TECHNICAL PAPER

NCVER

# ANZSCO Imputation in the National Apprentice and Trainee Collection

AUTHOR: BRIAN HARVEY

NATIONAL CENTRE FOR VOCATIONAL EDUCATION RESEARCH

This technical paper is a description of the methodology used to impute values for records in the National Apprentice and Trainee database that have missing ANZSCO codes.

# Contents

# Introduction

## Missing occupation codes

Data relating to occupations has been collected in the national apprentice and trainee collection since 1994. The coding used conforms to classifications endorsed by the Australian Bureau of Statistics (ABS). The latest version issued from the ABS is the Australian and New Zealand Standard Classification of Occupations (ANZSCO). The classification previous to ANZSCO is the second edition of the Australian Standard Classification of Occupations (ASCO). Currently occupation data is collected using both ANZSCO and ASCO codes.

Although ANZSCO was only introduced to the Apprentice and Trainee Collection in 2007, states and territories were asked to back-code ANZSCO on all contracts that were active as at 1 July 2000. Therefore while ASCO codes appear on records from the beginning of the collection, ANZSCO codes, only appear on records from 1 July 2000. Thus, although the database has a field for ANZSCO, the value is missing for all records prior to this date.

## The need to impute

An historical series of data classified by ANZSCO can only go as far back as 1 July 2000, however, there is a demand to produce series that go further back in time. To satisfy this demand, the missing ANZSCO codes need to be either assigned (a manual process), deduced or inferred.

Manually coding ANZSCO for the older records is not viable due to the amount of resources that would be required.

If ANZSCO codes had a one to one correspondence with ASCO codes, then it would be easy to calculate ANZSCO codes for the older records. If this was the case then there would be no need to invoke a (complex) imputation procedure. However, there isn't a simple one to one mapping between the ASCO and ANZSCO categories.

The remaining option is then to implement an imputation procedure to assign ANZSCO codes to the older data.

## Caveat

It is important to remember that imputation is associated with some degree of uncertainty. The assignment to individual records might not be correct even if broad level aggregates of the data have the correct distributional properties. In general, the finer the level of aggregation, the more risk is involved in using imputed ANZSCO codes. Unfortunately, quantifying the level of risk is not a straightforward task.

# Requirements

## Consistency

The most important requirement was the imputed ANZSCO coding had to be as consistent as possible with the way the existing ANZSCO had been coded. For example, if in a particular state, a given qualification has always been coded to a single ANZSCO category, then that category should be imputed whenever records for that state and qualification are missing a value for ANZSCO.

The basic idea underlying the process is to match a record that is missing ANZSCO with the most similar set of records among those that do have ANZSCO and assign the value shared by those records.

When imputing an ANZSCO for a record, it sometimes occurs that the most similar set of records that do have ANZSCO codes do not have a unique value. In this case, the requirement is to select one of the values with probability proportional to the frequency with which the values occur in the collection. This is a clear case of where the imputed value might be wrong for a given record, but the distribution of imputed values should reflect the distribution of known values.

# Method (overview)

This section gives a broad level description of the imputation process. The steps are executed in the sequence described below. At any point in the process, records that have not yet had an ANZSCO code imputed are passed on to the next step.

More detailed descriptions can be found in the appendices.

## 1. Prepare data

Data from the apprentice and trainee database are read into SAS datasets, keeping only those variables required for the imputation process. Qualification, state, AQF level and ASCO are retained as being particularly associated and ANZSCO.

For some contracts, every record will have an ANZSCO coded. These records can be used to provide data for the imputation process.

Some other contracts will have both records with ANZSCO coded and records without ANZSCO coded. The records without ANZSCO coded are imputed to have the same ANZSCO as the other records in the contract. These records can also be used to provide data for the imputation process.

The remaining contracts have no ANZSCO coded for any records. These records will need to have an ANZSCO code imputed.

## 2. Create imputation tables

Group the records that already have ANZSCO values by qualification, state, AQF level, ASCO and ANZSCO. Tables are created from this sorting that associate ANZSCO codes with all the combinations of the other variables. As a result it is possible to identify which combinations of qualification, state, AQF level and ASCO associate with which ANZSCO codes.

## 3. Match records with missing ANZSCO to the imputation tables

This is just a matching exercise. Records with missing ANZSCO codes are compared with the tables created in the previous step. If the combination of qualification, state, AQF level and ASCO match then the ANZSCO code in the imputation table is used as the imputed value. If the match corresponds to multiple ANZSCO codes, then as previously mentioned, a random selection is made (with probability proportional to frequencies).

Unmatched records pass on to the next step.

# 4. Use ABS ASCO to ANZSCO concordance for imputation

The ABS provides information on the relationship between the older ASCO and the newer ANZSIC classifications. In particular, there is a description of which ASCO codes correspond to which ANZSCO codes.

From this concordance, use only the ASCO codes that map to a single ANZSCO. Whenever a record with missing ANZSCO has been coded with one of these ASCO codes, then use the mapped ANZSCO for imputation.

Unmatched records pass on to the next step.

# 5. Map ASCO to ANZSCO based on distribution in data

Where the ABS maps an ASCO code to more than one ANZSCO code, the apprenticeship collection can give information on how to impute an ANZSCO code. It is possible that an ASCO has only ever been associated with one ANZSCO in the collection even though the ABS concordance gives a one to many mapping. When this is the case, the imputed value is the associated code.

When the collection associates an ASCO code with many ANZSCO codes, then, as once again, a random selection is made (with probability proportional to frequencies).

Unmatched records pass on to the next step.

# 6. Manually code remaining ANZSCOs

Any records still missing an ANZSCO code at this stage have exhausted the process. Fortunately the number of remaining records was small. These records were given an ANZSCO code by manually assigning a value based on the qualifications associated with those records.

# 7. Finalise imputation table

Append the data set of imputed ANZSCO codes from step 1 to the data set of matched records accumulated subsequently.

# Implementation

The method described above was translated into a SAS programme. The view of the apprentice and trainee database current at the end of December 2008 (collection 58) was used as the source data.

The small number of records that were assigned ANZSCO codes manually were associated with just two qualifications. These ANZSCO codes for these qualifications were hard coded into the programme as the last procedure of the imputation processes.

# Concluding Remarks

The fact that an ANZSCO could be assigned to every record in the views does not, by itself, prove the quality of the imputations nor validate the method used. As previously mentioned, the ANZSCO code assigned to any given record is not necessarily correct. The goal is to have aggregates of the data that exhibit the correct distribution of ANZSCO codes. As a general principle, the imputed values should "perform" better with data that is aggregated at broader levels than finer levels

Furthermore, the records without ANZSCO codes span a time period from 1994 to 2000. The records from 2000 should be more like the records with ANZSCO codes present than the records from 1994. For example, the variables collected have changed over time (early records do not even have qualification recorded) and the introduction and uptake of training packages phased in over time. It can reasonably be expected that the quality of the imputation is better for the records closer in time to 2000 and gets progressively worse for records that are from earlier years.

# Appendices

## Appendix 1 Method (detailed)

### 1. Prepare data:

1.1 Create an extract from the apprentice and trainee database.

1.2 Sort the records so that contracts are in time order.

1.3 Identify contracts with at least one record missing ANZSCO.

1.4 Split records into three data sets –

1.4.1 **HAVES**. Records with ANZSCO code present.

1.4.2 **IMPUTED.** Records with ANZSCO missing but present in other records belonging to the same contract - impute the ANZSCO from those records.

1.4.3 **HAVE_NOTS.** Records with ANZSCO missing for all records belonging to the same contract.

1.5 Append a copy of **IMPUTED** to **HAVES**.

### 2. Create imputation tables:

2.1 Sort **HAVES** by Qualification, State, AQF, ASCO and ANZSCO.

2.2 Compress **HAVES** to one record per "by group", keeping count of how many records contribute to each (keep as variable "numerator"). Save as **TEMP**.

2.3 Split **TEMP** into two data sets based on the number of ANZSCO categories per Qualification, State, AQF and ASCO groups. Create data set **ASCO_UNIQUE** to store records where there is only one ANZSCO per group. Create data set **NON_UNIQUE** to store the remainder.

2.4 Sort **ASCO_UNIQUE** by Qualification, State, AQF and ANZSCO and store as **AQF_UNIQUE**.

2.4.1 Compress **AQF_UNIQUE** to one record per "by group"

2.4.2 Retain only those records that are one record per Qualification, State and AQF group.

2.5 Sort **AQF_UNIQUE** by Qualification, State and ANZSCO and store as **STATE_UNIQUE**.

2.5.1 Compress **STATE_UNIQUE** to one record per "by group".

2.5.2 Retain only those records that are one record per Qualification and State group.

2.6 Sort **STATE_UNIQUE** by Qualification and ANZSCO and store as **QUAL_UNIQUE**.

2.6.1 Compress **QUAL_UNIQUE** to one record per "by group".

2.6.2 Retain only those records that are one record per Qualification group.

2.7 With the data set **NON_UNIQUE** (created in 2.3), do the following –

2.7.1 Accumulate "numerator" (see 2.2) for each Qualification, State, AQF, ASCO group (keep as variable "denominator"). Store as data set **DIVISORS**.

2.7.2 Merge **NON_UNIQUE** and **DIVISORS**. For each ANZSCO category associated with a Qualification, State, AQF, ASCO group, calculate variable "imputation_weight" from "numerator" and "denominator". Store in data set **NON_UNIQUE.**

2.7.3 Calculate "cumulative_weight" by accumulating "imputation_weight" within Qualification, State, AQF, ASCO groups.

2.7.4 Restructure **NON_UNIQUE** so that each Qualification, State, AQF, ASCO group has one record which contains details of all associated ANZSCO categories.

## 3. Match records with missing ANZSCO to the imputation tables:

3.1. Sort **HAVE_NOTS** by Qualification, State, AQF and ASCO.

3.2. Match **HAVE_NOTS** against the imputation table **ASCO_UNIQUE**. Store matched records in **MATCHED** and let the imputed ANZSCO be the corresponding value from the imputation table. Store unmatched records as **NOT_MATCHED**.

3.3. Match **NOT_MATCHED** against the imputation table **AQF_UNIQUE**. Store matched records in temporary data set **MATCHED2** and let the imputed ANZSCO be the corresponding value from the imputation table.

3.3.1. Unmatched records remain in **NOT_MATCHED**.Append **MATCHED2** to **MATCHED**.

3.4. Match **NOT_MATCHED** against the imputation table **STATE_UNIQUE**. Store matched records in temporary data set **MATCHED2** and let the imputed ANZSCO be the corresponding value from the imputation table. Unmatched records remain in **NOT_MATCHED**.

3.4.1. Append **MATCHED2** to **MATCHED**.

3.5. Match **NOT_MATCHED** against the imputation table **QUAL_UNIQUE**. Store matched records in temporary data set **MATCHED2** and let the imputed ANZSCO be the corresponding value from the imputation table. Unmatched records remain in **NOT_MATCHED**.

3.5.1. Append **MATCHED2** to **MATCHED**.

3.6. Match **NOT_MATCHED** against the imputation table **NON_UNIQUE**. Store matched records in temporary data set **MATCHED2** and let the imputed ANZSCO be the value selected (randomly) from the imputation table. Unmatched records remain in **NOT_MATCHED**.

3.6.1. Append **MATCHED2** to **MATCHED**.

## 4. Use ABS to ANZSCO concordance for imputation:

4.1. Import concordance into SAS from Excel.

4.2. Sort concordance by ASCO.

4.3. Where an ASCO code maps to a single **ANZSCO** code, store that record in the data set UNIQUE. Otherwise store in data set **MULTIPLE** (currently not used in subsequent processing).

4.4. Sort **NOT_MATCHED** (from 3.6) by ASCO.

4.5. Match **NOT_MATCHED** against the imputation table **UNIQUE**. Store matched records in temporary data set **MATCHED2** and let the imputed ANZSCO be the corresponding value from the imputation table. Unmatched records remain in **NOT_MATCHED**.

4.5.1. Append **MATCHED2** to **MATCHED**.

## 5. Map ASCO to ANZSCO based on distribution in data:

5.1. Append **MATCHED** to **HAVES** and store as data set **ASCO_MAP**.

5.1.1. Sort **ASCO_MAP** by ASCO and ANZSCO.

5.1.2. Compress **ASCO_MAP** to one record per "by group", keeping count of how many records contribute to each (keep as variable "weight").

5.1.3. For ASCOs that associate with only one ANZSCO, store that information in data set **UNIQUE**. Otherwise, store in **NON_UNIQUE**.

5.2. Match **NOT_MATCHED** to **UNIQUE** and store as data set **ASCO_MAP**. Store matched records in temporary data set **MATCHED2** and let the imputed ANZSCO be the corresponding value from **UNIQUE**. Unmatched records remain in **NOT_MATCHED**.

5.2.1. Append **MATCHED2** to **MATCHED**.

5.3. Accumulate "weight" on data set **NON_UNIQUE** by ASCO group (keep as variable "denominator"). Store in **ASCO_TOTALS**.

5.4. Merge **NON_UNIQUE** and **ASCO_TOTALS** and recalculate weight as a proportion.

5.4.1. Delete any records with missing ASCO values.

5.4.2. Restructure **NON_UNIQUE** so that there is one record for every ASCO and calculate cumulative weights.

5.5. Match **NON_UNIQUE** and **NOT_MATCHED**. Store matched records in temporary data set **MATCHED2** and let the imputed ANZSCO be the value selected from **NON_UNIQUE**. Unmatched records remain in **NOT_MATCHED**.

5.5.1. Append **MATCHED2** to **MATCHED**.

## 6. Manually code remaining ANZSCOs:

6.1. Hard code the values for the remaining qualifications on **NOT_MATCHED**.

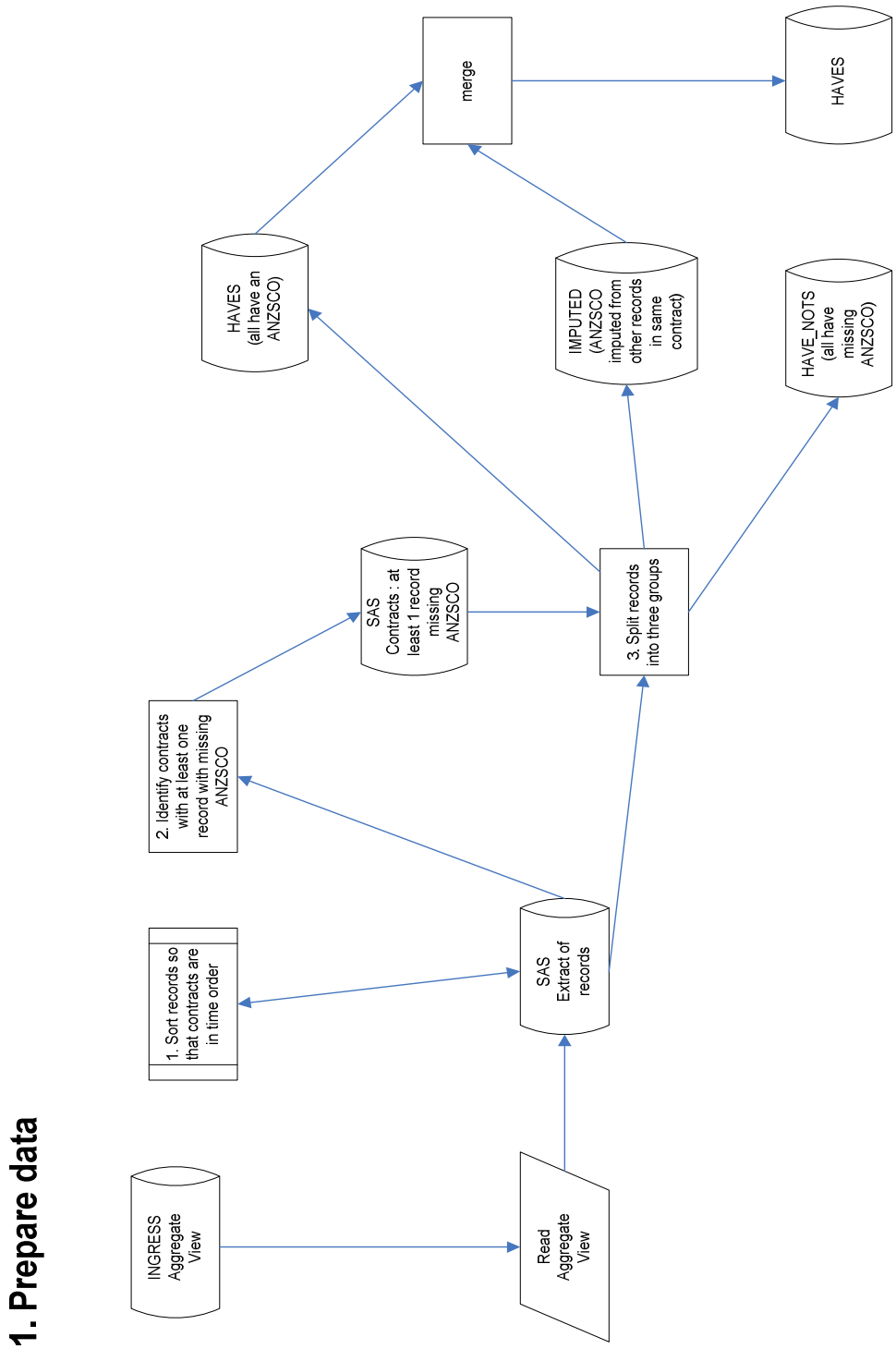
6.2. Append **NOT_MATCHED** to **MATCHED**.

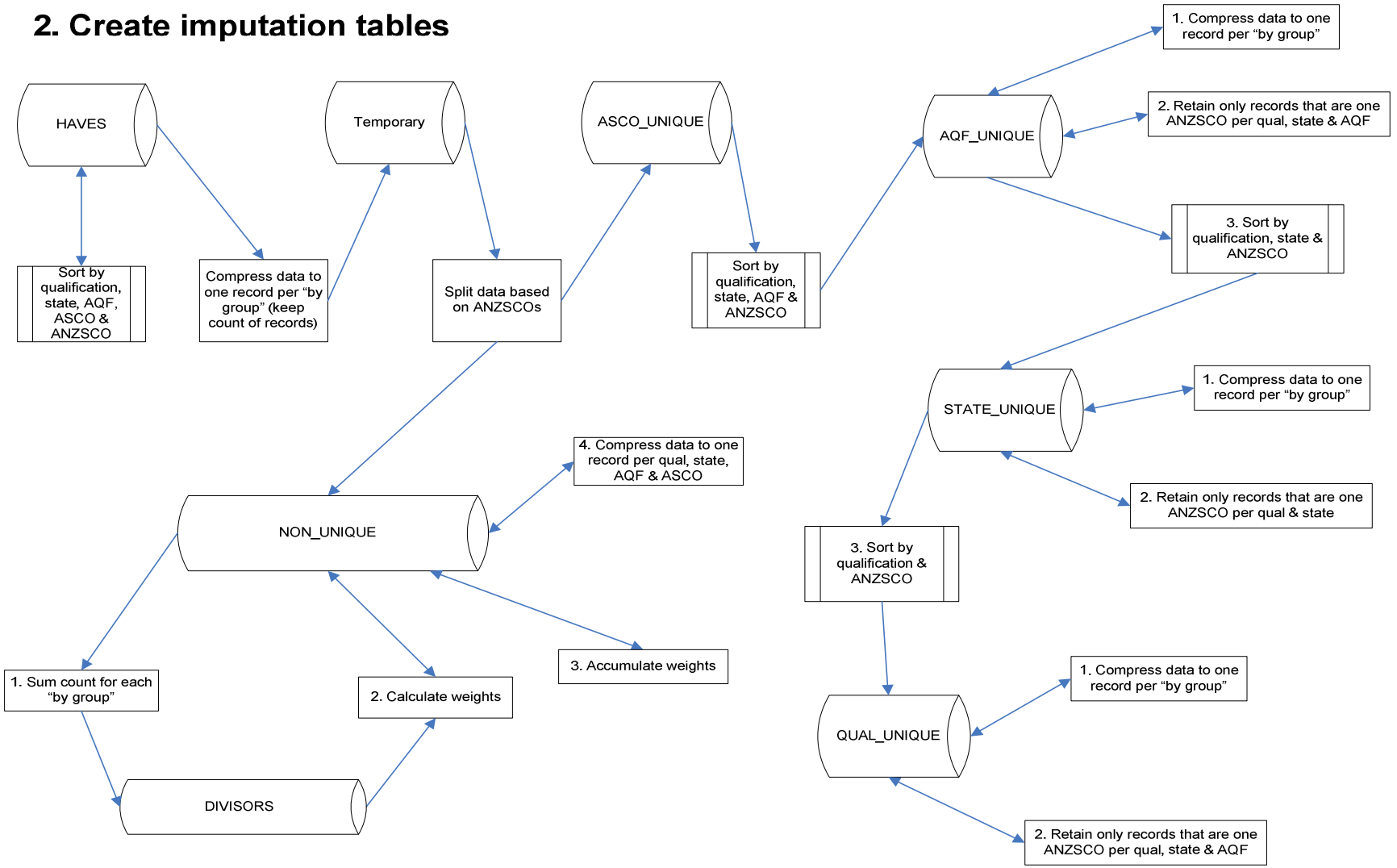
## 7. Finalise imputation table:

7.1. Append **IMPUTED** (from 1.4.2) to **MATCHED**.

# Appendix 2 System charts

This appendix gives a pictorial representation of the imputation process described in appendix 1. The diagram has been split into several parts due to the detailed nature of the process.



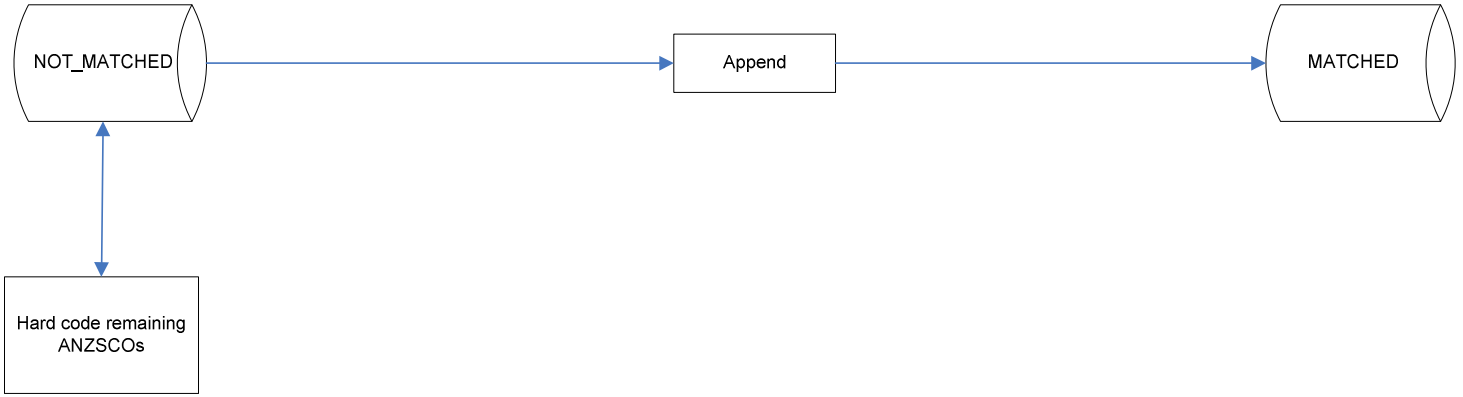

2. Create imputation tables
HAVES
Sort by qualification, state, AQF, ASCO & ANZSCO
Compress data to one record per "by group" (keep count of records)
Temporary
Split data based on ANZSCOs
ASCO_UNIQUE
Sort by qualification, state, AQF & ANZSCO
AQF_UNIQUE
1. Compress data to one record per "by group"
2. Retain only records that are one ANZSCO per qual, state & AQF
3. Sort by qualification, state & ANZSCO
STATE_UNIQUE
1. Compress data to one record per "by group"
2. Retain only records that are one ANZSCO per qual & state
3. Sort by qualification & ANZSCO
QUAL_UNIQUE
1. Compress data to one record per "by group"
2. Retain only records that are one ANZSCO per qual, state & AQF
NON_UNIQUE
4. Compress data to one record per qual, state, AQF & ASCO
1. Sum count for each "by group"
2. Calculate weights
3. Accumulate weights
DIVISORS

3. Match records with missing ANZSCOs to the imputation tables
HAVE_NOTS
Sort by qualification, state, AQF & ASCO
ASCO_UNIQUE
Match records
NOT_MATCHED
AQF_UNIQUE
Match records
NOT_MATCHED
MATCHED2
Append
STATE_UNIQUE
Match records
NOT_MATCHED
MATCHED2
Append
QUAL_UNIQUE
Match records
NOT_MATCHED
MATCHED2
Append
NON_UNIQUE
Match records
NOT_MATCHED
MATCHED2
Append
MATCHED

4. Use ABS ASCO to ANZSCO Concordance for imputation.
Excel ASCO/ANZSCO concordance
Import
SAS ASCO/ANZSCO concordance
Sort by ASCO
Split into unique and multiple mappings
MULTIPLE
UNIQUE
NOT_MATCHED
Sort by ASCO
Match records
NOT_MATCHED
MATCHED2
Append
MATCHED

5. Map ASCO to ANZSCO based on distribution in data so far.
HAVES
Join all records that now have ASCO and ANZSCO
ASCO_MAP
1. Sort by ASCO, ANZSCO
2. Compress to one record per "by group" (keep count of records)
3. Split data into unique and multiple associations
UNIQUE
NOT_MATCHED
Match records
MATCHED2
Append
MATCHED
Append
NOT_MATCHED
NON_UNIQUE
Accumulate weights within ASCO groups
ASCO_TOTALS
Calculate weights
1. Delete any records with missing ASCO
2. Flatten structure of data set and calculate cumulative weights
NON_UNIQUE
3. Match records
NOT_MATCHED
MATCHED2

## 6. Manually code remaining ANZSCOs.




## 7. Finalise imputation table