate between 2017 and 2014. So the statistic of interest is:

$$\hat{\Delta}_{t=2017} = \frac{\hat{X}_{t=2017}}{\hat{N}_{t=2017}} * 100 - \frac{\hat{X}_{t=2014}}{\hat{N}_{t=2014}} * 100$$

To estimate the variance of $\hat{\Delta}_t$ with replicate weights, we first calculate all the pieces of the $\hat{\Delta}_{t=2017}$ for each of the 80 replicates, make 80 replicate estimates of $\hat{\Delta}_{t=2017}$, repeat those

steps for 2014, and then apply Equation (2.1). Figure 6.6.1 shows how we do this with R.

Figure 6.6.1: R Command for Estimating Variance of a Longitudinal Change

```
# Calculating 2017 point estimates:
hu_vacs <- subset(hu, vac == 1)
vac_est17 <- sum(hu_vacs$FW)
total_est17 <- sum(hu$FW)
est17 <- (vac_est17 / total_est17) * 100


# Reading in 2014 data:
hu14 <- read.csv("full2014.csv", header=TRUE)

# Adding vacancy dummy:
hu14$vac <- ifelse(hu14$recid == 3, 1, 0)

# Calculating 2014 point estimates:
hu_vac14 <- subset(hu14, vac == 1)
vac_est14 <- sum(hu_vac14$fw)
total_est14 <- sum(hu14$fw)
est14 <- (vac_est14 / total_est14) * 100

# Final Point Estimate:
est <- est17 - est14

# Vector storing replicates:
reps <- c()

# Looping calculations for variance estimate:
## Again, make sure column numbers align with your dataset's repweights:
for (i in c(1:80)) {
      vac_rep17 <- sum(hu_vacs[,(i+946)])
      tot_rep17 <- sum(hu[,(i+946)])
      vac_rep14 <- sum(hu_vac14[,(i+3)])
      tot_rep14 <- sum(hu14[,(i+3)])

      temp_est <- ((vac_rep17 / tot_rep17) - (vac_rep14 / tot_rep14)) *
100

      # Adding Result to Vector of Squarred Difference Results:
      reps <- rbind(reps, (temp_est - est)^2)
}

est_var <- (4/80) * sum(reps)
est_se <- sqrt(est_var)
```

The R command of Figure 6.6.1 produces the output in Figure 6.6.2.

Figure 6.6.2: R Output for Estimating Variance of a Longitudinal Change

```
> # Final Point Estimate:
> est
[1] 2.240201

> # Final Estimate of Standard Error:
> est_se
[1] 0.2587804
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

So the change in the vacancy rate from 2014 to 2017 is 2.2 percent with a standard error of 0.3 percent.

Example 6.7. Estimating the Variance of a Percent Change in Two Cycles of NYCHVS

There are many different statistics that measure the change between two cycles of NYCHVS. Differences, percent change, and ratios can be used to measure how much a given statistic changed from one cycle of NYCHVS to another. This example considers the housing characteristic of the median gross rent and how it can change over time. Both the rate of change and resulting variance are demonstrated.

The statistic of interest is the percent change in the median gross rent from 2014 to 2017. Let $\widehat{M}_t$ the estimator of the median gross rent at time *t*. The statistic of interest is

$$\%\hat{\Delta}_t = \frac{\widehat{M}_t - \widehat{M}_{t-1}}{\widehat{M}_{t-1}}$$

To estimate the variance of $(\Delta\%)_t$, we first calculate the point estimates of $(\Delta\%)_t$. Then, we use both the 2017 and 2014 replicate weights to store 80 replicate estimates of $(\Delta\%)_t$ within a vector, and store the squared differences between these replicate estimates and the final point estimate in another vector. We sum these up, multiply by (4/80), and take the square root of that result, thus applying Equation (2.1) to arrive at our final estimate of standard error for $(\Delta\%)_t$.

Figure 6.7.1 shows how this can be done with R.

Figure 6.7.1: R Command for Estimating Variance of a Rate of Change

```
# Calculating 2017 point estimates:
hu_rent <- subset(hu, renters == 1)
rent17 <- weighted.median(hu_rent$gross_rent, hu_rent$fw)


# Adding renters and gross rent vars to 2014:
hu14$renters = ifelse(((hu14$sc116 == 2 | hu14$sc116 == 3) & hu14$uf26 !=
99999), 1, 0)
hu14$gross_rent <- hu14$uf26
```

Figure 6.7.1: R Command for Estimating Variance of a Rate of Change

```
hu14$gross_rent[hu14$uf26 == 99999] <- NA

# Calculating 2014 point estimates:
hu_rent14 <- subset(hu14, renters == 1)
rent14 <- weighted.median(hu_rent14$gross_rent, hu_rent14$FW)


# Final Point Estimate:
med_rent_increase <- (rent17 - rent14) / rent14

# Vector storing replicates:
reps <- c()

# Looping calculations for variance estimate:
## Again, make sure column numbers align with your dataset's repweights:
for (i in c(1:80)) {
      rent17_est <- weighted.median(hu_rent$gross_rent, hu_rent[,(i+946)])
      rent14_est <- weighted.median(hu_rent14$gross_rent, hu_rent14[,(i+3)])

      temp_est <- (rent17_est - rent14_est) / rent14_est

      # Adding Result to Vector of Squarred Difference Results:
      reps <- rbind(reps, (temp_est - med_rent_increase)^2)
}

est_var <- (4/80) * sum(reps)
est_se_med_rent <- sqrt(est_var)
```

The R code of Figure 6.7.1 produces the output in Figure 6.7.2.

Figure 6.7.2: R Output for Estimating Variance of a Rate of Change

```
# Final Point Estimate:
> med_rent_increase
[1] 0.09433962

# Final Estimate of Standard Error:
> est_se_med_rent
[1] 0.007371978
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

So, the median gross rent for NYC in 2017 has increased 9.4 percent from 2014, with a standard error of 0.7 percent.

7. How to Calculate Confidence Intervals

Lohr (1999; Section 9.5) has an excellent review of confidence intervals (CIs) for survey estimates, from which we borrow heavily for the introduction of this section.

Survey estimates that employ sample weights are different than unweighted estimates. However, under certain conditions it can be shown that $(\hat{\theta} - \theta)/\sqrt{\hat{v}(\hat{\theta})}$ asymptotically standard normal for survey estimates (Krewski and Rao 1981). Consequently, when the assumptions are met, an approximate 95 percent CI for θ may be constructed as:

$$\hat{\theta} \pm Z_{\alpha=0.05}\sqrt{\hat{v}(\hat{\theta})} = \hat{\theta} \pm 1.96\sqrt{\hat{v}(\hat{\theta})} \tag{7.1}$$

where $\sqrt{\hat{v}(\hat{\theta})}$ is the standard error of the estimator of $\hat{\theta}$.

Normal Distribution versus the *t*-distribution for Confidence Intervals

In this document, we suggest using a normal distribution with all confidence intervals. We suggest this for two reasons. First, there are no good guidelines for the number of degrees of freedom that should be used with a t-distribution and variances generated from SDR. There are recommendations for degrees of freedom for clustered and stratified sample designs, for example, see Valiant and Rust (2010), Korn and Graubard (1999), and Eltinge and Jang (1996). However, the degrees of freedom for clustered and stratified sample designs do not apply to the one-stage systematic random sample design employed by NYCHVS. Huang and Bell (2011) examined degrees of freedom for SDR but they do not provide general guidance. The second reason for our suggestion is that the American Community Survey (ACS) has the same sample design and also suggests using *z*-values from a normal distribution (U.S. Census Bureau 2014). Until more research is done in this area, we suggest following the lead of ACS.

We have not covered hypothesis testing but we do point the reader to Schenker and Gentleman (2001) who discuss the pitfalls of using confidence intervals in hypothesis testing.

Table 7 is a compilation of examples from throughout the guide. For each of these examples, we calculate a 95 percent confidence interval.

Table 7: Summary Table of Examples Estimated Confidence Intervals

Example	Example Number				Estimate	Standard Error	95% Confidence Intervals
	PROC SURVEY	Base SAS	STATA	R			
Total		4.1	5.1	6.1	3,109,955	8,627	(3,093,046, 3,126,863)
Difference		4.2	5.2	6.2	15,497	10,420	(-4,926, 35,920)
Mean	3.1	4.3	5.3	6.3	$1,694	$11	($1,673, $1,714)
Median		4.4	5.4	6.4	$1,450	$7	($1,435, $1,465)
Regression parameter	3.2	4.5	5.5	6.5	28.80	0.81	(27.20, 30.38)
Longitudinal change		4.6	5.6	6.6	2.2%	0.3%	(1.7%, 2.8%)
Percent change		4.7	5.7	6.7	9.4%	0.7%	(8.0%, 11.0%)

Source: U.S Census Bureau, 2017 New York City Housing and Vacancy Survey.

The standard errors of Table 7 were produced with SAS and Stata and not with R.

The subsequent examples of this section demonstrate how to estimate CIs using SAS, Stata and R with some of the earlier examples.

Example 7.1. Estimating CIs of a Total with Base SAS

In this example, we demonstrate how users calculate the CIs of the total in Example 3.1`. Figure 7.1.1 shows the code in SAS, and Figure 7.1.2 shows the output.

Figure 7.1.1: SAS Code for Estimating Variance of a Total with Confidence Intervals

```
data data6 (keep = est var se lowerci upperci) ;
  set data5 end=eof ;
  * Fill array with the replicate sums ;
  array repwts{80} rw1-rw80 ;
  * Fill array with the squared diffs ;
  array sdiffsq{80} sdiffsq1-sdiffsq80 ;
  do j = 1 to 80 ;
    sdiffsq{j} = (repwts{j} - est)**2 ;
    end ;
  * Sum the squared diffs ;
  totdiff = sum(of sdiffsq1-sdiffsq80) ;
  var = (4/80) * totdiff ;
  se = (var)**(0.5) ;

  *Calculate the endpoints of the confidence intervals. ;
  lowerci= est - 1.96 * se;
  upperci = est + 1.96 * se;
  output ;
run ;

proc print data=data6 noobs ;
  var est se lowerci upperci;
  format est se lowerci upperci comma12.2 ;
run ;
```

In Figure 7.1.2, we used the simple numeric value of 1.96 as the *z*-value with $\alpha = 0.05$ level because it makes the code easier to read. We could replace this with the more exacting SAS function `quantile ("Normal", 1 - alpha/2)` where `alpha = 0.05`.

Figure 7.1.2: SAS Output of Variance of Total with Confidence Intervals

est	se	lowerCI	UpperCI
3,109,954.63	8,626.87	3,093,045.96	3,126,863.29

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

In addition to the output we saw in Example 3.1, now we get the CIs as well: the 95 percent CI for the total number of occupied housing units is (3,093,0456, 3,126,863).

Example 7.2. Estimating CIs of a Regression Parameter with SAS PROC SURVEYREG

In this example, we demonstrate how users can use the SAS PROC SURVEYREG in Example 3.3 earlier, and add a `clparm` option in the `model` statement to get the estimates for confidence intervals.

Figure 7.2.1: SAS Code for Estimating Variance of Regression Parameters with Confidence Intervals

```
proc surveyreg data=Data3 varmethod=brr(fay) ;
  domain tenure ;
  model rent = yr_movein / solution clparm ;
  weight fw0 ;
  repweights fw1-fw80 ;
  ods output parameterEstimates = MyParmEst ;
run;

data MyParmEstfin ;
  set MyParmEst ;
  if _n_ = 2 ;
  se = stderr ;
  var = se**2 ;
  drop parameter dendf tvalue probt stderr ;
run ;

proc print data=myparmestfin;
  var estimate se lowercl uppercl ;
  format estimate se var lowercl uppercl 8.4 ;
run;
```

The SAS option `clparm` in the model statement of Figure 7.2.1. request the confidence interval for the regression parameters. The procedure generates the confidence interval as the variables `lowercl` and `uppercl`. The SAS code from Figure 7.2.1 produces the output of Figure 7.2.2.

Figure 7.2.2: SAS Output for Estimating Confidence Intervals Using t-Distribution

Estimate	se	lowercl	uppercl
28.7885	0.8122	27.1722	30.4049

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The point estimate and variances are the same as in Example 3.3, however, there are the two added columns at the end of the output for the confidence intervals. Please note that the confidence intervals constructed using SAS SURVEYREG is defaulted to t-distribution.

To construct confidence intervals using the normal distribution, users have to code this manually in Base SAS. Figure 7.2.3 shows the SAS code needed, in addition to Figure 7.2.1, to get the confidence intervals using normal distribution, and Figure 7.2.4 shows the results.

Figure 7.2.3: SAS Code for Estimating Variance of Regression Parameters with Confidence Intervals

```
*Calculate CI using normal distribution;
data one;
      set myparmestfin;
lowerci= estimate - 1.96*se;
upperci= estimate + 1.96*se;
run;
proc print; var estimate se lowerci upperci; run;
```

Figure 7.2.4: SAS Output for Estimating Confidence Intervals Using z-Distribution

Estimate	se	lowerci	upperci
28.7885	0.8122	27.1966	30.3805

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Example 7.3. Estimating CIs of a Total with Stata Manually

Users might notice the Stata outputs in Section 5, already include CIs. We now demonstrate how users can manually calculate CIs in Stata for Example 5.1. Figure 7.3.1 shows the code and Figure 7.3.2 provides the resultant output.

Figure 7.3.1.: Stata Code for Estimating Confidence Intervals Manually

```
* Calculating 2017 occupied totals:
quietly summarize fw if occ_final == 1
scalar occ_est17 = r(sum)

* Creating matrix that stores 2017 estimates:
matrix occ_ests_17 = (occ_est17)

* Now going through replicates:
foreach x of var fw1-fw80 {

      quietly summarize `x' if occ_final == 1
      scalar occ_est17_i = r(sum)

      * Adding to matrix that stores 2017 estimates:
      matrix occ_ests_17 = (occ_ests_17 \ occ_est17_i)
}

* Now getting squarred differences for variance estimation:
scalar var = 0
quietly forvalues i = 2/81 {
      scalar sq_pt_occ_est = (occ_ests_17[`i',1] - occ_ests_17[1,1])^2
      scalar var = var + sq_pt_occ_est
}

scalar var = (4/80) * var
scalar se_est = (var)^(0.5)

* Final Point Estimate:
display occ_ests_17[1,1]

* Final Estimated Standard Error:
display se_est

* 95% CI using standard normal (z) distribution:
scalar z_value = invnorm(.975)
display "(",occ_ests_17[1,1]-
z_value*se_est,",",occ_ests_17[1,1]+z_value*se_est,")"
```

Figure 7.3.2: Stata Output for Estimating Confidence Intervals Manually

```
. * Final Point Estimate:
. display occ_ests_17[1,1]
3109954.6

.
. * Final Estimated Standard Error:
. display se_est
8626.87

. display "(",occ_ests_17[1,1]-
z_value*se_est,",",occ_ests_17[1,1]+z_value*se_est,")"
( 3093046.3 , 3126863 )
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

From Figure 7.3.2, we see that the estimated total occupied housing units in NYC is 3,109,955 and its standard error is 8,627. There is a 95 percent chance that the sample total of occupied HUs in NYC would be between 3,093,046 and 3,126,863.

Comparing Stata's output with other methods, Stata produced the same estimate, standard error, and confidence intervals as the other methods.

Example 7.4. Estimating the CI of a Total with R

In R, users need to manually code the formulas for CIs. In this example, we demonstrate how users can manually calculate the estimated total occupied HUs in NYC and its standard errors for Example 6.1, as well as the 95 percent CIs. In next example (Example 7.5), we will demonstrate how to manually calculate the CIs.

Figure 7.4.1: R Code for Estimating Confidence Intervals

```
# Vector storing replicates:
reps <- c()
occ_data <- subset(hu, occ_final == 1)
est <- sum(occ_data[,1027])

# Looping calculations for variance estimate - (make sure column numbers
align with your dataset):
for (i in c(1:80)) {
  temp_est <- sum(occ_data[,(i+946)])

  # Adding Result to Vector of Squarred Difference Results:
  reps <- rbind(reps, (temp_est - est)^2)
}

est_var <- (4/80) * sum(reps)
est_se <- sqrt(est_var)
est_se
```

```
# For a 95% CI, we need 2.5% on either tail, so get the 97.5% percentile
value of a standard normal:
z_value <- qnorm(.975)

# Finally, the 95% CI using z distribution:
ci_95 <- c(est - z_value*est_se, est + z_value*est_se)
ci_95
```

The R code provided in Figure 7.4.1 produces the R output of Figure 7.4.2.

Figure 7.4.2: R Output for Estimating Confidence Intervals

```
                total              SE
occ_final     3,109,955      8,626.87

> est
[1] 3,109,955

> est_se
[1] 8626.87

z_value
1.959964

> ci_95
[1] 3093046 3126863
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

From Figure 7.4.2, we see that R produce the same exact estimate, standard error, and confidence intervals as the other methods.

8. How to Calculate Confidence Intervals for the Odds Ratio

This extended example shows how to calculate and how not to calculate a CI for the odds ratio. We show this by addressing the question "**Does rent stabilization status (pre-1947 or post-1947) affect the probability of having 3 or more maintenance deficiencies?**" with results from the 2017 NYCHVS. This example is different and much longer than Example 7.1 for two reasons: (1) the complexity of the estimator and the variance of the estimator and (2) we show how to calculate the CIs correctly, and several ways the confidence intervals should not be calculated.

As with the other statistics of this guide, using the replicate weights will best estimate the variance of the odds ratio. Table 8.1 summarizes the alternative methods that we will discuss and provides an overview of the estimates for each method.

Table 8.1: Examples of Odds Ratio

Example	Uses	Software	Odds Ratio $(\hat{\theta})$	$\hat{v}(\log \text{odds})$	$\hat{se}(\log \text{odds})$	95% Confidence Interval of $\hat{\theta}$
8.1	Replicate Wgts	SURVEYLOGISTIC	2.0947	*	*	(1.6940, 2.5902)
8.2	Replicate Wgts	SURVEYFREQ	2.0947	*	*	(1.6869, 2.6012)
8.3	Replicate Wgts	Base SAS	2.0947	0.0114	0.1067	(1.6993, 2.5821)
8.4	Replicate Wgts	Stata	2.0947	0.0114	0.1067	(1.6994, 2.5821)
8.5	Replicate Wgts	R	2.0947	0.0112	0.1060	(1.7019, 2.5785)
8.6	Final Wgts	SURVEYFREQ	2.0947	*	*	(1.6768, 2.6169)
8.7	Normalized Wgts	FREQ	2.0947	0.0125	0.1120	(1.6818, 2.6091)
8.8	Final Wgts	FREQ	2.0947	0.0000	0.0073	(2.0651, 2.1248)
8.9	Unweighted	FREQ	2.0608	0.0126	0.1121	(1.6543, 2.5672)

* SAS SURVEYLOGISTIC and SURVEYFREQ procedures do not output variance or standard errors.
Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

We make a few observations about the results of Table 8.1.

First, the weighted and unweighted estimates of the odds ratio differ slightly: the unweighted estimate of the odds ratio is 2.0608 and the weighted estimate is 2.0947. Using the weights accounts for the fact that sample units contribute unequally to the estimates.

With the variance of the log odds and the standard errors, we note the following differences from the Table 8.1:

- Base SAS and Stata (Examples 8.3 and 8.4) both produce approximately the same estimate of the odds ratio and standard error of the log odds using the replicate weights.
- Estimating the variance using normalized weights and no weights, or Example 8.7 and 8.9, respectively, produced variances that are an overestimate. Using final weights or

Example 8.8 underestimated variance. Only using the replicate weights or Examples 8.1-8.5 fully accounts for the complex sample design of NYCHVS.
- Using the final weights with PROC SURVEYFREQ and the normalized weights with PROC FREQ or Examples 8.6 and 8.7, respectively, provide same estimate but we caution that it may not be true with all analyses.
- Example 8.8 using the final weights, underestimates the variance by a large amount.

With the CI calculations, we note the following differences in Table 8.1.

- SAS PROC SURVEYLOGISTIC and SAS PROC SURVEYFREQ (Examples 8.1) produced different CIs because the PROC SURVEY procedures uses a critical value from a t-distribution with 80 degrees of freedom. Basically, SURVEYLOGISTIC is assuming that we have a stratified sample design with 80 strata. See Section 3 for a discussion.
- The CIs produced by SURVEYFREQ procedure (Example 8.2) are different from the results of SURVEYLOGISTIC (Examples 8.1) because SURVEYFREQ uses a different approximation in the variance calculation, which is not output by the procedure, but you see the effect in the difference in the confidence interval calculation. This is discussed more in Section 8.2.
- Base SAS and Stata (Examples 8.3 and 8.4) both produced approximately the same CIs using replicate weights.
- Estimating the variance using final weights or Example 8.8 underestimated variance by a large amount, which therefore resulted a much narrower CI.

The remainder of this extended example is organized as follows. First, we generally review the odds ratio. This is provided to define our use of different terms and for readers not familiar with an odds ratio. Readers familiar with this material can skip this section.

Then each of the nine methods for estimating the CI is reviewed separately. Although we say that only Examples 8.1-8.5 estimate the CI accurately because replicate weights account for the complex sample design of NYCHVS, we provide the four other methods because some data users are familiar with them and they provide excellent comparisons to Examples 8.1-8.5.

We have color-coded the odds ratio, variance of the log odds ratio, and the confidence interval of the odds ratio throughout the example as green, purple, and light blue, respectively. This was done to help keep track of the different statistics of the examples.

Review of Odds Ratio and Log Odds

Suppose we have two sets of binary random numbers, X and Y, and we want to measure the association between the two variables. We assume both random variables have a Bernoulli distribution. Table 8.2 shows the frequencies observed in the sample for all possible combinations of *X* and *Y*.

Table 8.2: Layout of Two Binary Variables

	$X = 0$	$X = 1$	Total
$Y = 0$	n_{00}	n_{01}	n_{0+}
$Y = 1$	n_{10}	n_{11}	n_{1+}
Total	n_{+0}	n_{+1}	n

The *odds* of a single binary random variable Y are defined to be the probability of $Y = 1$ divided by the probability of $Y = 0$, i.e.,

$$\{\text{Odds of } Y\} = \frac{P(Y=1)}{P(Y=0)}$$

In the presence of the variable X, the odds of Y given $X = 0$ is

$$\{\text{Odds of } Y \text{ given } X=0\} = \frac{P(Y=1 \mid X=0)}{P(Y=0 \mid X=0)}$$

and the odds of Y given $X = 1$ is

$$\{\text{Odds of } Y \text{ given } X=1\} = \frac{P(Y=1 \mid X=1)}{P(Y=0 \mid X=1)}$$

The vertical bar in these probability statements is the traditional symbol for conditional probabilities.

The odds ratio θ compares the odds of Y for $X = 0$ and $X = 1$ and is defined as the ratio of the two odds or:

$$\theta = \frac{\{\text{Odds of } Y \text{ given } X=0\}}{\{\text{Odds of } Y \text{ given } X=1\}} = \frac{\dfrac{P(Y=1 \mid X=0)}{P(Y=0 \mid X=0)}}{\dfrac{P(Y=1 \mid X=1)}{P(Y=0 \mid X=1)}} = \frac{P(Y=1 \mid X=0)P(Y=0 \mid X=1)}{P(Y=0 \mid X=0)P(Y=1 \mid X=1)}$$

An odds ratio of $\theta = 1$ suggests that the odds of Y is the same for $X = 0$ and $X = 1$.

In the presence of the variable X, the odds of Y given $X = 1$ is

$$\{\text{Odds of } Y \text{ given } X=1\} = \frac{(n_{11}/n_{+1})}{(n_{01}/n_{+1})} = \frac{n_{11}}{n_{01}} \qquad (8.1)$$

and we can estimate the odds ratio as

$$\hat{\theta} = \frac{(n_{11}/n_{+1})(n_{00}/n_{+0})}{(n_{01}/n_{+1})(n_{10}/n_{+0})} = \frac{n_{11} \cdot n_{00}}{n_{01} \cdot n_{10}} \tag{8.2}$$

The distribution of the odds ratio statistic has an atypical form, which unfortunately makes it hard to use for the purpose of statistical inferences. The natural log of the odds ratio or simply log odds ratio, by contrast, is approximately normally distributed, provided the cell counts within each of the four categories above are sufficiently high. Consequently, the variance of the log odds can be estimated as

$$\hat{v}(\log\ \text{odds}) = \frac{1}{n_{00}} + \frac{1}{n_{01}} + \frac{1}{n_{10}} + \frac{1}{n_{11}} \tag{8.3}$$

We can use the variance of the log odds to calculate a (1 - α) CI of the odds ratio by transforming the CI of the log odds as

$$\left(e^{\log\ \text{odds} + z_{\alpha/2} \cdot \sqrt{\hat{v}(\log\ \text{odds})}},\ e^{\log\ \text{odds} - z_{\alpha/2} \cdot \sqrt{\hat{v}(\log\ \text{odds})}} \right) \tag{8.4}$$

The Maintenance Deficiencies Example. Returning to maintenance deficiencies example, we ask the question:

"**Does rent stabilization status affect the probability of having 3 or more maintenance deficiencies?**"

To answer this, we have a binary random variable *Y* with *Y* = 1 indicating "having 3 or more maintenance deficiencies" and *Y* = 0 indicating "having fewer than 3 maintenance deficiencies"

$$Y = \begin{cases} 1, & \text{HU has 3 or more maintenance deficiencies} \\ 0, & \text{HU has fewer than 3 maintenance deficiencies} \end{cases}$$

We also define a second binary random variable *X* for rent stabilization status as

$$X = \begin{cases} 1, & \text{HU has rent stabilized pre} - 1947 \\ 0, & \text{HU has rent stabilized post} - 1947 \end{cases}$$

If we calculate the odds ratio *Y* given *X*, a value of $\theta > 1$ implies that rent stabilized pre-1947 HUs are more likely than the rent stabilized post-1947 units to have 3 or more maintenance deficiencies and, conversely, $\theta < 1$ implies rent stabilized post-1947 units are more likely to have 3 or more maintenance deficiencies.

Example 8.1. Estimating a CI of an Odds Ratio with Replicate Weights and SAS PROC SURVEYLOGISTIC

The easiest way of estimating odds ratios with replicate weights is to use SAS SURVEYLOGISTIC. This procedure produces the odds ratio estimates and CIs with replicate weights automatically. Use the `varmethod brr(fay)` option and the default Fay coefficient of 0.5. Put the effect (rent stabilization status, in our example) in the `class` statement.

Figure 8.1.1 provides the PROC SURVEYLOGISTIC code that will calculate the odds ratios using the replicate weights.

Figure 8.1.1: SAS Code for Calculating Confidence Interval of Odds Ratio with Replicate Weights

```
Data Def_analysis ;
  set HU17;
  * Exclude the vacant HUs and the ones did not report on deficiencies. ;
  where rec53 ne '9' and (sc115='1' or sc116 in ('2','3')) and uf_csr in
('30','31') ;
  if rec53 in ('4','5','6','7','8') then def='1'; else def='0' ;
  * pre-1947 ;
  If uf_csr='30' then stabilized='1' ;
  * post1947 ;
  else if uf_csr ='31' then stabilized='0' ;
run ;

proc surveylogistic data=def_analysis varmethod=brr(fay=0.5) ;
  class stabilized ;
  model def=stabilized ;
  weight fw0 ;
  repweights fw1-fw80 ;
run ;
```

Figure 8.1.2 shows some of the output that is generated by the SAS code of Figure 8.1.1.

Figure 8.1.2: Partial SAS Output for Calculating Confidence Interval of Odds Ratio with Replicate Weights

The SURVEYLOGISTIC Procedure

Model Information

Data Set	WORK.DEF_ANALYSIS
Response Variable	Def
Number of Response Levels	2
Weight Variable	FWO
Model	Binary Logit
Optimization Technique	Fisher's Scoring

Number of Observations Read	3523
Number of Observations Used	3523
Sum of Weights Read	837828.6
Sum of Weights Used	837828.6

Response Profile

Ordered Value	Def	Total Frequency	Total Weight
1	0	2895	689061.98
2	1	628	148766.65

Probability modeled is Def='0'.

Variance Estimation

Method	BRR
Replicate Weights	DEF_ANALYSIS
Number of Replicates	80
Fay Coefficient	0.5

Odds Ratio Estimates

Effect	Point Estimate	95% Confidence Limits	
stabilized 0 vs 1	2.095	1.694	2.590

NOTE: The degrees of freedom in computing the confidence limits is 80.

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The bottom of Figure 8.1.2 mentions that 80 degrees of freedom were used to calculate the confidence interval. This tells us that the critical value was 1.99 (from a t-distribution with 80 degrees of freedom) instead of 1.96 (from a normal distribution). Although we are using SDR replicate weights, SAS assumes that we are using Balanced Repeated Replication (BRR) replicate weights. For this reason, the estimates from PROC SURVEYLOGISTIC are slightly different than the other methods.

Example 8.2. Estimating a CI of an Odds Ratio with Replicate Weights and SAS PROC SURVEYFREQ

Another similar method as SURVEYLOGISTIC to estimate the CI for the odds ratio with the replicate weights in SAS, is to use the procedure PROC SURVEYFREQ.

Figure 8.2.1 provides the PROC SURVEYFREQ code that will calculate the odds ratio using the replicate weights.

Figure 8.2.1: SAS Code for Calculating Confidence Interval of Odds Ratio with Replicate Weights

```
proc surveyfreq data=def_analysis varmethod=brr(fay) ;
  tables def * stabilized / or ;
  weight fw0 ;
  repweights fw1-fw80 ;
run ;
```

The use of the `or` option in the `tables` statement requests that SAS produce the odds ratio with both PROC SURVEYFREQ and PROC FREQ. Note that the output generated by the option `or` is formatted differently with PROC SURVEYFREQ and PROC FREQ.

The SAS code of Figure 8.2.1 generates the output of Figure 8.2.2.

Figure 8.2.2: SAS Output for Calculating Confidence Interval of Odds Ratio with Replicate Weights

The SURVEYFREQ Procedure
Data Summary

Number of Observations	2523
Sum of Weights	837828.631

Variance Estimation

Method	BRR
Replicate Weights	DEF_ANALYSIS
Number of Replicates	80
Fay Coefficient	0.500

Table of Def by Stabilized

Def	Stabilized	Frequency	Weighted Frequency	Std Err of Wgt Freq	Percent	Std Err of Percent
0	0	888	214719	8110	25.6280	0.8476
	1	2007	474343	13632	56.6158	1.0282
	Total	2895	689062	16064	82.2438	0.8313
1	0	111	26435	2487	3.1552	0.3082
	1	517	122331	5823	14.6010	0.7043
	Total	628	148767	6681	17.7562	0.8313
Total	0	999	241154	8181	28.7832	0.8797
	1	2524	596674	13774	71.2168	0.8797
	Total	3523	837829	15216	100.0000	

Odds Ratio and Relative Risks (Row1/Row2)

Statistic	Estimate	95% Confidence Limits	
Odds Ratio	2.0947	1.6869	2.6012
Column 1 Relative Risk	1.7536	1.4734	2.0871
Column 2 Relative Risk	0.8371	0.8018	0.8741

Sample Size = 3523

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

The estimate of $\hat{\theta}$ = 2.0947 implies that housing units with rent stabilized prior to 1947 are about twice as likely to have 3 or more maintenance deficiencies than HUs with rent stabilized post-1947.

Please note that the CIs produced by SURVEYFREQ here are slightly different from the CIs produced by the other methods with replicate weights. This is due to two reasons: (1) the PROC SURVEY procedures use a critical value in the confidence intervals from a t-distribution with 80

degrees of freedom instead of using a critical value from a normal distribution) and (2) SURVEYFREQ uses a variance approximation when calculating the variance of the log odds, which is

$$\hat{v}(\log odds) = \frac{\hat{v}(odds\,ratio)}{(odds\,ratio)^2},$$

while the other methods calculate the variance of log odds directly using Equation (8.3) (SAS 9.2 User's Guide, 2019).

Example 8.3. Estimating a CI of an Odds Ratio with Replicate Weights and Base SAS

A point estimate of the odds ratio is straightforward to calculate with weights – it just uses the sum of all the weights in each of the four categories and follows the procedure above. The variance estimate is more complicated to calculate manually.

Table 8.3.1 provides the weighted estimates of the frequencies of each cell. The final weight and Equation (8.2) were used to produce the totals.

Table 8.3.1: Estimated Counts for Maintenance Deficiencies Example

	Fewer than 3 maintenance deficiencies	3 or more maintenance deficiencies	Total
Rent Stabilized Post 1947	n_{00} = 214,719	n_{01} = 474,343	n_{0+} = 689,062
Rent Stabilized Pre-1947	n_{10} = 26,435	n_{11} = 122,331	n_{1+} = 148,767
Total	n_{+0} = 241,154	n_{+1} = 596,674	n = 837,829

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Using the weighted totals, the odds of having 3 or more maintenance deficiencies given the HU is rent stabilized prior to 1947 is calculated as

$$\{\text{Odds of } Y \text{ given Rent Stabilized Pre} - 1947\ (X = 1)\} = \frac{122{,}331}{474{,}343} = 0.2579$$

and we calculate the odds ratio as

$$\hat{\theta} = \frac{214{,}719 * 122{,}331}{474{,}343 * 26{,}435} = 2.0948$$

The natural log of the odds ratio is *log odds* = 0.7395. To estimate the variance of the log odds, the log odds is calculated for each replicate and Equation (2.1) is applied. This gives $\hat{v}(\log \text{odds})$ = 0.0114, which can be used with Equation (8.3) to calculate the CI:

$$\left(e^{0.7395-1.96\sqrt{\hat{v}(\log odds)}}, e^{0.7395+1.96\sqrt{\hat{v}(\log odds)}}\right)$$

The SAS code of Figure 8.3.3 shows how to calculate the variance, standard error, and CIs of the odds ratio directly.

Figure 8.3.3: SAS Code for Calculating Confidence Interval of Odds Ratio with Replicate Weights

```
* Define 4 categories. ;
data def_analysis ;
      set def_analysis ;
      if def='1' and stabilized='0' then n_xx='n_10' ;
      else if def='1' and stabilized='1' then n_xx='n_11' ;
      else if def='0' and stabilized='0' then n_xx='n_00' ;
      else if def='0' and stabilized='1' then n_xx='n_01' ;
run ;

* Calculate the total sums for each of our 4 categories ;
proc means data = Def_analysis sum noprint ;
  class n_xx ;
  var fw0 fw1-fw80 ;
  output out = def_collapse sum = est rw1 - rw80 ;
run ;

* Drop the first observation of totals ;
data def_collapse ;
  set def_collapse ;
  if _TYPE_ = 0 then delete ;
run ;

* Reshape into wide format to calculate ratio estimates
  with arrays later ;
proc transpose data = def_collapse out = def_collapse_transposed ;
  by _TYPE_ ;
  id n_xx ;
  var est rw1 - rw80 ;
run ;

* Now calculating log odds ratio estimate for newly transposed data. ;
data def_collapse_transposed_odds ( keep = _NAME_ dummy_id ln_odds_ratio );
  set def_collapse_transposed;

  * Formula for odds ratio ;
  ln_odds_ratio = log( (n_00 * n_11) / (n_01 * n_10) ) ;

  * ID var needed for tranposition later. ;
  dummy_id = 1 ;
```

Figure 8.3.3: SAS Code for Calculating Confidence Interval of Odds Ratio with Replicate Weights

```
run ;

* Transpose back into wide format for array calculations. ;
proc transpose data = def_collapse_transposed_odds out =
def_fay_brr_estimates;
  by dummy_id ;
  id _NAME_ ;
  var ln_odds_ratio ;
run;

* Calculate variance estimate. ;
data fay_estimate_final (keep = est var se odds_est lowerci upperci) ;
  set def_fay_brr_estimates (drop = _NAME_);
  * Filling in array with odds ratio estimates ;
  array repwts{80} rw1 - rw80 ;

  * Fill in other array with squared differences ;
  array sqdiff{80} sqdiff1 - sqdiff80 ;

  * Looping to create differences ;
  do i = 1 to 80;
    sqdiff[i] = (repwts[i] - est) ** 2 ;
  end;

  * Sum differences. ;
  total_diff = sum(of sqdiff1-sqdiff80) ;

  * Calculate variance. ;
  var = (4/80)*  total_diff ;

  * Calculate standard error. ;
  se = var ** (0.5) ;

  * Create CI in odds ratio form. ;
  odds_est = exp(est);
  lowerci = exp((est - (1.96 * se))) ;
  upperci = exp((est + (1.96 * se))) ;
run ;

* Print final log odds ratio estimate, the variance, SE and
  then the odds ratio estimate with it's 95% CI ;
proc print data = fay_estimate_final noobs ;
  format est  se lowerci upperci odds_est 8.4;
run;
```

The SAS code of Figure 8.3.3 generates the output of Figure 8.3.4. These are slightly different than PROC SURVEYFREQ due to rounding.

Figure 8.3.4: SAS Output for Calculating Confidence Interval of Odds Ratio with Replicate Weights

est	se	odds_est	lowerci	upperci
0.7394	0.1067	2.0947	1.6993	2.5821

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Example 8.4. Estimating a CI of an Odds Ratio with Replicate Weights and Stata

This example estimates the odds ratio and variance of rent stabilization status (pre-1947 or post-1947) and the presence of 3 or more maintenance deficiencies. The code in Figure 8.4.1 identifies the domains of interest, as well as the odds ratio and variance calculations using replicate weights in Stata.

Figure 8.4.1: Stata Code for Calculating Confidence Interval of Odds Ratio with Replicate Weights

```
* Binary variable for three or more deficiencies:
gen defect = (rec53 == 4 | rec53 == 5 | rec53 == 6 | rec53 == 7 | rec53 == 8)

* Need to exclude non-reported defects:
gen occ_reported_defects = (occ_final == 1 & rec53 != 9)

* Need binaries for time demarcation:
gen time_stable = 1 if uf_csr == 30
replace time_stable = 0 if uf_csr == 31

* Now running log odds ratio:
svy, subpop(occ_reported_defects): logit defect time_stable

* Now for the odds ratio:
svy, subpop(occ_reported_defects): logistic defect time_stable
```

The Stata code of Figure 8.4.1 generates the output of Figure 8.4.2.

Figure 8.4.2: Stata Output for Calculating Confidence Interval of Odds Ratio with Replicate Weights

```
Log Odds Ratio:

Survey: Logistic regression                     Number of obs     =      6,910
                                                Population size   =  1,552,419
                                                Subpop. no. obs   =      3,523
                                                Subpop. size      = 837,828.63
                                                Replications      =         80
                                                Wald chi2(1)      =      48.00
                                                Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |                 SDR *
      defect |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 time_stable |   .7394279   .1067255     6.93   0.000     .5302497    .9486061
       _cons |  -2.094626    .107582   -19.47   0.000    -2.305483   -1.883769
------------------------------------------------------------------------------


Odds Ratio:

Survey: Logistic regression                     Number of obs     =      6,910
                                                Population size   =  1,552,419
                                                Subpop. no. obs   =      3,523
                                                Subpop. size      = 837,828.63
                                                Replications      =         80
                                                Wald chi2(1)      =      48.00
                                                Prob > chi2       =     0.0000


------------------------------------------------------------------------------
             |                 SDR *
      defect | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 time_stable |   2.094737   .2235618     6.93   0.000     1.699357    2.582108
       _cons |   .1231163   .0132451   -19.47   0.000     .0997106    .152016
------------------------------------------------------------------------------
Note: _cons estimates baseline odds.
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Figure 8.4.2 shows that the odds ratio estimate from Stata is $\hat{\theta} = 2.0947$ which implies that housing units rent stabilized prior to 1947 are about twice as likely to have 3 or more maintenance deficiencies than HUs rent stabilized post-1947. The standard error of the log odds is 0.1067, which is approximately the same as the SAS method.

Example 8.5. Estimating a CI of an Odds Ratio with Replicate Weights and R

This example estimates the odds ratio and variance calculation of rent stabilization status (pre-1947 or post-1947) and the presence of 3 or more maintenance deficiencies. The code in

Figure 8.5.1 identifies the domains of interest, and the odds ratio and variance estimate using replicate weights in R.

Figure 8.5.1: R Code for Calculating Confidence Interval of Odds Ratio with Replicate Weights

```
occ_reported_defects <- subset(hu_design, occ_final == 1 & REC53 != 9)
odds_ex <- svyglm(defect~time_stable, design=occ_reported_defects,
family=quasibinomial)

summary(odds_ex)

# Point Estimate of Odds Ratio:
pt_est <- exp(odds_ex$coefficients[2])
pt_est

# SE of Log Odds:
log_se <- coef(summary(odds_ex))[2,2]
log_se

# SE of Odds Ratio:
# Note: Converting standard errors from log odds back to odds ratios are a
little tricky:
odds_se <- sqrt((exp(odds_ex$coefficients[2])^2) *
(coef(summary(odds_ex))[,2]^2))[2]
odds_se

# 95% Confidence Intervals for Odds Ratios:
lower_ci <- exp(odds_ex$coefficients[2] - (1.96*log_se))
upper_ci <- exp(odds_ex$coefficients[2] + (1.96*log_se))
c(lower_ci, upper_ci)
```

The R code of Figure 8.5.1 generates the output of Figure 8.5.2.

Figure 8.5.2: R Output for Calculating Confidence Interval of Odds Ratio with Replicate Weights

```
Call:
svyglm(formula = defect ~ time_stable, design = occ_reported_defects,
    family = quasibinomial)

Survey design:
subset(hu_design, occ_final == 1 & REC53 != 9)

Coefficients:
             Estimate  Std. Error   t value    Pr(>|t|)
(Intercept)  -2.0946    0.1040     -20.150    < 2e-16 ***
time_stable   0.7394    0.1060      6.978      8.66e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 0.3041508)

Number of Fisher Scoring iterations: 5

> pt_est
time_stable
   2.094737 <- Odds Ratio Point Estimate

> log_se
[1] 0.1059598 <- Log Odds SE

> odds_se
time_stable
   0.221958 <- Odds Ratio SE

> c(lower_ci, upper_ci)
time_stable time_stable
   1.701902    2.578246 <- 95% CI for Odds Ratios
```

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

From the output in Figure 8.5.2, we can see that the odds ratio estimate is $\hat{\theta} = 2.0947$, the standard error of the log odds is 0.1060, and the confidence interval of the odds ratio is (1.7019, 2.5782).

Example 8.6. Estimating a CI of an Odds Ratio with Final weights and SAS PROC SURVEYFREQ

If you merely use the final weights with the PROC SURVEYFREQ command, omitting the replicate weights and the `varmethod` option then PROC SURVEYFREQ defaults to the Taylor Linearization Method for estimating the variance of the odds ratio. You will get the same point estimate and estimated counts as Example 8.1, but the CI will be slightly narrower.

Figure 8.6.1 provides the PROC SURVEYFREQ code that will calculate the odds ratio using just the final weights.

Figure 8.6.1: SAS Code for Calculating Confidence Interval of Odds Ratio with Final weights

```
proc surveyfreq data=def_analysis ;
  table def * stabilized / or ;
  weight fw0 ;
run ;
```

The code of Figure 8.6.1 generates the output of Figure 8.6.2.

Figure 8.6.2: SAS Output for Calculating Confidence Interval of Odds Ratio with Final weights

The SURVEYFREQ Procedure

Data Summary

Number of Observations	3523
Sum of Weights	837828.631

Table of Def by Stabilized

Def	Stabilized	Frequency	Weighted Frequency	Std Err of Wgt Freq	Percent	Std Err of Percent
0	0	888	214719	6344	25.6280	0.7496
	1	2007	474343	7163	56.6158	0.8470
	Total	2895	689062	5807	82.2438	0.6511
1	0	111	26435	2501	3.1552	0.2983
	1	517	122331	5044	14.6010	0.6014
	Total	628	148767	5465	17.7562	0.6511
Total	0	999	241154	6579	28.7832	0.7765
	1	2524	596674	6618	71.2168	0.7765
	Total	3523	837829	2317	100.0000	

Odds Ratio and Relative Risks (Row1/Row2)

Statistic	Estimate	95% Confidence Limits	
Odds Ratio	2.0947	1.6768	2.6169
Column 1 Relative Risk	1.7536	1.4656	2.0982
Column 2 Relative Risk	0.8371	0.8007	0.8753

Sample Size = 3523

Source: U.S. Census Bureau, 2017, New York City Housing and Vacancy Survey.

Example 8.7. Estimating a CI of an Odds Ratio with Normalized Weights and SAS PROC FREQ

In this estimation of the CI, we estimate the variance with the normalized weights, which reduces the underestimation of the variance that we will see in estimating a CI with final weights, or Example 8.8. Normalized weights are not provided by NYCHVS, but can easily be calculated as the final weight divided by the mean of all of the final weights with a positive value.

Normalizing the final weight produces a weight that accounts for the unequal weights that reflect that sample units contribute unequally to the estimates. The normalized weights have a mean value of 1.0. So the normalized weights have the same magnitude as if you didn't use weights – all sample units have a weight of 1.0 – and the variance (Example 8.7) produces a more reasonable estimate of the variance as compared to Example 8.8.

Table 8.7.1 provides the weighted estimates of the frequencies of each cell. The final weight and Equation (8.2) were used to produce the totals.

Table 8.7.1: Estimated Counts for Maintenance Deficiencies Example

	Fewer than 3 maintenance deficiencies	3 or more maintenance deficiencies	Total
Rent Stabilized Post 1947	n_{00} = 902.88	n_{01} = 1,994.57	n_{0+} = 2897.45
Rent Stabilized Pre-1947	n_{10} = 111.16	n_{11} = 514.39	n_{1+} = 625.55
Total	n_{+0} = 1014.03	n_{+1} = 2508.97	n = 3523.00

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Using the weighted totals, the odds of having 3 or more deficiencies given the HU is rent stabilized pre-1947 is calculated as

$$\{\text{Odds of Y given Rent Stabilized Pre} - 1947\ (X = 1)\} = \frac{514.39}{1{,}994.57} = 0.2579$$

and we calculate the odds ratio as

$$\hat{\theta} = \frac{514.39 * 902.88}{1{,}994.57 * 111.16} = 2.0947$$

Figure 8.7.1 provides the PROC FREQ code that will calculate the odds ratio using the normalized weights.

Figure 8.7.1: SAS Code for Calculating Confidence Interval of Odds Ratio with Normalized Weights

```
data def_analysis ;
  set def_analysis ;
  * This is the formula for normalizing weights
    Number of observations in sample (n) divided by sum of weights
    (i.e. the mean of the weights) ;
  fw_normalized = fw0 * (3523/837829) ;
run ;

proc freq data = def_analysis ;
  table def * stabilized / or nocol norow ;
  weight fw_normalized ;
run ;
```

The SAS code of Figure 8.7.1 generates the output of Figure 8.7.2.

Figure 8.7.2: SAS Output for Calculating Confidence Interval of Odds Ratio with Normalized Weights

The FREQ Procedure

Table of Def by Stabilized

Def / Stabilized	0	1	Total
0	902.875	1994.57	2897.45
	25.63	56.62	82.24
1	111.159	514.393	625.551
	3.16	14.60	17.76
Total	1014.03	2508.97	3523
	28.78	71.22	100.00

Statistics for Table of Def by Stabilized

Odds Ratio and Relative Risks

Statistic	Value	95% Confidence Limits	
Odds Ratio	2.0947	1.6818	2.6091
Relative Risk (Column 1)	1.7536	1.4691	2.0933
Relative Risk (Column 2)	0.8371	0.8012	0.8747

Sample Size = 3522.9984489

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Example 8.8. Estimating a CI of an Odds Ratio with Final weights and SAS PROC FREQ

If we calculate the variance with the usual estimator of the variance of a log odds or Equation (8.3) and insert the weighted estimates from Table 8.3.1, we get

$$\hat{v}(\log odds) = \frac{1}{214{,}719} + \frac{1}{474{,}343} + \frac{1}{26{,}435} + \frac{1}{122{,}331} = 0.0000053$$

and a standard error of $\widehat{se}(\log odds) = 0.0073$.

This answer is about two orders of magnitude smaller than the weighted estimate using the replicate weights, and is misleading because it results in much smaller CIs.

Figure 8.8.1 provides the PROC FREQ code that will calculate the odds ratio using the replicate weights.

Figure 8.8.1: SAS Code for Calculating Confidence Interval of Odds Ratio with Final weights

```
proc freq data = def_analysis ;
  table def * stabilized / or nocol norow;
  weight fw0 ;
run ;
```

The SAS code of Figure 8.8.1 generates the output of Figure 8.8.2.

Figure 8.8.2: SAS Output for Calculating Confidence Interval of Odds Ratio with Final weights

The FREQ Procedure

Table of Def by Stabilized

Def / Stabilized Frequency Percent	0	1	Total
0	214719 25.63	474343 56.62	689062 82.24
1	26435.4 3.16	122331 14.60	148767 17.76
Total	241154 28.78	596674 71.22	837829 100.00

Odds Ratio and Relative Risks

Statistic	Value	95% Confidence Limits	
Odds Ratio	2.0947	2.0651	2.1248
Relative Risk (Column 1)	1.7536	1.7336	1.7739
Relative Risk (Column 2)	0.8371	0.8348	0.8395

Effective Sample Size = 837828.63086

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

Example 8.9. Estimating a CI of an Odds Ratio with No Weights and SAS PROC FREQ

The unweighted counts for the maintenance deficiencies example are provided in Table 8.9.1.

Table 8.9.1: Unweighted Counts for Maintenance Deficiencies Example

	Fewer than 3 maintenance deficiencies	3 or more maintenance deficiencies	Total
Rent Stabilized Post-1947	$n_{00} = 888$	$n_{01} = 2{,}007$	$n_{0+} = 2{,}895$
Rent Stabilized Pre-1947	$n_{10} = 111$	$n_{11} = 517$	$n_{1+} = 628$
Total	$n_{+0} = 999$	$n_{+1} = 2{,}524$	$n = 3{,}523$

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

If we completely ignore the sample weights, the odds of having 3 or more maintenance deficiencies given the HU is rent stabilized pre-1947 is calculated as

$$\{\text{Odds of Y given Rent Stabilized Pre} - 1947\ (X = 1)\} = \frac{517}{2{,}007} = 0.2576$$

and we estimate the odds ratio as

$$\hat{\theta} = \frac{888 * 517}{2{,}007 * 111} = 2.0608$$

The log odds or the natural log of the odds ratio $\hat{\theta}$ is log odds$= \ln(\hat{\theta}) = 0.7231$. The unweighted variance of the log odds can be calculated using Equation (8.3) and Table 8.8.1 as

$$\hat{v}(\log odds) = \frac{1}{888} + \frac{1}{2{,}007} + \frac{1}{111} + \frac{1}{517} = 0.0126$$

and the standard error is $\widehat{se}(\log odds) = 0.1121$. With the estimate of $\hat{\theta}$ and the standard error, we can use Equation (7.4) to calculate the CI of (1.6543, 2.5672)

The SAS code of Figure 8.9.1 shows how the CIs for the odds ratio can be calculated directly.

Figure 8.9.1: SAS Code for Calculating Confidence Interval of Odds Ratio without Weights

```
proc freq data=def_analysis ;
    table def * stabilized / or nocol norow ;
run ;
```

The SAS code of Figure 8.9.1 generates the output of Figure 8.9.2.

Figure 8.9.2: SAS Output for Calculating Confidence Interval of Odds Ratio without Weights

The FREQ Procedure

Table of Def by Stabilized

Def Stabilized

Frequency Percent	0	1	Total
0	888 25.21	2007 56.97	2895 82.17
1	111 3.15	517 14.67	628 17.83
Total	999 28.36	2524 71.64	3523 100.00

Statistics for Table of Def by Stabilized

Odds Ratio and Relative Risks

Statistic	Value	95% Confidence Limits	
Odds Ratio	2.0608	1.6543	2.5672
Relative Risk (Column 1)	1.7354	1.4532	2.0724
Relative Risk (Column 2)	0.8421	0.8062	0.8796

Effective Sample Size = 3523

Source: U.S. Census Bureau, 2017 New York City Housing and Vacancy Survey.

9. How the Replicate Weights Are Calculated

As with most large-scale household surveys, in an effort to control costs, the NYCHVS uses a complex sample design involving multi-stage sampling, and unequal sampling rates. Weights are needed in the analysis to compensate for unequal sampling rates as well as for nonresponse. Further, most estimates from complex samples are non-linear statistics, so estimates of the standard errors are often obtained using the first-order Taylor series approximations or replication methods such as balanced repeated replication or jackknife replication. Therefore, the complex sample design needs to be taken into account in estimating the precision of survey estimates. Not accounting for these sample design features will lead to inaccurate point estimates and an underestimation of the precision. This section describes methods used for generating point estimates and variance estimation.

Variance Estimates with Replication

Replication methods are able to provide estimates of variance for a wide variety of designs using probability sampling, even when complex estimation procedures are used. This method requires that the sample selection, the collection of data, and the estimation procedures be carried out (replicated) several times. The dispersion of the resulting estimates can be used to measure the variance of the sample.

In the variance estimation of NYCHVS, we use Successive Differences Replication (SDR). This technique is embodied by the replicate factors that are produced for the NYCHVS replicate variance estimator.

Replicate Weights

The unbiased weights (baseweight × special weighting factors) are multiplied by the replicate factors to produce the replicate weights. The replicate weights are further adjusted with the same weighting adjustments as applied to the final weight including a noninterview adjustment, the ratio adjustment for the 2010 Census frame totals, the ratio adjustment for the in remediation frame totals, and the ratio adjustments for housing unit totals produced by demographic analysis. For more details on the sample design of NYCHVS, please refer to the 2017 NYCHVS Sample Design, Weighting, and Error Estimation document (U.S. Census Bureau, 2018). By applying the other weighting adjustments to each replicate, the final replicate weights reflect the impact of the weighting adjustments on the variance.

Replicate Factors

The theoretical basis for the successive difference method was discussed by Wolter (2007) and extended by Fay and Train (1995) to produce the SDR method. See also Ash (2014) and Opsomer, Breidt, White, and Li (2016). The following is a description of SDR.

To apply SDR to the sample, we sort the sample by borough (variable on the PUFs: BORO) and then within borough by the same order that was used to select the original *systematic* sample. Then each sample unit is assigned two rows of the given Hadamard matrix. For example, the assignment for a Hadamard of order 80 would be rows (1,2) assigned to the first unit, rows (2,3) assigned to the second unit, … rows (80,1) assigned to the 80th unit. The assignment is repeated in further cycles until the entire sample is assigned two rows.

For a sample, two rows of the Hadamard matrix are assigned to each pair of units creating replicate factors, $f_{i,r}$ for $r = 1,..., R$ as

$$f_{i,r} = 1 + 2^{-3/2} h_{i+1,r} - 2^{-3/2} h_{i+2,r}$$

where

i	the index on the units of the sample
r	the index on the set of replicates
$h_{i,r}$	number in the Hadamard matrix (+1 or −1) for the ith unit in the systematic sample
R	the number of total replicate samples or simply replicates

This formula yields replicate factors of approximately 1.7, 1.0, or 0.3.

Example 9.1. Successive Difference Replication

The following is a simple example showing the SDR method. The sample contains $n = 4$ units and their weights are shown in Table 9.1.1.

Table 9.1.1: Sample Weights

Sample HU	Sample Weight
1	15.00
2	23.00
3	19.00
4	16.00

We choose to use the following 4 × 4 Hadamard matrix to define the replicate factors.

$$\mathbf{H}_4 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

Two consecutive rows of $\mathbf{H}_4$ are assigned to each sample unit as denoted in Table 9.1.2.

Table 9.1.2: Assignment of Rows of the Hadamard Matrix

Sample HU	Sample Weight	Row I	Row II
1	15.00	1	2
2	23.00	2	3
3	19.00	3	4
4	16.00	4	1

Plugging these values into our replicate factor formula of Equation (2.1) we get

Sample HU 1

$$f_{1,1} = 1 + 2^{-\frac{3}{2}} h_{1,1} - 2^{-\frac{3}{2}} h_{2,1} = 1 + 2^{-\frac{3}{2}}(+1) - 2^{-\frac{3}{2}}(+1) = 1.0$$

$$f_{1,2} = 1 + 2^{-\frac{3}{2}} h_{1,2} - 2^{-\frac{3}{2}} h_{2,2} = 1 + 2^{-\frac{3}{2}}(+1) - 2^{-\frac{3}{2}}(-1) = 1 + \frac{1}{\sqrt{2}} \cong 1.7$$

$$f_{1,3} = 1 + 2^{-\frac{3}{2}} h_{1,3} - 2^{-\frac{3}{2}} h_{2,3} = 1 + 2^{-\frac{3}{2}}(+1) - 2^{-\frac{3}{2}}(+1) = 1.0$$

$$f_{1,4} = 1 + 2^{-\frac{3}{2}} h_{1,4} - 2^{-\frac{3}{2}} h_{2,4} = 1 + 2^{-\frac{3}{2}}(+1) - 2^{-\frac{3}{2}}(-1) = 1 + \frac{1}{\sqrt{2}} \cong 1.7$$

Sample HU 2

$$f_{3,1} = 1 + 2^{-\frac{3}{2}} h_{2,1} - 2^{-\frac{3}{2}} h_{3,1} = 1 + 2^{-\frac{3}{2}}(+1) - 2^{-\frac{3}{2}}(+1) = 1.0$$

$$f_{3,2} = 1 + 2^{-\frac{3}{2}} h_{2,2} - 2^{-\frac{3}{2}} h_{3,2} = 1 + 2^{-\frac{3}{2}}(-1) - 2^{-\frac{3}{2}}(+1) = 1 - \frac{1}{\sqrt{2}} \cong 0.3$$

$$f_{3,3} = 1 + 2^{-\frac{3}{2}} h_{2,3} - 2^{-\frac{3}{2}} h_{3,3} = 1 + 2^{-\frac{3}{2}}(+1) - 2^{-\frac{3}{2}}(-1) = 1 + \frac{1}{\sqrt{2}} \cong 1.7$$

$$f_{3,4} = 1 + 2^{-\frac{3}{2}} h_{2,4} - 2^{-\frac{3}{2}} h_{3,4} = 1 + 2^{-\frac{3}{2}}(-1) - 2^{-\frac{3}{2}}(-1) = 1.0$$

Sample HU 3

$$f_{2,1} = 1 + 2^{-\frac{3}{2}} h_{3,1} - 2^{-\frac{3}{2}} h_{4,1} = 1 + 2^{-\frac{3}{2}}(+1) - 2^{-\frac{3}{2}}(+1) = 1.0$$

$$f_{2,2} = 1 + 2^{-\frac{3}{2}} h_{3,2} - 2^{-\frac{3}{2}} h_{4,2} = 1 + 2^{-\frac{3}{2}}(+1) - 2^{-\frac{3}{2}}(-1) = 1 + \frac{1}{\sqrt{2}} \cong 1.7$$

$$f_{2,3} = 1 + 2^{-\frac{3}{2}} h_{3,3} - 2^{-\frac{3}{2}} h_{4,3} = 1 + 2^{-\frac{3}{2}}(-1) - 2^{-\frac{3}{2}}(-1) = 1.0$$

$$f_{2,4} = 1 + 2^{-\frac{3}{2}} h_{3,4} - 2^{-\frac{3}{2}} h_{4,4} = 1 + 2^{-\frac{3}{2}}(-1) - 2^{-\frac{3}{2}}(+1) = 1 - \frac{1}{\sqrt{2}} \cong 0.3$$

Repeat the process for Sample HU #4 using rows 4 and 1, respectively.

If we calculate the replicate factors for every replicate and unit in the sample, we get the values in Table 9.1.3.

Table 9.1.3: Replicate Factors

Sample HU	Replicate Factors			
	Replicate 1	Replicate 2	Replicate 3	Replicate 4
1	1.0	1.7	1.0	1.7
2	1.0	0.3	1.7	1.0
3	1.0	1.7	1.0	0.3
4	1.0	0.3	0.3	1.0

Next, multiply the sample and corresponding factors to get the replicate weights of Table 9.1.4.

Table 9.1.4: Final Replicate Weights

Sample HU	Full	Replicate Weights			
	Sample Weight	Replicate 1	Replicate 2	Replicate 3	Replicate 4
1	15.0	15.0	25.5	15.0	25.5
2	23.0	23.0	6.9	39.1	23.0
3	19.0	19.0	32.3	19	5.7
4	16.0	16.0	4.8	4.8	16.0

Other Weighting Adjustments for Replicate Weights

In Example 9.1, we end at the step of adjusting the replicate base weights for the different replicates. The next step is to calculate the rest of the weighting adjustments for each set of replicate weights. The replicate weights also account for the effect on the variance of the other weighting factors. Recalculating the noninterview and ratio adjustments for each replicate ensures that the randomness injected or mitigated by the different weighting adjustments is represented in each of the replicate estimates. See also Judkins (1990; p. 224) and Brick and Kalton (1996) for additional discussion of application of other weighting adjustments within replicate weighting.

The Factor of 4 in Equation (2.1)

The replication variance estimator of Equation (2.1) includes what looks like an odd term of 4. This 4 is required by SDR because as Ash (2014) explains, each of the four rows of the Hadamard matrix produce one estimator of the successive difference variance estimator. So with NYCHVS, we need to divide the usual replication estimator by 80/4 = 20 and not 80.

The easiest solution to adding the 4 to the PROC SURVEY procedures of SAS is to use the varmethod=brr(fay) option which has a default value of 0.5 for the perturbing factor. SAS will then use the 4 in the calculation of all replicates and variance estimates will be correct.

The alternative is to use to the PROC SURVEY procedures of SAS is to use the varmethod=brr option and multiply the standard error by 2 or multiply the variance by 4.

10. References

Ash, S. (2014). "Using Successive Difference Replication for Estimating Variances," Survey Methodology, 40, 1, 47-59.

Brick, J.M. and Kalton, G. (1996). "Handling Missing Data in Survey Research," Statistical Methods in Medical Research, 5, 215-238.

Cochran, W.G. (1977). *Sampling Techniques*, 3rd Edition, John Wiley & Sons, New York.

Eltinge, J.L. and Jang, D.S. (1996). "Stability Measures for Variance Commponent Estimators under a Stratified Multistage Design". Survey Methodology, 22, 157-165.

Fay, R.E. and Train, G.F. (1995). "Aspects of Survey and Model-Based Postcensal Estimation of Income and Poverty Characteristics for States and Counties," Joint Statistical Meetings, Proceedings of the Section on Government Statistics, 154-159.

Heeringa, S.G., West, B. T., and Berglund, P.A. (2010). *Applied Survey Data Analysis*, CRC Press.

Huang, E.T. and Bell, W.R., (2011). "A Simulation Study of the Distribution of Fay's Successive Difference Replication Variance Estimator", retrieved February 11, 2019 from https://www.census.gov/content/dam/Census/library/working-papers/2011/demo/huang-bell2011.pdf

Judkins, D.R. (1990). "Fay's Method for Variance Estimation," Journal of Official Statistics, 6, 3, 223-239.

Kolenikov, S. (2019). Programs and Tutorials Written by Stas Kolenikov, retrieved March 3, 2019 from http://staskolenikov.net/stata

Korn, E.L. and Graubard, B.I. (1999). *Analysis of Health Surveys*, John Wiley & Sons.

Kreuter, F. and Valliant, R. (2007). "A survey on survey statistics: What is done and can be done in Stata." *The Stata Journal*, 7(1), 1 – 21. Retrieved July 18, 2011 from http://www.stata-journal.com/article.html?article=st0118

Krewski, D. and Rao, J.N.K. (1981). *Inference from Stratified Samples: Properties of the Linerization, Jackknife and Balanced Repeated Replication Method*, Annals of Statistics, 9, 5, 1010-1019.

Lewis, T. H. (2017). "Complex Survey Data Analysis With SAS." CPC Press, Boca Raton, FL. Section 8.

Lohr, S. (1999). *Sampling Design and Analysis*, Brooks/Cole Publishing Company.

Mukhopadhyay, P.K., An, A.B., Tobias, R.D., and Watts, D.L. (2008). "Try, Try Again: Replication-Based Variance Estimation Methods for Survey Data Analysis in SAS® 9.2," Proceedings of the 2008 NESUG.

Opsomer, J.D., Breidt, F.J., White, M., and Li, Y. (2016). "Successive Difference Replication for Variance Estimation in Two-Phase Sampling," Journal of Survey Statistics and Methodology, 4, 43-70.

SAS 9.2 User's Guide (2019), Second Edition, SAS Institute Inc. retrieved from https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_surveyfreq_a0000000220.htm

Särndal, C.-E., Swensson, B. and Wretman, J. (1992). *Model Assisted Survey Sampling*, Springer-Verlag.

Schenker, N. and Gentleman, J.F. (2001). "On Judging the Significances of Differences by Examining the Overlap Between Confidence Intervals," The American Statistician, 55:3. August 2001.

Taylor, L.H. (2016). *Complex Survey Data Analysis with SAS*, CRC Press.

U.S. Census Bureau. (2014). American Community Survey Design and Methodology (January 2014).

U.S. Census Bureau. (2018). 2017 New York City Housing and Vacancy Survey Sample Design, Weighting, and Error Estimation.

Valliant, R., Rust, K. (2010). "Degree of Freedom Approximations and Rules-of-Thumb," Journal of Official Statistics, Vol. 26, No. 4, 2010, pp. 585-602.

Wolter, K.M. (2007). *Introduction to Variance Estimation*, Springer-Verlag, 2nd Edition.

GLOSSARY OF TERMS

This glossary provides the definitions of several terms used with the technical document. It includes both terms related to statistics and the sample design.

Balanced Repeated Replication (BRR) – A method of variance estimation that is often used with two-stage sample designs that select one or two PSUs per first-stage strata. The method is valuable because it can be applied to estimating the variance of linear and non-linear estimates. Also, the intermediate replicate weights can be provided to data users so that they can estimate variances themselves using simple expressions for the variance. The main ideas of replication are outlined by McCarthy (1966).

Bias – The formal definition of bias of an estimator $\hat{\theta}$ of some statistic θ is the expected value of the absolute value of the difference between the estimator and statistic and its expected value, i.e., $B(\hat{\theta}) = E|\hat{\theta} - \theta|$. Informally, the bias is a measure of how close the estimator is to the value it is estimating.

Borough – A borough is determined by borough code variable (BORO) on the NYCHVS data file(s). The code BORO has the values: 1 = Bronx; 2 = Brooklyn; 3 = Manhattan; 4 = Queens; 5 = Staten Island.

Calibration – As described by Deville and Särndal (1992), calibration is a technique that can be used to reduce the variance of an estimator. Sometimes it can also have the effect of improving the coverage of the estimator. Calibration uses a set of known totals: either an "imported totals" (Särndal and Lundström, 1999) or a set of variables that are known for all units in the universe. Calibration finds weights that are "close" to the original design weights so that the estimated known totals with the new weights are the same as the known total.

Coefficient of Variation (CV) – The square root of the variance of an estimate divided by the estimate, i.e., $\sqrt{v(\hat{\theta})}/\hat{\theta}$.

Coverage – A measure of how well a frame and sample design includes the universe of interest. Usually coverage is expressed as a proportion. For example, if a study has 75% coverage then 75% of the universe of interest was included by the frame and the sample design.

Domain of Interest or Domain – A specific subset of the universe.

Eligible / Ineligible – Refers to the whether a unit of interest is in the universe of interest or not in the universe of interest. See also AAPOR (2011).

Final Weight – The sample weight that can be used with the sample to make estimates of variables collected with the sample. The name "final weight" is used in some sections of the guide to distinguish the final weight from the replicate weights. The final weight is also the same as the weight for Replicate 0.

Frame – The list of units in the universe of interest.

Generalized Variance Function (GVF) – As explained by Wolter (2007), the GVF is a simple model that expresses the variance as a function of the expected value of the survey estimate.

Householder (Reference Person) – The householder (reference person) is the household member or one of the household members who owns or rents the sample unit. If no household member owns or rents the sample unit, the first person listed is designated as the householder (reference person). The term "reference person" is used in the questionnaire but is replaced by the term householder in the final data presentations.

Housing Unit – A housing unit is a house, an apartment, a group of rooms, or a single room occupied or intended for occupancy as separate living quarters.

Interviews – Interviews consists of units that are either:

- Occupied - Record Type (RECID) has a value of 1; or
- Vacant - Record Type (RECID) has a value of 3.

Interviews also have a value of 01 (Questionnaire complete) in Source Code 33 (Noninterview reason) of the NYCHVS questionnaire (Use SAS variable RECID=1, 3 on the NYCHVS data files).

Noninterview – Units are classified as noninterviews when we expect to get interviews but don't for one of the following reasons:

1. In-scope (Type A) - refused, no one home, temporarily absent, other.

2. Out-of-scope (Type C) - demolished, condemned, nonresidential, merged, damaged by fire, boarded up, list procedure applied, no such address, other.

Units that are noninterviews have a value within the range of 02 - 14 (questionnaire not complete) of Source Code 033 (Noninterview reason) on the NYCHVS questionnaire. Please refer to the 2017 NYCHVS Sample Design, Weighting, and Error Estimation document for more detailed descriptions of *Type A Noninterviews* and *Type C Noninterviews.*

Occupied Units – Sample units are classified as occupied units if they have people living in them at the time of observation, regardless of their condition. Units that are occupied have a value of 1 for RECID on the NYCHVS questionnaire. They are units that were interviewed.

Owner Occupied – An occupied housing unit is owner occupied if the owner or co-owner lives in the unit, even if it's not mortgaged or not fully paid for. See *Owner Occupied Units* for values of units that are owner occupied.

Owner Occupied Units – Units having a value of 1 for RECID and a value of 1 in Source Code 115 (SC115) on the NYCHVS questionnaire.

Relative Variance, Relvariance, or Relvar – is a measure of the relative dispersion of a probability distribution and is defined as the variance divided by the square of the estimate. It is also equal to the square of the coefficient of variation, i.e., $relvar(\hat{\theta}) = v(\hat{\theta})/\hat{\theta}^2$

Renter Occupied – An occupied housing unit is renter occupied if the unit is rented for cash rent or occupied without payment of cash rent. See *Renter Occupied Units* for values of renter occupied units.

Renter Occupied Units – Units having a value of 1 for RECID and a value of either 2 or 3 in Source Code 116 (SC116) on the NYCHVS questionnaire.

Replicate Sample or Replicate – In replication variance estimation, we make several replicate samples where each has a different set of weights. Each of the replicate samples and the corresponding replicate weights used to make a replicate estimate that is used in variance estimation.

Replicate Weight – The replicate weight that can be used with the sample to make estimates of variables collected with the sample.

Replicate 0 – Refers to the replicate that uses the final weight.

Sample Design – Everything about the selection of units into the sample that determines the probability of selection for each unit. We think of estimation as separate from sample design, in that some estimation procedures are more appropriate than others for a given sample design, but any estimator could be used with the sample derived from a given sample design.

Sampling Fraction – The fraction of the universe that is in the sample. With an equal probability sample design, the sampling fraction is the ratio of the sample size and the size of the universe, often represented as $f = n/N$.

Sampling Interval – The inverse of the sampling fraction. Sometimes referred to as the "take-every" because we take every f^{-1} units of the universe into the sample.

Standard Error – Is the square root of the variance, i.e., $se(\hat{\theta}) = \sqrt{v(\hat{\theta})}$.

Successive Difference Replication (SDR) – A replication variance estimation method that mimics the successive difference variance estimator and can be used to estimate the variance from a *sys* sample design. The main ideas of replication are outline by Fay and Train (1995).

Systematic Random Sampling or *sys* – A random sampling method that requires selecting samples based on a system of intervals in an ordered population.

Unit – The following definition is from Hájek (1981, p. 4):

"The units making up the population S may be any elements worth studying – persons, families, farms, account items, temperature readings, and so on – and their nature will be irrelevant for theoretical considerations. We shall assume that the units are identifiable by certain labels (tags, names, addresses) and that we have available a frame (list, map) showing how to reach any unit given its label."

For NYCHVS, unit of interest is the housing unit.

Universe of interest – In finite population sampling, the universe of interest, or simply the universe, is the well-defined set of units for which we would like to produce an estimate.

Vacant Housing Units – Vacant housing units include vacant for sale, vacant for rent, and vacant not for sale or rent units.

Variance or Sample Variance – Is a measure of the variability of an estimate. With finite population sampling, variance refers to the measure of how the estimate differs if we were to select other samples. Formally, the variance of an estimator $\hat{\theta}$ is the expected value of the squared difference between the estimator $\hat{\theta}$ and its expected value, i.e., $v(\hat{\theta}) = E\left(\hat{\theta} - E(\hat{\theta})\right)^2$.

1) Vacant for Sale: Vacant housing units that are intended by the owner for owner-occupancy, having:

   - a value other than 1 (not dilapidated) in Source Code 23 (Condition of Unit),
   - a value of 3 (vacant) for RECID (Occupancy Status), and
   - a value of 2 (available for sale) in Source Code 534 (Status of unit)

2) Vacant for Rent: Vacant housing units that are intended by the owner for renter-occupancy, having:

- a value other than 1 (not dilapidated) in Source Code 23 (Condition of Unit),
- a value of 3 (vacant) for RECID (Occupancy Status), and
- a value of 1 (available for rent) in Source Code 534 (Status of unit)

3) Vacant (Not for Sale or Rent): Sample units are classified as vacant if they were housing units and there was no one living in them at the time of observation. Units that are vacant have:

- a value of 1 (dilapidated) in Source Code 23 (Condition of Unit) and a value of 1 or 2 in Source Code 534 (Status of Unit),
- a value of 3 (vacant) for RECID (Occupancy Status), and
- a value of 3 (not available for rent or sale) in Source Code 534 (Status of unit)

References

The American Association for Public Opinion Research. 2011. *Standard Definitions: Final Dispositions of Case Codes and Outcome Rates for Surveys*, *7th edition*, AAPOR.

Deville, J.C. and Särndal, C.E. (1992). "Calibration Estimators in Surveys," Journal of the American Statistical Association, 87, 418.

Fay, R.E. and Train, G.F. (1995). "Aspects of Survey and Model-Based Postcensal Estimation of Income and Poverty Characteristics for States and Counties," Joint Statistical Meetings, Proceedings of the Section on Government Statistics, 154-159.

Hájek, J. (1981). *Sampling From a Finite Population*, Marcel Dekker Inc.

McCarthy, P.J. (1966). "Pseudo-replication: half-samples," Review of the International Statistical Institute, 37, 239-264.

Särndal, C. E. and Lundström, S. (1999). "Calibration as a Standard Method for Treatment of Nonresponse", Journal of Official Statistics. 305-327.

U.S. Census Bureau. (2018). 2017 New York City Housing and Vacancy Survey Sample Design, Weighting, and Error Estimation.

Wolter, K.M. (2007). *Introduction to Variance Estimation*, Springer-Verlag, 2nd Edition.