Implementing Whole Dwelling (Household) Imputation

Prepared by **Statistics Canada**

May 2023

This note is a follow-up to the PRASC note "On Adjusting for Dwelling Non-response in a Census" and it focuses on simple approaches to implement whole household imputation (WHI) as means of adjusting for unit non-response in a census as suggested in that note. Implementation approaches are suggested for R, Stata and SPSS.

Imputation is considered both for complete nonresponse and for partial response where basic demographics have been provided but nothing else (sometimes called last resort information). Such records are referred to here as recipients.

The strategy described in this document is actually one of whole **dwelling** imputation. Response or nonresponse is considered at the dwelling level regardless of the number of households in the dwelling. Imputation is done at the level of complete dwellings. So, from this point forward in this document the procedure will be called whole dwelling imputation (WDI).

The approach described here uses donor imputation and matches to find donors using as much information as possible from recipients such as:

- ED identification (always present)
- Dwelling type (usually available from dwelling listing)
- Basic demographics for usual residents (sometimes available for dwellings/households where only demographics were collected) such as:
    - age, sex for each person
    - functions of the above such as total number of persons, possibly by sex and/or age group

We recommend that original unedited and unimputed data as received from data collection be retained on a response database. A second database, called a processing database should be created where all editing, imputation and data cleaning is done.

Note that the resulting census database should carry a flag to indicate records whose data were imputed via the WDI process.

The note starts with an explanation of some pre-processing that is necessary.

Preprocessing

First, there are some initial data check-in and data cleaning steps to be done in concluding field operations. There may be other activities needed but the following ones are directly related to nonresponse adjustment requirements.

1. Enumerators and their supervisors need to ensure that every case is in a fit state to be uploaded to servers. Amongst other things this will include ensuring that each one has a dwelling type and a questionnaire completion status.
2. All tablets must by synced a final time to ensure all data have been uploaded. At headquarters, validate that all data have been received.
3. Conduct edits and reviews to identify and remove duplicate records. Ensure every questionnaire has complete and correct identification information. Ensure listings and enumerations are in correct EDs.
4. Review listing counts, ensuring there is a complete listing for every ED. Identify EDs with dwelling counts significantly different from that expected. Follow-up regarding such situations.

To do whole dwelling imputation, some key definitions are needed.

- *Eligible as a donor* – These are high quality responses that are reasonably complete with a minimum of edit failures/warnings. A balance will need to be struck between having donors of very high quality and having an adequate supply of donor records.
- *Sufficient response* – Responses that are considered acceptable but which are not clean enough to be used as donors. We recommend a minimal requirement. Any dwelling response that provides at least basic demographics for all persons as well as some minimal amount of additional response should be considered a sufficient response for this purpose.
- *Last resort response* – A dwelling which has provided, at most, basic demographics for each person. Dwellings providing as little as total number of usual residents should be included here. "Basic demographics" refers to age and sex. In some cases, this might also include variables like marital status as well as relation to head of household.
- *Non-response or insufficient response* - An occupied dwelling that was either completely non-responding (e.g. no contact, refusal) or did not provide enough information to meet the minimum requirement for a last resort response.
- *Out of scope for imputation* – dwellings which are not occupied, or do not meet the definition of having usual residents

The following is the information needed on the processing database in order to do the whole dwelling imputation:

- Essential
    - Unique identifier for the dwelling
    - Geography identifiers (i.e. full ED identification, including higher-level areas)
    - Dwelling occupancy status
    - Response status (full response, sufficient partial, last resort only, non-response, out of scope)
- If available and applicable

- Dwelling type (perhaps from listing)
- Number of households (if applicable)
- Last resort information (number of persons, demographic information for persons in the dwelling) (if applicable)

The first step of the preprocessing is to take all dwellings on the census database and classify as follows. Call this dwell_status:

1. Occupied responses – eligible as a donor
2. Occupied sufficient responses – not eligible as a donor
3. Occupied non-responses (imputation recipients)
4. Occupied last resort responses (also imputation recipients)
5. Everything else (e.g. vacant, not a dwelling, dwelling with zero usual residents)

The objective is to do whole dwelling imputation for groups 3 and 4 above.

Dwellings in categories 2 and 5 will not participate in WDI in any way.

While census databases may be structured as relational databases with separate records or tables for buildings, dwellings, households, and individuals this discussion of whole dwelling imputation considers a file extracted from such a database consisting of records at the dwelling level where the dwelling record comprises all the data for the dwelling, including all households and all individuals in the dwelling.

In the strategy described here we will ignore the number of households in the dwelling. In the case of the complete nonresponse (category 3), the actual number of households in the recipient dwelling is unknown and so this strategy does not change anything. In the case of the last resort responses (category 4), the number of households will be known. The donor record selected for any given recipient may have a differing number of households but, depending on the success of finding a good donor, is likely to have the same dwelling size (i.e. total number of persons). Note that since the very large majority of dwellings have exactly one household, this recipient-donor mismatch of household counts will be very infrequent. If the selected donor record does have a different number of households, the recipient information should be discarded and replaced with the donor information as part of the imputation result.

Extract from the census database a file including only records with dwell_status equal to 1, 3 or 4. Define the following variables on the file(s) to be used as input for the WDI procedure:

- ED_id: the unique identifier of the enumeration district
- Unique_dwgID: an overall unique dwelling identifier
- Super_id: id of supervisor area or some next level up aggregation of EDs
- Dwell_type_d
- Dwell_size_d
- Dwell_size_grp
- Dwell_status
- Age, sex values for each usual resident

Dwell_type_d is a derived categorical variable based on the dwelling type as recorded (usually) during the listing process. Note that for the purposes here, it is assumed that dwelling type is non-missing for all dwellings. If that is not the case, the value should be imputed during the pre-processing (using either hotdeck or random imputation based on the frequency distribution amongst responses in a relevant imputation domain). A simple classification such as the following should be sufficient:

- Single detached
- Single attached
- Apartment
- Other

Dwell_size_d is a derived categorical variable based on the total number of usual residents in the dwelling, coded as:

- 1
- 2
- 3
- 4
- 5 meaning 5 or more usual residents
- missing missing count of usual residents

Dwell_size_grp – broader categories of dwelling size. For example

- 1, 2 or 3
- 4 or more
- Missing

The above top coding for dwelling is of course just an example value. This can be altered to suit national circumstances and dwelling size distributions.

A final step in preprocessing that is important for designing a good whole dwelling imputation strategy is some exploratory analysis of the census database. Such analysis should focus on topics like:

- Response rates at low levels of aggregation such as EDs
- Identification of problem areas where high nonresponse and thus a low supply of donors will make good imputation more difficult. This could take the form of a low supply of donors in general or a low supply of donors for certain categories of imputation group; say by dwelling type or dwelling size.
- Consideration of the statistical distribution for variables under consideration for use in matching to donors (i.e. the imputation groups), and thus assessment of their use and possible need for collapsing such as proposed above.

Special Circumstances alternate procedure

In cases where census data collection has not gone well, there may be evidence of response bias where the dwelling/household size distribution amongst respondents is clearly materially different from that

expected. It is vital to evaluate such situations with care. Such changes could be real. But if evidence of response bias is strong, then an adjustment to the imputation for nonresponse may be able to help.

In such a case, for nonresponses where the dwelling size (total of usual residents) is unknown one could impute the dwelling sizes so that the overall census distribution of dwelling sizes is closer to that expected. This imputation would be done prior to running the whole dwelling imputation procedure described above.

Strategy for Complete Nonrespondents (dwell_status = 3)

In this situation we know very little about the recipient records – only the geographic identifiers and the dwelling type. We search for a donor record (i.e. from the group with dwell_status = 1) from the same ED and then if there is no suitable donor or an inadequate supply of donors search within a set of EDs, such as the super_id noted above.

In some censuses where dwelling listing is done shortly before or as an integral part of enumeration, some reliable basic information such as total usual residents, perhaps also by sex and age group, may be available from the listing procedures. If so, these variables may also be used in the donor matching procedure.

In the approaches described below we will define an imputation group by the combination of the variables (Super_id, ED_id, Dwell_size_grp, Dwell_size_d, dwell_type_d). Note again that unless available from listing, Dwell_size_d and Dwell_size_grp will be missing for the complete nonrespondents.

A donor imputation methodology is used with Super_id, ED_id, dwell_size_grp, dwell_size_d and dwell_type_d as matching variables. The imputation will search for a donor which matches the recipient on the non-missing values of as many as possible of these variables.

The list of Super_id, ED_id, dwell_size_grp, dwell_size_d and dwell_type_d is in what we view as most important to least important matching variable. The following explanation is provided on the basis of that. However for example, it might be seen that within a Super_id, matching on dwelling size is more important than matching on ED_id in which case the order Super_id, dwell_size_grp, dwell_size_d, ED_id, dwell_type_d would be reasonable.

After the above preprocessing, for each dwelling with dwell_status = 3 we want to find a donor (with dwell_status = 1) that matches as best as possible with the above variables. During the actual imputation step (or call of an imputation routine) the only thing imputed will be the unique identifier (dwell_id) of the matched donor; this will be specified in the call to the donor imputation procedure.

Afterwards in a simple matching step, the data from the donor will need to be copied to the recipient. The entire donor dwelling record is copied to the recipient on the processing database. This includes all

dwelling, household and individual variables. The unique identifier of the donor record should also be retained.

In a good donor imputation procedure no donor record should be used "excessively". In the description of the implementation in Stata we describe a procedure to control this. The SPSS procedures we found appear capable of controlling re-use of donor records. However, the procedure we found in R is very automated and there does not appear to be any easy way to control any re-use.

The inclusion of both dwell_size_d and dwell_size_grp may appear redundant but they are both included to facilitate collapsing of dwelling sizes when needed without entirely discarding the dwelling size information.

Super_id is similarly included to facilitate collapsing of EDs should that be necessary.

Strategy for Last Resort Responses (dwell_status = 4)

For this group we have more information for recipients. In particular, we know the total number of usual residents (dwell_size_d) and often also have basic demographic data such as age and sex for some or all usual residents. In some cases, additional demographic information such as relationship to head of household, marital status may also be available. For the purpose of explanation in this note, we assume this is not the case.

For whole dwelling imputation, it would be very difficult to directly use the age, sex variables for each of the usual residents as matching variables. The main difficulty is that it would frequently be difficult to impossible to find a donor record matching exactly on all these variables. And so we must find a good way to consolidate this information into something more useable.

On the basis of our knowledge of typical patterns of census nonresponse and coverage error we propose creation of a derived categorical variable called dwell_comp (dwelling composition). It will have a number of categories that are meaningful for this purpose but not so many as to be likely to create difficulties for donor imputation. Each category is briefly explained. Key factors that matter for this purpose include:

- Males are usually missed more often than females
- Presence of children is closely associated with both coverage and response likelihood. As well, dwellings with children will differ in characteristics from those without
- Single parent dwellings are less likely to respond that those with two
- Number of children matters in terms of obtaining a good matching donor

For this explanation we call persons less than 16 years old children and older persons adults. The subset of adults aged 16-24 are called young adults.

An example for dwell_comp of such a categorization is the following:

1. One female
2. One male

3. One male, one female with an age difference of <= 10 years (i.e. most married couples without offspring)
4. One adult with one child or young adult where the older person is at least 12 years older than the younger (i.e. single parent with one offspring)
5. One adult with two or more children or young adults where the oldest person is at least 12 years older than the next oldest (single parent with two or more offspring)
6. One adult male with one adult female with one child or young adult at least 12 years younger than the female adult. (nuclear family with one offspring)
7. One adult male with one adult female with two or more children or young adults where the oldest of these is at least 12 years younger than the adult female (nuclear family with two or more offspring)
8. All other dwellings with no children
9. All other dwellings with children
10. Missing (meaning age and/or sex missing for at least one member of the dwelling)

A simple collapsed or grouped version of this variable could be a binomial one, Dwell_comp_grp, with the categories: (1) dwellings with no children, (2) dwellings with children.

In cases where a dwell_comp category cannot be definitively determined then dwell_comp should be assigned a missing value. Similarly for where dwell_comp_grp cannot be definitively determined then dwell_comp_grp should be assigned a missing value.

This variable is then used as an additional one in the list of matching variables. A donor imputation methodology is used with Super_id, ED_id, dwell_size_grp, dwell_size_d, dwell_comp_grp, dwell_comp and dwell_type_d as matching variables. The imputation will search for a donor which matches the recipient on the non-missing values of as many as possible of these variables.

Please note that the above example description is just that – an example. Other sets of matching variables could be very reasonable. For example, the availability of marital status or relationship to head could contribute usefully to effective matching. A different threshold between children and adult might be chosen. It might be reasonable to disaggregate dwelling size to dwelling size by sex.

Software Choices

Possibly suitable software includes:

- CANCEIS – a very powerful edit and imputation system from StatCan. Much more powerful than what is needed for this purpose.
- CS Pro – documentation says it can do donor imputation. We have not researched this.
- R – the VIM package includes routines for donor imputation, and includes the matchImpute procedure. Although we have not tested it, it specifically uses donor imputation, which is what is needed to implement WDI. Documentation for matchimpute is available at rdocumentation.org/VIM/versions/6.2.2/topics/matchimpute (April 5, 2023)

- Stata – Hotdeckvar will do donor imputation, within imputation groups, where donors are selected with replacement. The donor imputation strategy is implemented by working through the levels of imputation groups iteratively calling the routine at each level. Hotdeckvar is not part of the base Stata; it must be installed by running the command "install ssc hotdeckvar".
- SPSS– Kirill's SPSS Macros Page (spsstools.net/en/macros/KO-spssmacros/) provides links for information on a wide variety of macros. One of these, "Impute missing data" will download a paper which provides details on two macros for hot deck imputation. We have not tested these but both appear to be suitable for this application. Alternatively, a Google search on "SPSS hot deck macro" will provide useful links.

No matter which software you're using, it's worthwhile to do some internet searches since new programs are added all the time and existing ones updated.

Matchimpute in R

Suitable donors are searched based on matching of categorical variables. A list of matching variables is specified in priority order (e.g. Super_id, ED_id, dwell_size_grp, dwell_size_d and dwell_type_d). For each recipient record the routine first attempts to find a donor by matching on all of these variables. If no match is found, variables are dropped in reverse order until a match can be found.

The Matchimpute procedure does not appear to control the frequency of donor re-use. Once imputation has been run, it is thus particularly important to evaluate donor re-use. Donor re-use up to about five times is acceptable. Significant donor re-use beyond that frequency can indicate that for the most affected imputation groups, the least important matching variable(s) should be dropped.

There may be other routines available in R that will do a good job of the donor imputation. Knowledgeable R users could search for and consider the suitability of such other routines.

Hotdeckvar in Stata

This algorithm allows users to specify one or more variables to be imputed, and also allows the use of "by" groups which would contain the list of imputation group variables. Within each combination of values for the imputation groups, the number of donors and number of recipients is determined (based on number without/with missing values in the variable(s) to be imputed. For the purposes here, the variable to be imputed is the donor ID. The hotdeckvar algorithm does not automatically iterate through the priority order of the imputation group variables, so instead this needs to be programed within a "forvalue" (do) loop. There is no automatic handing of the number of times a donor is used, but since the iteration is not automatic, some checks can be done after each level of iteration and donors can be removed from the pool once they've been used a maximum number of times. In addition, combinations of the imputation group variables can be excluded from an iteration if the number of available donors relative to the number of recipients is "too small"; this must be implemented within the do-loop for the iteration, as it is not part of the hotdeckvar algorithm itself. An example of the implementation of this algorithm is included in the ***Appendix***.

Diagnostics

A first evaluation to do is to verify that all recipients were assigned a donor. If not, it probably means that some imputation groups need to be collapsed and/or any restrictions on number of times a donor is used may need to be loosened.

Typical diagnostics should be prepared reporting on quantity and rates of non-response (and hence WDI) by:

- Various levels of geography
- Dwelling type
- Dwelling size

Non-response rates at low levels of aggregation such as EDs and supervisory districts will also be useful. These should have been tracked during operations but the final results (using the definitions used here) will be valuable. They will provide information on the performance of data collection staff and the challenges they faced. This information can also be valuable to indicate situations where there may be data quality issues related to high rates of non-response.

Whether R, Stata or something else is used, the extent of re-use of donor records should also be examined. This can be done by running frequency counts by donor ID and then preparing frequency distributions by geography, dwelling size and dwelling type. Review of these diagnostics may indicate imputation groups with some “excessive” re-use of donor records. If there is only a small amount of this, it is likely not a big concern. But if there are multiple imputation groups with such cases, it may be worth considering to rerun imputation for these groups using a reduced set of matching variables.

Appendix

The example below shows an implementation of the hotdeckvar algorithm in Stata where the following information is available on the data file:

- ED_id: the unique identifier of the enumeration district
- unique_dwgID: an overall unique dwelling identifier
- super_id: id of supervisor area or some next level up aggregation of Eds
- dwell_size_d
- dwell_size_grp
- dwell_status

In this example, type of dwelling was missing for a small percentage of dwellings, and therefore this information was imputed prior to running the rest of the program. After imputation, the grouped variable dwell_type_d was created.

Stata code:

```
clear
use "$cleanup1\clean_occupied_dwellings"
/* first impute dwelling type using hotdeckvar algorithm, using donors within
the same ED */
/* the "typedwel" variable is the one to be imputed; the hotdeckvar algorithm
copies the value into a new variable called typedwel_i; if the original value
of typedwel is missing, then the imputed value from the donor is replace in
the typedwel_i variable. */
by ED_id: hotdeckvar typedwel              /* the result is stored in variable
called typedwel_i */
gen dwell_type_d=typedwel_i
replace dwell_type_d=4 if typedwel_i>4


/* set up donor and recipients for WDI */
/* define donors, recipients and those to ignore based on the response and
completion status of the dwelling */
gen donor= pers_resp==1  /* fully completed questionnaires */
gen ignore= inlist(pers_resp,2,3,4,5)  /* partially completed questionnaires,
but insufficient to use as donor */
gen recipient= pers_resp>5  /* completely non-responding dwellings, some of
which may have number of members */

gen dwell_size_d=nmems   /* NOTE: if missing, then nmems=.  */
replace dwell_size_d=12 if dwell_size_d>12 & dwell_size_d<.
gen dwell_size_grp=.
replace dwell_size_grp=1 if inlist(dwell_size_d,1,2,3)
replace dwell_size_grp=2 if inrange(dwell_size_d,4,12)

/* IMPORTANT:  The hotdeckvar algorithm requires the imputation variables to
be of type numeric.  If the unique identifier for dwellings is of type
string, then a numeric variable needs to be created for temporary use in the
imputation.  This is done as follows:  */
```

```
gen dwell_no=_n     /* assigns a sequential number to each observation on the
dwelling file */

keep dwell_no donor ignore recipient ED_id hhsize super_id unique_dwgID
dwel_type_d dwell_size_d dwell_size_grp   /* NOTE:  In this example
“unique_dwgID” is the original dwelling identifier of string type; it must be
retained on the file in order to be able to link back to pick up the donor
information */
save occupied, replace

/* Create the “imputation base”, the file that will be used in the imputation
procedure */
clear
use occupied
drop if ignore==1 /* keep only good donors and recipients */
sort super_id ED_id dwell_no
/* create a variable called donor_id that is set to the dwell_no for donors
and set to missing for recipients; this is the variable that will be
specified for imputation */
gen donor_id=dwell_no
replace donor_id=. if recipient==1

/* create some variables to be used during the iterations of imputation to
keep track of which level of imputation yielded the donor, and how many times
the donor was used */
gen imputed=0
gen ntimes_used=0
save impute_base, replace

/* define the set of variables to be used at each level of iteration */
global list_1 "super_id ED_id dwell_type_d dwell_size_d"
global list_2 " super_id ED_id dwell_type_d"
global list_3 " super_id ED_id"
global list_4 "super_id dwell_type_d dwell_size_grp "
global list_5 " super_id dwell_type_d"
global list_6 "super_id"

/* loop through the defined iteration levels */
/*  see note at end  for more information on restricting number of times a
donor/group is used */

forvalues i=1/6 {
      clear
      use impute_base
      set seed 5844           /* setting a seed allows you to reproduce the
results */
      sort ${list_`i'}, stable   /* sort by the variables in the list for the
imputation group */
      /*  check for availability of donors and number of times a donor has
already been used */
      drop if donor==1 & ntimes_used>=3
      bysort ${list_`i'}: egen ndonors=sum(donor)  /* this calculates the
number of donors in each group */
```

```
      bysort ${list_`i'}: egen nrecipients=sum(recipient) if imputed==0  /*
this calculates the number of recipients which have not yet had a donor
imputed, in each group */
      keep if ndonors>0 & nrecipients>0 & ndonors>nrecipients & imputed==0 /*
keep those donors and recipients not yet imputed only if the group has more
donors than recipients needing a donor*/

      /*check the number of remaining records and run the iteration only if
records are available*/
      describe
      if r(N)>0 {
      di "running imputation for level `i'"
      by ${list_`i'}: hotdeckvar donor_id
      keep if donor_id==. & donor_id_i!=. /*keep recipients that found a
donor*/
      keep dwell_no donor_id_i
      rename donor_id_i donor_id
      save impute_result, replace

      /* match the imputation result back to the base file and update the
donor_id for those recipients that received a donor in the current iteration
*/
      clear
      use impute_base
      drop ntimes_used
      merge 1:1 dwell_no using impute_result, update
      replace imputed=`i' if _merge==4  /* assign the iteration number to the
imputed variable */
      drop _merge
      bysort donor_id: gen ntimes_used=_N  /* calculate the number of times
each donor has been used */
      save impute_base, replace
      }
      else {
      di "no observations for level `i'"
      }
}  /* end of iterations */

/* get some info to match back to the base file */
clear
use impute_base
keep if imputed>0  /* use just the recipients that had a donor imputed */
keep dwell_no donor_id imputed
bysort donor_id: gen ntimes_used=_N  /* get number of times each donor was
used */
save imputed_ids, replace
/* keep one observation per donor with the count of number of times used */
bysort donor_id: keep if _n==_N
keep donor_id ntimes_used
rename donor_id dwell_no  /* the unique id of the dwelling for the donors */
save donors_used, replace
```

```
/* this step places the donor's original unique dwelling ID (instead of the
sequential number) onto the recipient records; also adds the imputed variable
(level of iteration that generated the donor) and the number of times the
same donor was used */
clear
use impute_base
keep if donor==1
keep donor_id unique_dwgID
merge 1:m donor_id using imputed_ids
keep if _merge==3
keep dwell_no unique_dwgID imputed ntimes_used
rename unique_dwgID donor_unique_dwgID
rename imputed imputed_level
rename ntimes_used donor_used_ntimes
save imputed_ids, replace

/* match the imputation results back to the original file of all occupied
dwellings; create a variable called dwg_imputed=1 if a donor was assigned or
=0 otherwise */
clear
use occupied
merge 1:1 dwell_no using imputed_ids
gen dwg_imputed= _merge==3
drop _merge
/* then match the donor information to add the count of the number of times
the donor record was used; also create a variable called donor_used to
indicate (for each donor record) if the record was used as a donor or not */
merge 1:1 dwell_no using donors_used
gen donor_used= _merge==3
drop _merge
replace ntimes_used=0 if donor==1 & ntimes_used==.   /* ntimes_used=0 if the
observation is a donor that was not used; =1 if the observation was a donor
that was used; =. if the observation was not a donor */
save occupied_with_donorids, replace

tab2 ntimes_used donor, missing  /* show frequency counts on # times donors
were used */
tab2 imputed recipient, missing  /* show frequency counts on which level of
iteration resulted in assigning a donor to recipients */
```

Note on restriction of donor use:

This particular example contains a check for the number of times a donor is used. The check can only be performed after an iteration is finished and cannot restrict the number of times a donor is used within a given iteration since the selection of donors is done with replacement. At the end of an iteration, the number of times each donor has been used is calculated. If that number is greater than or equal to the designated amount (in this example, the number is 3), then that donor is dropped from the file going into the next iteration level.

An additional check is done to calculate the number of available donors and the number of remaining recipients (not yet assigned a donor) within each imputation group. In this example, if the number of available donors is not greater than the number of remaining recipients, then all the observations in that imputation group are dropped from the current level of iteration. What this means in practice is that the observations that are dropped will go to the next iteration level.