



This module is part of the

Memobust Handbook

on Methodology of Modern Business Statistics

26 March 2014

Method: Manual Coding

Contents

General section.....	3
1. Summary	3
2. General description of the method	3
3. Preparatory phase	3
4. Examples – not tool specific.....	4
5. Examples – tool specific.....	4
6. Glossary.....	4
7. References	4
Specific section.....	5
Interconnections with other modules.....	7
Administrative section.....	8

General section

1. Summary

We will briefly describe some aspects related to the manual coding of open text answers. In this era dominated by the use of computers, most of the coding is done either by computers or at least computer-assisted. However, there always remains a small part that cannot be coded and needs the attention of an expert.

2. General description of the method

This module will focus on the organisational aspects related to manual coding. Nowadays, manual coding without computer support seems almost unthinkable: for aspects related to the computer-assisted part, see the module “Coding – Computer-Assisted Coding”. Apart from that a number of issues remain:

- Administrative tool(s): when a group of coders is working on the coding of text, it may be useful to have some sort of administrative (workflow) tool to distribute the workload amongst the coders. If there are more classifications to be coded it becomes even more convenient. Such a tool may also include import (from the survey) and export (to the subsequent processing of the survey) options. Also, the coding itself could be done using such a tool: displaying the text(s) relevant to code the answer and allow for searching/browsing the classification.
- Knowledge sharing: many descriptions will be coded, but a small fraction cannot, either because the information is vague or ambiguous. These cases can be discussed and rules can be established to code these difficult descriptions. These discussions will enhance the standardisation of the coding process and help to share knowledge.
- Educating new coders: educating a coder is mostly training on the job. In practice they will code the easier codes at first and leave the ambiguous descriptions to the experienced coders. In our experience it may take many months before they code at the level of the existing coders, for the more complex codes.
- Interaction with code designers: often, the coders are not the ones that design or maintain the classification. Therefore, some amount of interactions is wanted: on the one hand the “code designers” need to explain the philosophy of the classification (e.g., by what criterion are certain occupations grouped together). On the other hand, the coders need to give feedback to the “code designers”: for example, certain distinctions in the classifications may be too subtle which makes it hard to code.

3. Preparatory phase

The preparation consist mainly of the training of the coders. This is a continuous process, since classification systems change over time and there are frequent changes of coders in the coding teams. Therefore, the (new) coders must build up experience with new codes.

4. Examples – not tool specific

5. Examples – tool specific

6. Glossary

For definitions of terms used in this module, please refer to the separate “Glossary” provided as part of the handbook.

7. References

Hacking, W. and Willenborg, L. (2012), *Coding – interpreting short descriptions using a classification*. Contribution to the CBS Methods Series, Statistics Netherlands, The Hague and Heerlen.

Specific section

8. Purpose of the method

The purpose of manual coding method is to describe relevant (e.g., organisational) aspects when coding open text answers from surveys.

9. Recommended use of the method

1. When trying to code open texts from surveys, the order should preferably be:
 - a. Code the answers (semi-)automatically during the interview.
 - b. Code the answers automatically at the statistical office.
 - c. Code the answers manually at the statistical office.

The idea is that the most expensive step is done last.

2. Another reason may be the difficulty of the texts and the necessity to use other variables to arrive at a valid code. Such complex coding processes are much more difficult to automate. Hence, manual coding can be used:
 - for all the texts collected;
 - only for those texts which could not be coded with a computer-assisted method (automatic coding or assisted coding).
3. Due to the disadvantages implied with manual coding, it should be better to:
 - use this method only for texts not coded with the computer assistance;
 - when the text to be coded is not sufficient to assign a code, use other variables to arrive at a valid code.

10. Possible disadvantages of the method

1. Compared to other coding scenarios, manual coding is rather expensive. Other disadvantages are:
 - this method is error prone when the knowledge/experience of the coder is not sufficient;
 - manual coding results in less standardisation of the process (each coder, even if well trained, has his own knowledge and can make deductions according to his interpretation of the text).

11. Variants of the method

- 1.

12. Input data

1. A text to be coded, possibly combined with a number of other variables correlated with the classification at hand, e.g., the kind of goods when coding economic activity.

13. Logical preconditions

1. Missing values
 - 1.
2. Erroneous values
 - 1.
3. Other quality related preconditions
 - 1.
4. Other types of preconditions
 - 1.

14. Tuning parameters

- 1.

15. Recommended use of the individual variants of the method

- 1.

16. Output data

1. For each input text, a code is added unless the text has not an informative content sufficient to assign a code and there are no other variables which could help to arrive at a valid code.

17. Properties of the output data

- 1.

18. Unit of input data suitable for the method

Incremental processing

19. User interaction - not tool specific

- 1.

20. Logging indicators

1. The coder may log which variables were used to arrive at a code. Such a scenario would only be feasible if the manual coding is supported by a computer program, though: the program could trace the interactions of the coder while trying to arrive at a proper code. For example, the coder may use additional variables if the text is ambiguous.

21. Quality indicators of the output data

1. The quality of coding can be measured with two indicators (as described in “Coding – Measuring Coding Quality”):
 - Coding rate (efficacy) → percentage of coded texts on the total of texts to be coded;

- Precision rate (accuracy) → percentage of *correct* coded texts on the total of coded texts.

The verification of coding can be performed by having a different team of coders recode a sample of the texts. If the original code and the verification code differ, the ‘correct’ code can be decided by expert coders by a reconciliation process. The set of correct codes can then be used to estimate the values for coding rate and precision rate.

22. Actual use of the method

- 1.

Interconnections with other modules

23. Themes that refer explicitly to this module

1. Coding – Main Module
2. Coding – Measuring Coding Quality

24. Related methods described in other modules

1. Coding – Automatic Coding Based on Pre-coded Datasets
2. Coding – Automatic Coding Based on Semantic Networks
3. Coding – Computer-Assisted Coding

25. Mathematical techniques used by the method described in this module

- 1.

26. GSBPM phases where the method described in this module is used

1. 5.2 Classify and code

27. Tools that implement the method described in this module

- 1.

28. Process step performed by the method

Coding

Administrative section

29. Module code

Coding-M-Manual Coding

30. Version history

Version	Date	Description of changes	Author	Institute
0.1	02-04-2013	first version	Wim Hacking	CBS
0.2	20-01-2014	following review by Stefania Macchia	Wim Hacking	CBS
0.3	30-01-2014	following review from EB	Wim Hacking	CBS
0.3.1	30-01-2014	preliminary release		
1.0	26-03-2014	final version within the Memobust project		

31. Template version and print date

Template version used	1.0 p 4 d.d. 22-11-2012
Print date	21-3-2014 18:06