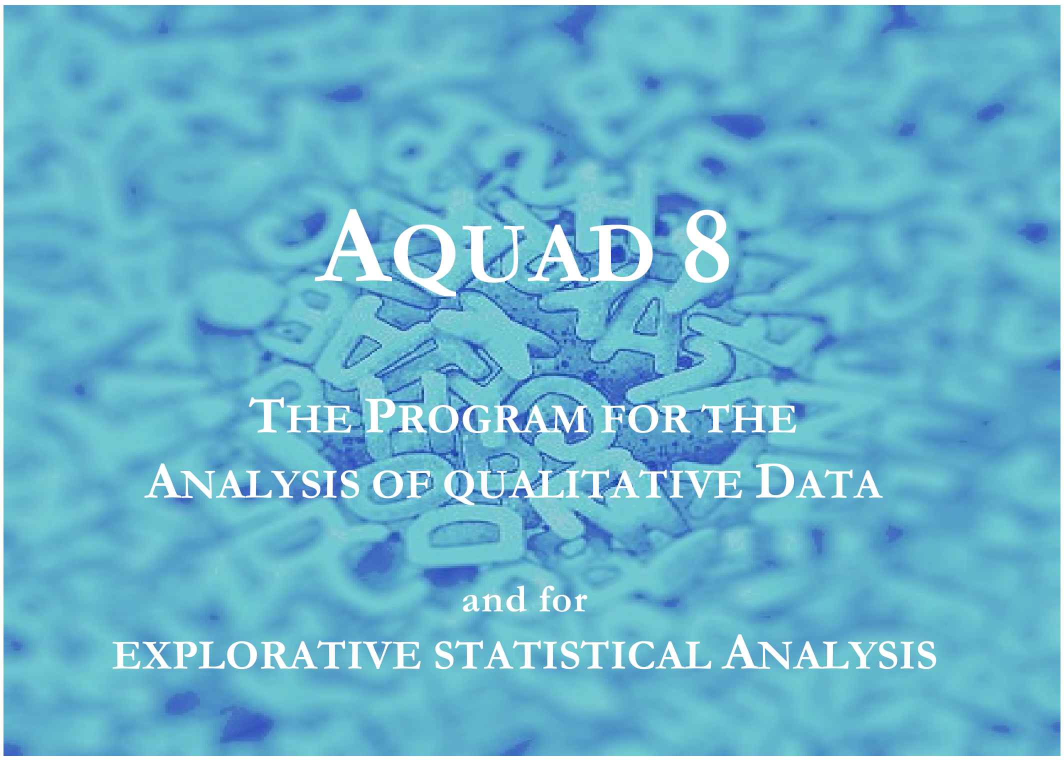


AQUAD 8

THE PROGRAM FOR THE ANALYSIS OF QUALITATIVE DATA

and for

EXPLORATIVE STATISTICAL ANALYSIS

Günter L. Huber
Leo Gürtler

AQUAD 8 is programmed with LAZARUS 2.0.10 / FPC version 3.2.0 and is compiled as 32-bit version. To analyze audio or video recordings (with aquad_8d_audio.exe or aquad_8d_video.exe) AQUAD 8 uses the 32-bit version of the media player "VLC" (www.videolan.org). In addition, the software includes scripts for use with the free statistical package "R" (www.r-project.org). AQUAD 8 shows, copies, prints etc. results with "notepad.exe". Notepad.exe is included by default in "Microsoft Windows", but "R" and "VLC" may need to be installed on your computer.

1st edition 2021

Authors: Günter L. Huber and Leo Gürtler

© Günter Huber, Viktor-Renner-Str. 39, 72074 Tübingen, Germany
phone 07071-885147
e-mail: info@aquad.de

Contents

Introduction

The first version of AQUAD (Analysis of Qualitative Data) was developed in 1987 so that a research project could be accomplished within a tight grant period. At that time, there were already some software tools for analyzing qualitative data. The simplest ones used the search functions of available text or database programs. Some programs had been written specifically for the requirements of qualitative analysis, but usually offered no more than simple counting and search functions. In contrast, a qualitative leap was made by an American program, the Qualog software package by Anne Shelly and Ernest Sibert for large computers. It used the advanced possibilities of so-called "logical programming" and was the inspiration and model for AQUAD.

In particular, AQUAD was developed to support the finding of meaning relationships in data material, as described, for example, by the methodological approach of "Grounded Theory" (Glaser & Strauss, 1967). In addition, methodological ideas of Miles and Huberman (1984, 1994) on tabular analysis or matrix analysis have been incorporated into AQUAD, as well as the qualitative comparative analysis approach of Ragin (1987) for comparing configurations of meaning in different data sets. Since version 7, a module for objective hermeneutic text interpretation according to Oevermann (1981) has also been incorporated, following Wernet's (2009) suggestions for sequential analysis. Simple functions e.g. for managing files, searching for encodings, counting words etc. are of course also available in AQUAD.

In order to simplify the software and better adapt it to the needs of users, AQUAD has been split into five separate subroutines in version 8:

- Text files of the "*.txt" format are analyzed qualitatively most frequently. For this purpose, "AQUAD_8e.exe" is available.

- Even without complex transcription, for example, audio recordings of interviews in many formats (e.g. "*.wav", "*.mp3", "*.aac" etc.) can be analyzed directly with the version "AQUAD_8e_audio.exe". Of course, possibilities like searching for keywords or the written output of typical text passages are omitted. On your computer the 32-bit version of the media player VLC must be available (www.videolan.org).

- Video recordings (in numerous formats, e.g. "*.avi", "*.mp4", "*.mov", etc.) can also be analyzed without transcription with the version "AQUAD_8e_video.exe". The 32-bit version of the media player VLC must be available on your computer (www.videolan.org).

- Photos, scanned drawings and other graphical data provided in files of the format "*.jpg" or "*.png" can be analyzed with the version "AQUAD_8e_graph.exe".

- These four subprograms automatically install a fifth separate module "AQUAD-eda.exe", with which the results of frequency evaluations can be further evaluated and graphically displayed according to the principles of exploratory data analysis (Tukey, 1977), e.g. with cluster analyses, multidimensional scaling, etc. The four versions for qualitative analysis therefore automatically save results of frequency evaluations in tables of the format "*.csv". These can be exported directly into programs for spreadsheet analysis (e.g. "Excel") or statistical analysis such as "R". The module "AQUAD-eda.exe" loads corresponding scripts, which are also installed automatically. On your computer "R" (www.r-project.org) must be installed.

The operating logic of the program and the internal processing of the encodings differ only slightly for the various file formats in AQUAD, so that familiarization is easy. There are no fundamental changes for users of version 7; the existing files can be converted for version 8. When encoding texts in version 8, only the line numbers of the selected text segments are taken into account for localization in the text, but no longer also their exact beginning and length (number of characters) in the text file. Existing code files from version 7 limit the localization of the text segments to the line numbers after the conversion.

Chapter 1: Installing the AQUAD program

On our website www.AQUAD.de you will find in the section "Download" in the section "AQUAD 8" the following Installation modules of the program package:

- AQUAD8-en-setup.exe Module for qualitative and quantitative analysis of texts according to the coding paradigm and the reconstruction paradigm (see chapter 3).
- AQUAD8-en-audio-setup.exe Module for qualitative analysis of audio recordings
- AQUAD8-en-video-setup.exe Module for qualitative analysis of video recordings
- AQUAD8-en-grafik-setup.exe Module for the analysis of graphical data material (photos, drawings, handwritten recordings etc.)

These modules also install the program "AQUAD_eda.exe" for explorative data analysis according to Tukey (1977), which is realized with the statistical package "R". Corresponding algorithms ("scripts") for "R" are included in the installation routine of AQUAD. Therefore, "R" must be pre-installed on your computer. When setting up a project for the first time, AQUAD asks for the directory where "R" (more precisely: "Rgui.exe") can be found.

AQUAD 8 uses 32-bit algorithms from Robert Olsztyn (https://prog.olsztyn.pl/paslibvlc/) for audio and video playback with the free VLC MediaPlayer. The audio and video versions of AQUAD 8 will work fine if you install the **32-bit version of VLC** on your computer.

Functions of text processing of results are provided by "notepad.exe", a component of Microsoft Windows. If you want to print, cut, copy, paste, save results to other than AQUAD directories, etc., please use the functions in the menu of the respective window of "notepad.exe".

AQUAD displays all results of frequency evaluations as lists in "notepad.exe" and also saves them as tables in "*.csv" format for further exploratory data analysis or import into statistical programs.

During the installation AQUAD automatically creates the following directories on your computer. For simplicity we assume here that you install the module for text analysis and you take the suggestion "C:\AQUAD_8d" of the installation routine for the root directory of all files related to AQUAD (for audios, videos and graphic files the specifications apply analogously):

Directory	*Content*
C:\AQUAD_8d	base or root directory; here supplied sample texts are installed automatically. The texts (*.txt) of your project must also be stored in this directory.
C:\AQUAD_8d\cod	Codes which are created during your interpretation work (*.aco); Here you will also find the parameters of the projects and directories of codes, keywords, etc. that you are likely to save for further analysis.
C:\AQUAD_8d\cod_s	This directory is used to back up your codes, especially so that earlier codes can be restored in case you merge different meaning-like codes and replace them with the new metacode(s).
C:\AQUAD_8d\lit	This manual. Here you can also store publications, links, etc. that are important for your work on the current project.
C:\AQUAD_8d\mco	Metacode files (summaries of meaning-like codes).
C:\AQUAD_8d\prg	The program itself, the help file and parameters of the current project.
C:\AQUAD_8d\res	The results of all evaluations (results) as txt and/or csv files.
C:\AQUAD_8d\scripts	The R scripts for further explorative statistical analysis of your results.

Chapter 2: Preparing and saving data

With the appropriate module (see above) of AQUAD 8, you can analyze

Text files in **.txt* format,
audio files in all sound formats available in the VLC media player (e.g. **.mp3 , *.wav, *.aac,* etc.),
video files in all video formats available in the VLC media player (e.g. **.mp4, *.avi, *.mov*, etc.) or
graphic files in **.jpg* or **.png* format.

These files must be stored in the **root directory** of AQUAD (in the example of text analyses in Chap.1: C:\AQUAD_8d), since the program searches for the data to be analyzed there by default. Of course, you can back up your original files elsewhere, even on another disk, for safety, but a working copy must be available to AQUAD in the root directory. For preparing text files for analysis, here are some hints that have proven helpful to many AQUAD users:

2.1 Working with text files

Before you define your own project, you must prepare your text files so that AQUAD can access them. Specifically, you must

- format your text documents in your word processing program,
- convert them to *txt* format and
- then copy them to the root directory of AQUAD (see installation: "C:\AQUAD_8d").

2.1.1 Format your text files

You can use any word processing program to enter your data. However, before you can work meaningfully with the texts within AQUAD, you must prepare each file for it. First, reformat the text so that the lines contain no more than about 60 characters, a value that has proven effective in practice. The lines of this text are about 90 characters each, so they should be shortened by about a third and left-justified, like this:

```
For the input of your data you can use any word processing
program to enter your data. However, before you can work
with the texts within AQUAD, you have to prepare each file
for this purpose. First, format ...
```

The reason for this is that AQUAD uses line numbers to mark meaningful text segments. So the individual lines should not be too long, so that the codes can be precisely assigned to the relevant parts of the text.

This formatting is best done with your text program while you are entering the data. However, you can also reformat the data later (see next paragraph). With most word processing programs you can determine the length of the lines by setting the margins accordingly (no margin on the left side, on the right side the margin should be set so that there is room for about 60 characters).

But what do you do if your texts have already been entered with the computer and have been formatted to "fill the page"? No problem: with any text program, even with many very simple text editors, you can subsequently reformat texts according to the principle just described. Many text programs usually allow you to specify a page format (including line lengths) to which all texts are then automatically adapted. You then only have to load your texts in sequence - and then save them again. Please refer to the *manual of your text program* for details.

2.1.2 Name the files sensibly

When you save your text files, it makes sense in AQUAD that the last three characters of the name are digits. As an agreement for naming text files, AQUAD has the following:

- File names are 60 characters or less.
- The last three characters of the name should be digits.
- As extensions, text programs append ".txt" to file names when converting to plain text files; AQUAD uses these extensions to identify text files.
- For anonymization, all text files in a project are given the same name but different end digits. Or, in other words, the files of a project are numbered consecutively (not necessarily with sequential numbers); each file is assigned a different three-digit number.

You will probably choose the names so that the contents of the files are recognizable to you. For example, if your data consists of 24 interview transcripts and you have chosen "interview" as the name of your project, you might name your original files as follows:

interview_001.txt, interview_002.txt, ..., interview_024.txt.

It is very important that the file names are unique, i.e. that they differ in numbers. Of course you can do without numbers and use different names, e.g. "Meyer_Hegelschule.txt". However, by doing so you renounce anonymity of the interviewees and it becomes difficult if you store data of several projects in the same root directory and want to make a specific selection from it. Also, never use "work" as a name because AQUAD uses it internally!

While AQUAD is working with your files, it also creates parallel files that contain, for example, the code information for each file. The program creates new file names from the elements of the original names in such a way that the connection to the original files can be recognized. In our example, the code files would be named as follows:

interview_001.aco, interview_002.aco, ..., interview_024.aco.

After coding, you can see these names in the AQUAD subdirectory "..\cod", which is automatically set up for storing code files. In this way, when working with AQUAD, many different special files are created, and you do not have to worry about naming them exactly. For this reason, we will not give an exhaustive list here.

If your files are stored with names that deviate from this convention, you can rename these files. You can do this either within a "file manager" like Microsoft's "Windows Explorer" by right-clicking on the file name and selecting "Rename" from the pop-up menu that appears. Or you can type "CMD" (for "command") in the search window at the bottom of the Windows taskbar. Make sure that the subdirectory of your text files (original texts) is displayed and then use the command "REN" (for "rename") as follows:

```
REN old_name new_name
```

Let's assume that the interview texts of our example were originally named after the interviewees (which we would advise against, if only for reasons of data protection). Let's take a look at the procedure for the first two texts, which are stored as wolfgang.txt and andrea.txt. We rename them like this

```
REN wolfgang.txt inter_001.txt
REN andrea.txt inter_002.txt
```

and proceed analogously with the other 22 texts of our example.

2.1.3 Convert the texts into the ANSI text format (*.txt)

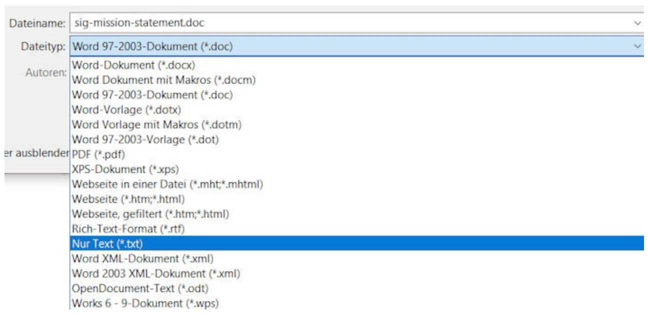
All text documents that you want to use in AQUAD must be converted to ANSI format (*.txt). "ANSI" is an abbreviation and registered designation for the American National Standards Institute; ANSI code is the code commonly used in Windows programs to represent characters and exchange text information between different programs.

When you write text, your word processor adds codes for font size, color, word endings, and other information to your text invisibly to you, such as instructions to the program to start a new line, set margins, indent lines, start a new page, italicize or bold or color the text, and so on. This also includes instructions to the printer on when to start a new page and much more. Unfortunately, these instructions are program-specific and cannot be read in other programs - unless these programs contain a transformation routine, for which, however, license fees often have to be paid. For trouble-free export and import of text files, all characters that only make sense in a particular program must be removed, i.e. the files must be standardized. For import into AQUAD, your specific data documents must be converted into standard ANSI documents.

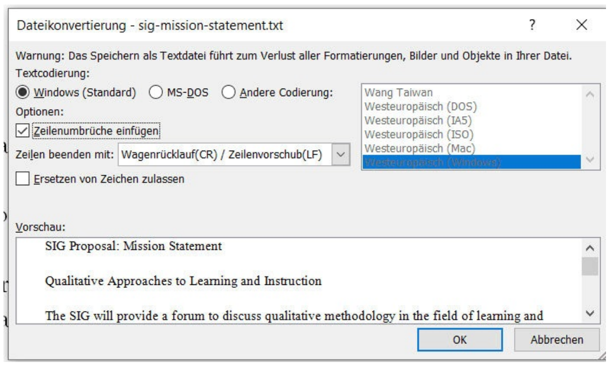
Therefore, for data exchange, text programs usually have a helper function that converts your documents into ANSI texts (txt). Inform yourself in the manual of your word processing program under "ANSI text", "txt format", "import/export", "text-in, text-out" or similar keywords.

The following two screenshots show the procedure when working with MS Word:

The option "Save as/Speichern unter" was selected, then the file type must be changed from *.doc to "Only Text/Nur Text (*.txt)".


Dateiname: sig-mission-statement.doc
Dateityp: Word 97-2003-Dokument (*.doc)
Autoren:
er ausblenden

Word-Dokument (*.docx)
Word Dokument mit Makros (*.docm)
Word 97-2003-Dokument (*.doc)
Word-Vorlage (*.dotx)
Word Vorlage mit Makros (*.dotm)
Word 97-2003-Vorlage (*.dot)
PDF (*.pdf)
XPS-Dokument (*.xps)
Webseite in einer Datei (*.mht;*.mhtml)
Webseite (*.htm;*.html)
Webseite, gefiltert (*.htm;*.html)
Rich-Text-Format (*.rtf)
Nur Text (*.txt)
Word XML-Dokument (*.xml)
Word 2003 XML-Dokument (*.xml)
OpenDocument-Text (*.odt)
Works 6 - 9-Dokument (*.wps)


Dateikonvertierung - sig-mission-statement.txt

Warnung: Das Speichern als Textdatei führt zum Verlust aller Formatierungen, Bilder und Objekte in Ihrer Datei.
Textcodierung:
[✓ Windows (Standard)] MS-DOS Andere Codierung:
Wang Taiwan
Westeuropäisch (DOS)
Westeuropäisch (IA5)
Westeuropäisch (ISO)
Westeuropäisch (Mac)
[✓ Westeuropäisch (Windows)]
Optionen:
[✓ Zeilenumbrüche einfügen]
Zeilen beenden mit: Wagenrücklauf(CR) / Zeilenvorschub(LF)
Ersetzen von Zeichen zulassen

Vorschau:
SIG Proposal: Mission Statement

Qualitative Approaches to Learning and Instruction

The SIG will provide a forum to discuss qualitative methodology in the field of learning and

OK Abbrechen

In the window that appears then we leave the default settings "Windows" and to the right of it the language unchanged, but we click on "Insert line breaks" and check that under it is marked that the line ends are executed with the standard codes "CR/LF" (Carriage Return/Line Feed).

You do not need to read the next section if you only want to orientate yourself once at first. It will only become important when you are working with AQUAD and realize that you still have to change something in one or the other file.

Editing files after conversion to ANSI text format

If you find errors in your original data, you may wish to correct them. However, there is a basic rule in qualitative analysis that text documents should not be changed during the analysis. So check the texts before you start qualitative text analysis. For your interpretations in the form of codes, AQUAD stores the line numbers of each coded text segment. If you delete or add letters or whole words, you may change the line structure of your text and thus the assignment of the codes.

2.2 Working with audio files

To use the modules for analyzing audio recordings, the 32-bit version of the free media player VLC (download: www.videolan.org) must be installed on your computer.

The recommendations for naming text files (see above) also apply to audio files; in particular, file names should not be longer than 60 characters.

If you want to analyze audio files that are already available, save them in the root directory of AQUAD in the same way as for text files. For example, if you want to record interviews first, you can simply connect a microphone to the appropriate input of your notebook and use the device as a recorder with appropriate software (we recommend the free software Audacity, available for download from http://audacity.sourceforge.net/download). A digital voice recorder or a corresponding app on a smartphone are also suitable for most purposes. Again, you will need to copy the audio files from these devices to AQUAD's root directory.

2.3 Working with video files

To use the modules for analyzing video recordings, the **32-bit version** of the free media player VLC (download: www.videolan.org) must be installed on the computer.

The recommendations for naming text files (see above) also apply to video files; in particular, file names should not be longer than 60 characters.

If you want to analyze video files that are already available, save them in the root directory of AQUAD in the same way as text files. If you want to record interviews or interaction sequences on video, you can use digital camcorders or the video function of a digital camera or smartphone. Again, you must copy the video files from these devices to the AQUAD root directory.

2.4 Working with graphic files

Preparation is not necessary for your image files. AQUAD 8 is prepared for importing images in "*.jpg" and "*.png" format. Therefore, using your scan or image processing program, please save images intended for analysis in AQUAD preferably in the format

.jpg (compressible, small and handy).

Other formats such as .tif or .bmp require much more memory. The image files can be opened directly in AQUAD, as you are used to from texts. The dimensions of the images are automatically and proportionally adjusted.

Since virtually all graphics programs and also many image viewing programs ("viewers") can convert image files from almost any common format to almost any other, it seemed unnecessary to bloat AQUAD with a collection of conversion routines. You can also find downloadable viewing and converting programs on the Internet, for example, on the pages of many computer magazines, freeware sites, and university servers.

We have good experience with the program "IrfanView" by Irfan Skiljan. For non-commercial use, for example in university research projects, the program can be obtained free of charge from

http://irfanview.tuwien.ac.at

Chapter 3: The structure of the AQUAD program

3.1 Paradigms of data analysis

In the qualitative analysis of data, in any case, the interpretation of the content, i.e. the determination of a concrete meaning of relevant content segments, is necessary at critical stages of the process. Let us first look at the processes and the program functions provided for this purpose in the case of the frequently used coding paradigm (cf. Overview on the following page):

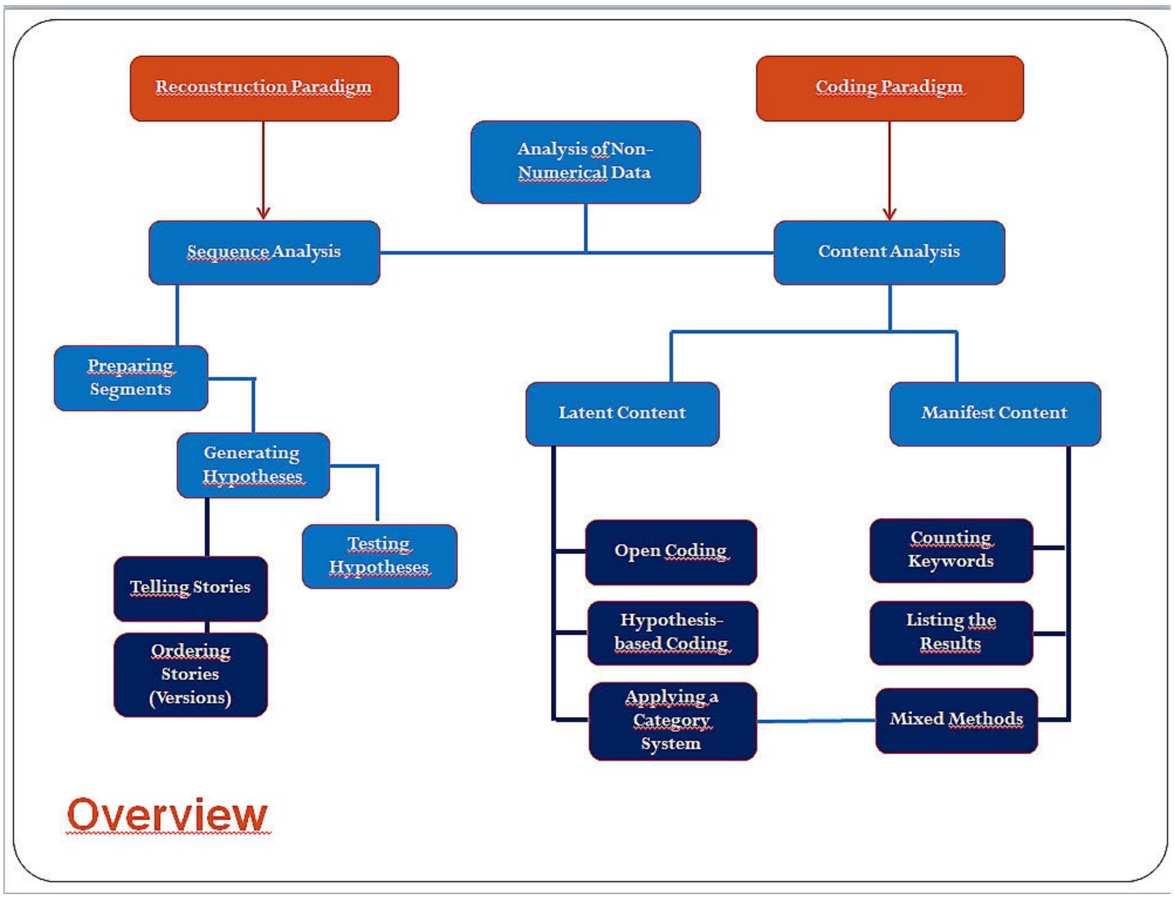
The purpose of qualitative analysis of latent content, i.e., the meaning of non-numerical data, is to reduce the descriptions, explanations, justifications, field notes, observation protocols, interviews, etc., which are usually formulated in a variety of ways, so that the original data are summarized into systematic statements about their meaning. The way this task is accomplished varies from individual to individual and ultimately depends on the research question of a particular study. However, one invariant is found in all qualitative analyses that follow the coding paradigm: the classification or categorization of data and their representation by codes. Categories can be thought of as "containers" into which data are placed according to their meaning. One can

- proceed completely openly and inductively develop categories as they "emerge" in the interpretation of the data or even as they emerge in the course of data collection (cf. Glaser & Strauss, 1979).
- Develop categories deductively based on key research questions or hypotheses under which one approaches the interpretation of the data.
- Make use of a system of categories or codes already available in the research literature or one's own work.

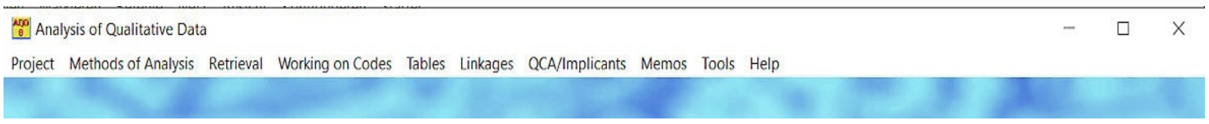
AQUAD supports both the deductive and inductive process and also a combination of both. All file types (texts, audios, videos, images) that AQUAD works with can be analyzed.

In the quantitative analysis of manifest content, the determination of frequencies and frequency differences of elements critical to the research question in the data is necessarily preceded by a qualitative-interpretative phase: It must be determined in advance what is to be counted subsequently, i.e., in the case of texts, the critical keywords. In this phase, hypotheses or theories and constructs relevant to the research question play an important role in the selection. Subsequently, the quantitative results are ordered, i.e. summarized in thematic lists or tables. In the sense of the "mixed methods" approach, these quantitative findings can be compared with the qualitative findings of the latent content interpretation. The qualitative-interpretative definition of keywords or lists of meaning-related words enables the automatic evaluation of texts in AQUAD - but, of course, not of audio, video and image files.

In AQUAD, the possibility of qualitative analysis according to the reconstruction paradigm as "sequence analysis" according to the approach of Objective Hermeneutics (Oevermann, 1996; Oevermann, Allert, Konau, & Krambeck, 1979) is integrated in a separate module. In contrast to analyses of qualitative data according to the coding paradigm, sequence analyses do not start from an overview of the entire text and do not search for file segments that are significant for the research question; instead, in a first phase of hypothesis generation, all somehow conceivable meanings are noted for each text segment (sentence, sentence part, word sequence). In concrete terms, this means that one tries to tell stories in which the text segment could occur in a meaningful way. The multiplicity of possible interpretations is reduced in the course of the analysis, nonsensical, i.e. contradictory "stories" are eliminated, meaning-same and/or coherent "stories" are each reduced to a common "version". Hypothesis generation is strictly sequential, file segment by file segment. Only when one is convinced to have noted hypotheses significant to the research question on the sifted data segments, the rest of the data is used to confirm or falsify all noted hypotheses. Only at this stage does one go through the remaining text segments non-sequentially in search of justifications for retaining or rejecting the previously generated hypotheses.



3.2 Modules and functions for data analysis

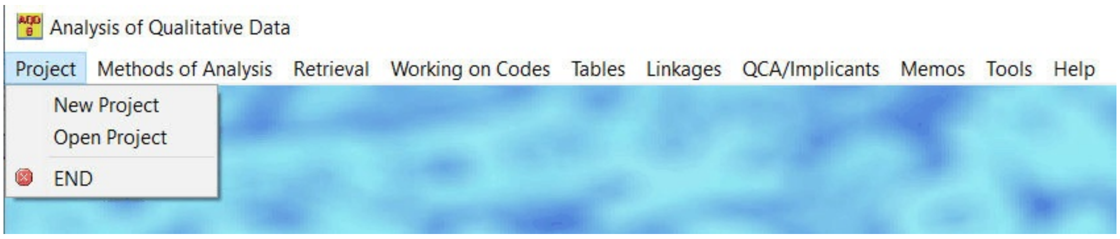


The following descriptions except those for "Methods of Analysis" apply to all variants of AQUAD, i.e., to the modules for text, audio, video and image analysis, since after the determination and interpretation of units of meaning in the data material, further operations are basically performed with the codes. The codes contain the necessary information for the localization of a meaning unit in the file (beginning and end of the interpreted segment) and the meaning itself (expressed by a code): in the case of the example video "Lucy with box" you will find, among others, the coding " 195 200Biting", which expresses that the filmed puppy bites between the image positions 195 and 200. For localization in the other cases line numbers, times of the sound recording or image coordinates are available.

Methods of analysis available in AQUAD include qualitative, quantitative, and objective-hermeneutic approaches for text data only. Sound data, video data and image data can be divided into meaning segments here only qualitatively-interpretatively. For this reason, the manual contains separate sections for them.

3.2.1 Project

In the item "New project" you give your research project a name and then select all the files (texts, audios, videos, images - depending on the AQUAD module you are working with) that you want to analyze in your project.



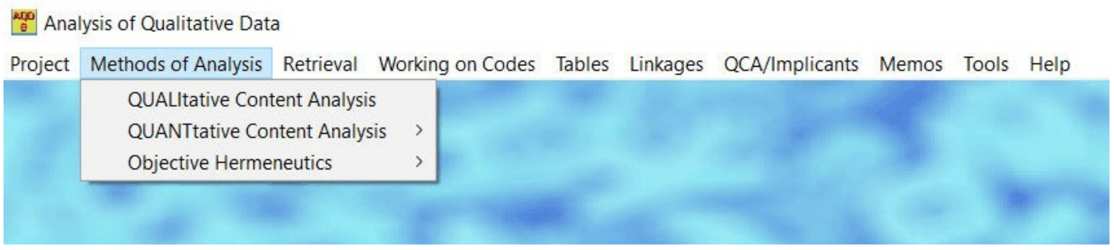
You only need the "Open project" function if you are working on several projects at the same time, for example if you have grouped certain texts in certain project versions and want to switch between the projects. Otherwise, AQUAD automatically reopens the last open project each time you start it.

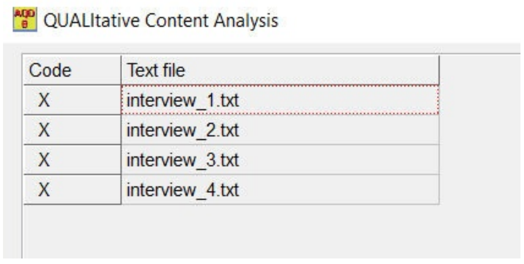
With the function "END" you conclude your work in each case and leave the program environment of AQUAD.

3.2.2 Methods of analysis

Since the beginnings of a broad social scientific reception of text analytic methods, content analysis has been carried out either as qualitative or as quantitative analysis. In the same year, 1952, seminal articles by Kracauer and by Berelson each appeared. While for Kracauer qualitative content analysis attempts to reveal the hidden, latent categories of meaning in the text independently of the manifest textual content, for Berelson quantitative content analysis serves to systematically, objectively, and quantitatively capture manifest communication content.

Selecting "QUALItative content analysis" immediately opens a screen for selecting one of the texts in the project and then the working window for latent content analysis (see below).

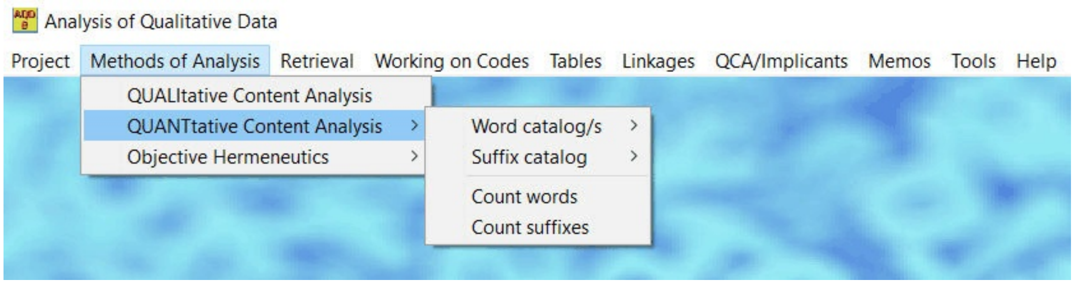



QUALItative Content Analysis

Code	Text file
X	interview_1.txt
X	interview_2.txt
X	interview_3.txt
X	interview_4.txt

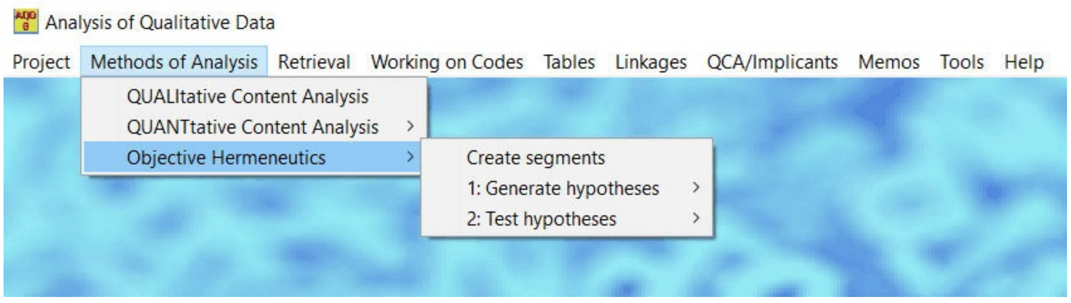
The following options "QUANTitative content analysis" and "Objective hermeneutics" are missing in the AQUAD modules for sound, video and image analysis. Here you will only find the option "QUALitative Analysis" and then the option to select a file of your project for further analysis.

The "QUANTItative content analysis" selection offers the possibility or, more precisely, requires you to enter the critical elements to be counted into a catalog, select an already created catalog, merge catalogs or edit an available catalog. The same applies mutatis mutandis if one does not want to count keywords but certain suffixes.



The quantitative analysis then starts with "Count words" or "Count suffixes". The results of these functions are stored according to various conditions, e.g. lower and upper limits of consideration of elements, in lists and in CSV tables ("comma separated values") in the subdirectory "..\res".

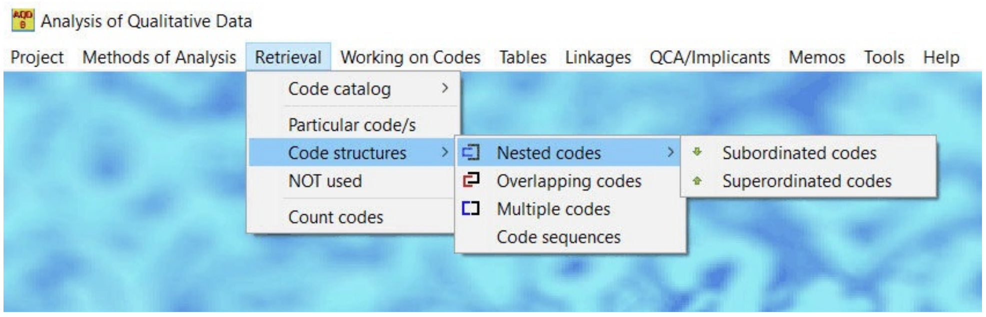
To work according to the conditions of the reconstruction paradigm of Objective Hermeneutics, that is here in AQUAD with the method of sequence analysis, first of all the text sequences must be prepared. The function "Preparation" offers to split the text in advance into complete sentences or into sentence parts (each up to the next punctuation mark). During the later analysis, only one of these sequence elements is displayed at a time for generating hypotheses ("telling stories"). In a third variant, one current segment at a time can be assembled by selecting any number of consecutive words.



In a first analytical phase of hypothesis generation all somehow conceivable meanings are noted for each text segment. Hypothesis generation is strictly sequential, file segment by file segment. Only when one is convinced to have noted all hypotheses significant for the research question on the sifted data segments, the rest of the data is used to confirm or reject all noted hypotheses. In this second phase of hypothesis testing, one goes through the remaining text segments non-sequentially in search of justifications for maintaining or rejecting the previously generated hypotheses.

3.2.3 Retrieval

There are many reasons why a researcher might want to survey all file locations that deal with the same topic, i.e., areas in text, every now and then during the course of a project. Audio or video recordings, or images that fall under the same category. One of these reasons, which plays a large role in many descriptive-interpretive studies (see Tesch, 1990), is to detect commonalities across the entire database. Another reason can be seen in the effort to control the consistency of coding; one has to make sure that they have always been used according to the same principles.



Other reasons can be found in looking for evidence of where code has been used correctly and where it has been used incorrectly, where there are parallels between certain files, etc. AQUAD shows the areas searched for in all files. This is done in the order in which they are listed in the file directory (created when a new project is created). Because one file is searched at a time, we call this search one-dimensional or linear analysis. It differs from two-dimensional analyses, called table or matrix analyses in AQUAD, and from analyses of complex links. The results of one-dimensional analyses can be read on screen, printed on paper, or initially saved to disk in txt format.

The linear search for coded file segments is started in the main menu item "Retrieval" with the option "specific code/s". It is possible to search for several codes (from a code catalog) at the same time. It is also possible to search for certain coding structures, namely whether a code occurs superior or subordinate to another, whether its range of meaning overlaps with that of other codes, whether a unit of meaning has been marked several times with different codes, whether a code occurs in a certain sequence with others, and finally whether such code sequences are repeated. In addition, the files can be compared for whether certain codes were not used for interpretation. Finally, the frequency of codes can be counted and, in video and audio files, the length of certain scenes can be counted.

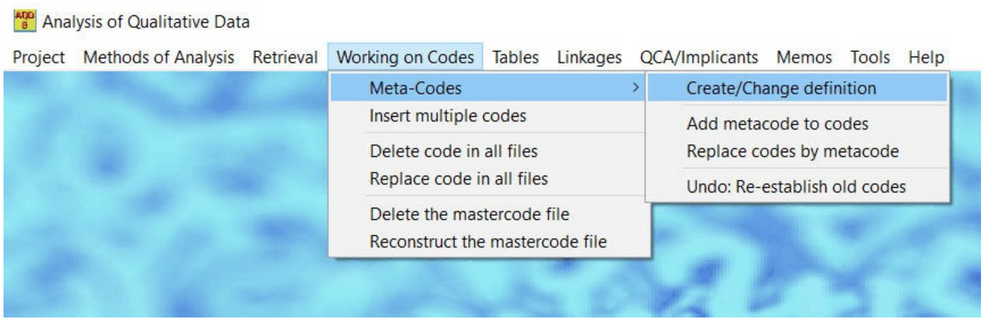
It is also possible to search for several codes at the same time. These codes are summarized with the first function ("code catalog") in lists.

In the program AQUAD_8d_grafik.exe for the analysis of graphic material can be searched with the option "code structures" only for multiple codes, since in the picture material the total information is available simultaneously and the coordinates of the marked picture sections do not define necessarily a sequence. Inclusions and overlaps could in principle be analyzed, but so far there are no user requests for this.

With information about the frequency of selected encodings of a project, many, depending on the research question, quite reasonable quantitative analyses are possible. In any case, the starting point is counting the frequency of codes using the "Count codes" function in the "Retrieval" submenu. The result is shown in a simple frequency list of the counted codes and can be saved as a text file (*.txt). Automatically it is saved in a table structured by codes and files in the format

*_DT.csv in the subdirectory ..\res (_DT stands for "data table"). These tables can be used without problems in the module "AQUAD_eda.exe" for exploratory data analysis or exported to programs for spreadsheet or other statistical analysis.

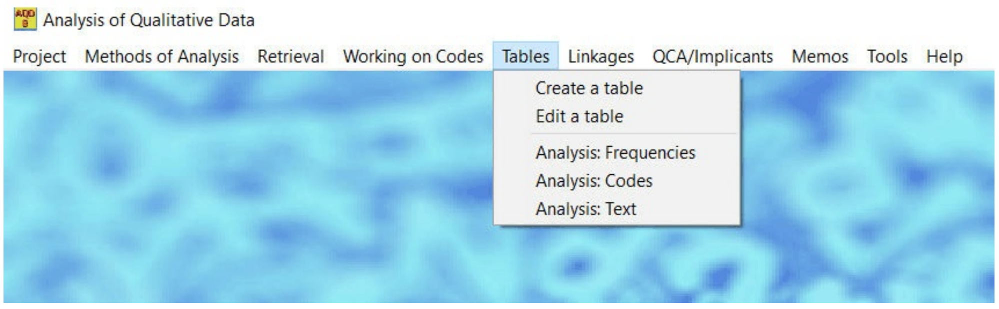
3.2.4 Working on codes



To edit codes, the "Metacodes" function can be used to combine meaning-related categories into a single code. To certain codes (e.g. to a code "interviewer") it is also possible to have additional codes added automatically afterwards (e.g. the control code "$do not count", which excludes the coded text segment from word counting). Unsuitable codes can be deleted from all files or replaced with a more suitable (or correctly written) code. When coding, AQUAD keeps a record in a code register of which codes have been used in a project so far and how often. If errors occur here, for example, by direct editing of the code files outside AQUAD, one can delete the code register without worrying and have it automatically restored in improved form with the help of the available code files.

3.2.5 Tables

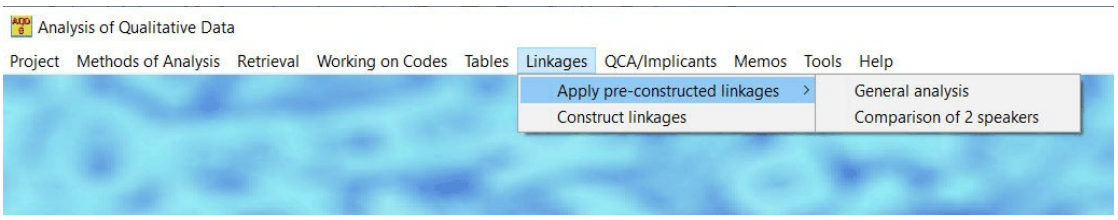
In this module, one can compare how certain codes occur in files characterized by different "profile codes", i.e., the table function compares certain meaning segments under the condition of other codes specific to the files.



Suppose that in the analysis of interviews the gender of the speakers was recorded, and we want to compare what the interviewees said about work and leisure. With the codes "/female" and "/male" we could now define the columns, with the codes "work" and "leisure" the rows of a 2x2 matrix.

For the first of three possible analysis functions ("Frequencies") for a quick overview only the frequency of the codes in the cells of the matrix out. The second analysis function ("codings") gives the locations of the codes in the files, the third analysis function ("text segments") gives in detail also all text segments, which satisfy the respective condition combination, i.e., the four cells of the matrix of our example will be filled with segments of the texts, in which men speak about vacations, then about work, afterwards we find expressions of women first to vacations, in the next cell to work.

3.2.6 Linkages



This is one of the two most important modules of AQUAD for theory building. One can use it to identify significant systematic relationships between file sections. Or, in other words, AQUAD supports interpretation according to the coding paradigm not only in categorizing file sections and ordering data by categories, but it further facilitates discovering relationships between categories. One formulates expectations that relationships between categories that are considered typical or significant will occur systematically. If one suspects such relationships, one must also be able to test the data for them; or to put it in the words of Miles and Huberman (1984), Shelly and Sibert (1985), and Shelly (1986), one must be able to test one's hypotheses. This is what the "linkages" module in AQUAD is for. A positive result, i.e., a finding that certain combinations of statements actually occur systematically in a data set, would be considered evidence for the hypothesis.

Of course, one must first form a hypothesis about the regular succession of two or more coded units of meaning, e.g., "Whenever interviewees mention topic A, they come up with B or C." The program then checks whether, how often, and where such a presumed linkage of codes exists in the data.

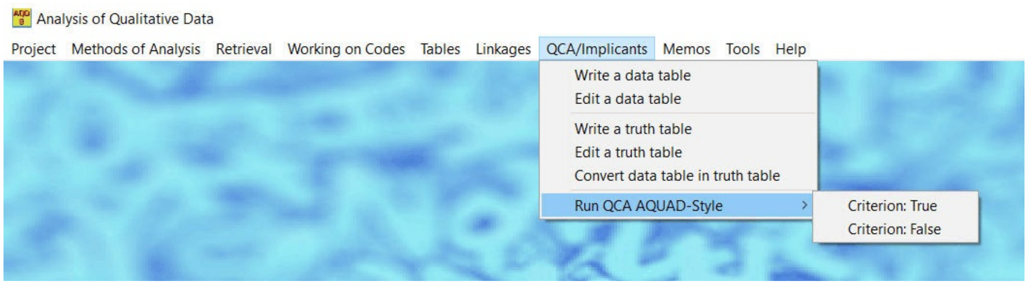
One can select predefined linkage structures and use one's own codes as "variables" or construct linkages of up to five codes oneself using the logical operators "and", "or" and "not".

The check for such links can be performed in general, without further conditions, or can be limited to the comparison of the file sections of one speaker with the file sections of another speaker.

3.2.7 QCA/Implicants

AQUAD provides even greater help in making conjectures about conditional relationships in the data. Such hypotheses are generally considered so closely related to experimental and statistical procedures that they have been largely avoided in qualitative research. Ragin (1987) reminded us, however, that the analysis of causal relationships has a venerable tradition within qualitative analysis that can be traced back to John Stuart Mills (Ragin 1987, p. 36f.), and that qualitative methods may even be superior to quantitative causal analyses in some respects if generalizability is not given priority over complexity (Ragin 1987, p. 54). A detailed discussion of Ragin's argument is beyond the scope of this manual, but his methodological approach, the so-called "Boolean Method of Qualitative Comparative Analysis" or "Boolean Minimization" is explained in a separate chapter of the manual. This qualitative comparative method integrates-not simply combines-characteristic features of experimental and interpretive designs by treating the existence of a particular "condition," i.e., the presence of a particular category in a data set as a dichotomous categorical variable or "truth value." The available information about the event or "condition" is reduced to the value "true" or "false," i.e., whether the event exists or does not exist in the given data set. Causes are always viewed as complex combinations of conditions associated with a specific "outcome". All data are examined for the presence or absence of all types of possible combinations. The results, i.e., presence or absence of a condition, are entered into a table, with each cell occupied by either a zero or a one.

Through the use of algebraic procedures developed by mathematician George Boole (1815-1864), known as "combinatorial logic" or "minimization" and the use of "principal implicants," inferences are drawn from the table about the various combinations of conditions in the data at hand.

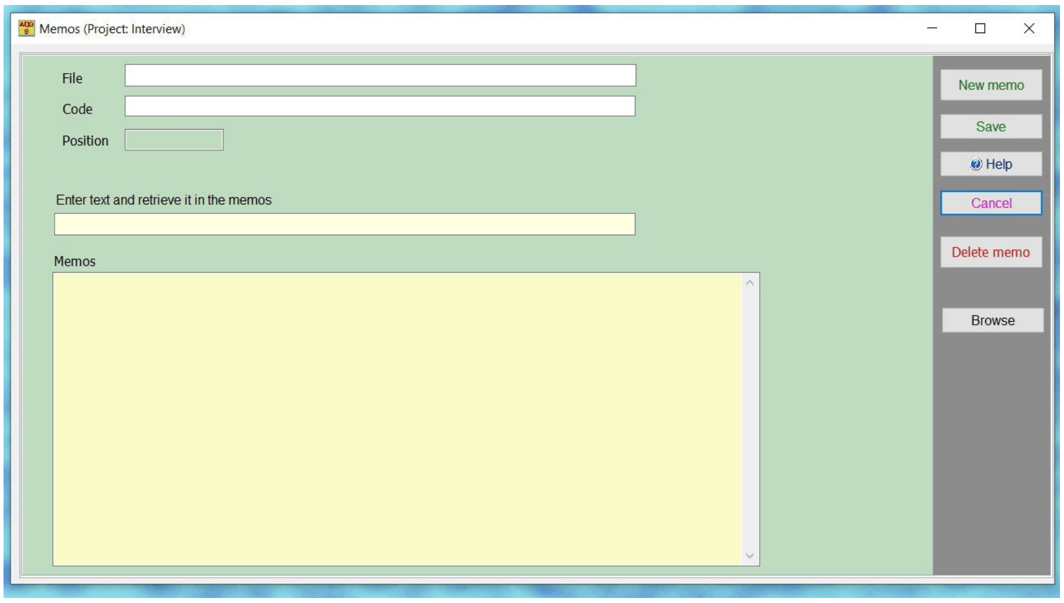


Briefly summarized: This module uses the Boolean minimization procedure to compare the presence or absence of a number of interrelated codes, where it is suspected that one of the codes is critically related to certain configurations of the others. A sufficiently large number of cases should be available for the result of such a comparison to make any sense.

The data table needed as output can be written and edited manually or it can be generated by the program from a table of code frequencies (see module "Edit codes") and converted into a table of truth values (yes/no, true/false, 1/0). However, one can also create or edit this table of truth values manually.

For the analysis, one of the hypothetically linked codes is selected as a positive or negative criterion. The program then examines all cases where the criterion occurs with the value "true" or "false" and gives the configurations of the other codes found under this condition.

3.2.8 Memos

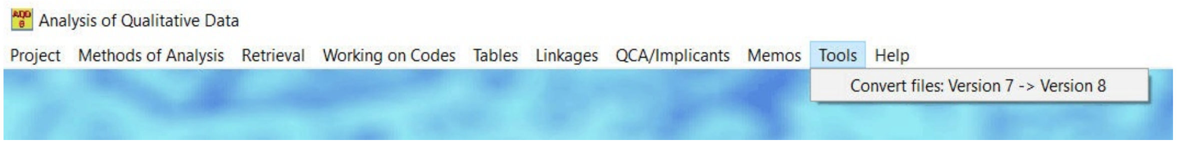

Memos (Project: Interview)
File
Code
Position
Enter text and retrieve it in the memos
Memos
New memo
Save
Help
Cancel
Delete memo
Browse

AQUAD offers two possible ways to access its memo functions, depending on the modules you are currently working with:

1. there is a "Memos" option in the main menu, and
2. there is a "Memo" column on the left side of the coding table (in the "Content Analysis" options in the main menu).

In the Memos module, following Glaser's (1978) recommendation, you can immediately record what comes to mind about codes and coding, what connections, contradictions, concerns, exceptions, etc., come to your attention while you are busy interpreting a file and cannot immediately follow up on that idea. Faced with the alternative of forgetting perhaps extremely important ideas, or losing the thread in the interpretation of a file due to a prolonged interruption, taking short-term memos is a good compromise. AQUAD allows you to tag these memos with the number of the file, the location of a relevant segment, and perhaps important code so that you can search specifically for the memo later. At least one remembers one or the other keyword in the memo text later; therefore a free search for words suspected in the memos is also possible.

3.2.9 Tools



The Tools module offers users of the previous version AQUAD 7 the possibility to take over files, codings and memos and to edit them further in AQUAD 8, i.e. you can use this module to convert your old files into the modified format of AQUAD 8.

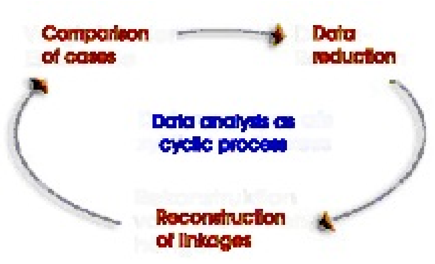
Other tools, e.g. for collaboration in content analysis in groups or for calculating picture elements when working with the graphics module, the author will be happy to take over from version 7 and adapt them to version 8 if desired.

Chapter 4: The Process of qualitative content analysis

This chapter tries to give an overview of typical phases in the analysis of qualitative data on a few pages. For a better understanding, individual steps are highlighted as characteristic for these phases, even though in reality the analytical process always runs in cycles, i.e. one is temporarily engaged, at least "in the back of one's mind", in activities that characterize other phases. Further, this chapter attempts to outline how to take advantage of the possibilities that become available with AQUAD. Details of the practical approach will then be explained in more detail in later chapters. More than an initial overview of the methodological approach to qualitative analysis and possibilities of computer use, this manual cannot provide here. For more detailed presentations of the approach of qualitative analysis, interested readers are recommended to consult the following volumes:

- General contributions on the use of the computer in qualitative research can be found, for example, in Tesch (1990), Huber (1992), Kelle (1995), Fielding and Lee (1998), or Lissmann (2001).
- Miles and Huberman (2nd ed. 1995) provide a detailed introduction to interpretive analysis of qualitative data. They draw on numerous examples. Other specific examples of a variety of qualitative studies from educational research can be found in Bos and Tarnai (1998) and Schratz (1993), and from psychological research in Kiegelmann (2001, 2002, 2003).
- A brief and clear overview of the basic principles of qualitative content analysis is given by Mayring (2012).
- An introduction specifically to techniques of theory-constructing qualitative analysis is given by Strauss and Corbin (1990).

The following overview is a highly abbreviated adaptation of a separate contribution in Huber (1992); for details and further examples, please refer to that volume.



Typical phases in the analysis of qualitative data can be distinguished as the reduction of the original data, the reconstruction of contextual relationships, and the comparison of findings with the goal of generalization. Especially in psychological studies the inductive conclusion of data of a person to regularities of his experience and behavior as a generalized finding is often of interest. Whether commonalities can be found across several persons or observation situations is also of interest, but only at a later stage of investigation.

- The first work phase consists in reducing the amount of data and at the same time the variety of expression contained therein, i.e. the linguistic formulations in texts and additionally the non-verbal forms of expression in video recordings. For this purpose, more or less extensive data segments (text, audio tape or video segments or image excerpts) are determined, to each of which definable meaning is attributed. Thereupon a code is attached to the data segment as an abbreviation or designation of the meaning. In the following these codes are used as representatives of the data segments or the meaning units of the files. Basically, this is a categorization process in which the exact categories usually emerge only during the interpretation. However, they can also be taken from an already existing category system. This depends entirely on the epistemological orientation of the researcher.

- In the second phase, one reconstructs from these units of meaning the subjective systems of meaning of the data producers, i.e., the interview partners, diarists, observers, and so on. In the reconstruction, one looks for regular links of meaning units within the data files that are characteristic of the data producers and/or their situation.

- Finally, in the third phase, invariances or general connections are inferred by comparing the insights into individual meaning systems obtained in this way (cf. Ragin, 1987).

It is important to note that these phases are not strictly delineable and linearly sequential, but tend to overlap and cycle (cf. Shelly & Sibert, 1992).

Already during reduction, for example, one thinks about implicit theories of the data producers, one starts to compare constantly after reviewing a few files; in doing so, perhaps an aspect emerges clearly in person B/file B that one had overlooked in person A/file A - so one goes through the reduction phase again at least partially there, and so on. In each of these phases, it is always important to deductively reassure the validity of the categorizations by inferring particularities of the preceding analysis phase and searching for corresponding evidence.

4.1 The reduction of qualitative data

The principle of reduction is seemingly simple, but its application proves to be very labor-intensive, time-consuming, and error-prone: extensive verbal material must be reduced to the units of meaning it contains.

If the source data for the analysis is available as video or audio recordings, the question arises whether to first make the data accessible for detailed analysis by transcribing it, i.e. converting it into a text version. AQUAD 8 helps to save time and money, as the software allows direct reduction of the data to encodings without the detour of transcriptions. For critical content sequences and their subsequent presentation in reports, essays, etc., it is advisable to transcribe them in excerpts. This can be done using the "memo" function in AQUAD.

The critical question in this initial phase of qualitative analysis is instead: How and where to find units of meaning in the files? For beginners to qualitative analysis as well as when learning new content areas, this question arises again and again. Weber (1985) refers to six common ways of determining units for text files in general, namely to choose words, word meanings, sentences, themes, sections, the whole text (possible e.g. for headlines, summaries, short letters to the editor, etc.) as units of analysis. However, these determinations cannot be made mechanically; they themselves require qualitative preliminary decisions. In the case of videos, additional decisions are necessary for the non-linguistic, non-verbal information, but also on the level of individual words, but quite obviously in the case of the more complex alternatives such as word meanings, sentences, etc. as analytical units, there must be a prior realization or hypothesis that the selected unit is relevant in the sense of the research question. Further, qualitative additional decisions are necessary, for example, about which words are used synonymously or which idiomatic expressions also have the same meaning for the producers of the texts to be analyzed.

With the determination of the unit of meaning, an essential difference between quantitative and qualitative approaches has been established since the beginnings of a broad social scientific reception of text analytic procedures. In the same year, 1952, seminal articles were published by Berelson and by Kracauer. While for Berelson content analysis serves to systematically, objectively and quantitatively capture manifest communication content, Kracauer's qualitative content analysis attempts to reveal the hidden, latent categories of meaning in the text, independent of the manifest text content. Both authors take up opposing positions in a debate that Thomas and Znanecki (1918) had initiated 35 years earlier with a now classic sociological analysis of the letters of Polish emigrants from America. The question of the appropriate analytical unit-words vs. meanings-is obviously confounded with the objective of textual analysis.

At the procedural level, the two approaches are not fundamentally mutually exclusive. Especially with computer support, one can use different procedures of the quantitative approach for qualitative text and image analyses with profit to support one's own interpretation efforts.

Basically, for the selection of units of meaning in qualitative data analysis, one should always keep in mind that regardless of whether one wants to understand people's experience and actions from their point of view, i.e., from their own verbal representations, or whether one wants to explain certain of their recorded actions by tracing them back to the frame of reference of these people's subjective theories, one must infer the analytical units during the interpretation of the data. Only this inductive approach guarantees that one finds access to the subjective worldview of the interlocutors or the observed and that not only some partial aspects of it, possibly detached from the subjective context of meaning, get stuck in the analytical grid of the researcher. This strategy corresponds to Glaser and Strauss' (1979) approach of "grounded theory".

However, this approach requires considerable work as soon as one has more than a single data set to analyze. Since one wants to make the individual results comparable at a more advanced stage of the analysis, one has to examine - and usually several times - the meaning units and their coding in all files for coherence and possibly modify them. Miles and Huberman (1994) have therefore proposed a compromise in which, prior to reading the texts/viewing the files, one establishes a very general, non-content-specific orientation framework for searching for units of meaning, within which specific units are then decided upon according to the conditions of the individual file. This compromise seems justifiable insofar as data reduction also starts at the beginning of an investigation, indeed at its planning.

4.2 Scheme of the data analysis procedure

(1) At the beginning, it is essential to familiarize oneself with the investigated subject area from the perspective of the interviewees or the observed - especially if one has not been present in the interview situation oneself, e.g., if one has not spoken to the interviewees as an interviewer, or if one has not participated in video recordings oneself. In other words, one should not try to develop a differentiated category system already with the first data set. It would certainly only be valid for this current data set and would have to be fundamentally revised already when analyzing the second data set. Instead, it is recommended to either randomly select 8-10 transcripts or recordings from a larger number of data sets or, analogous to case selection in individual case analyses, to selectively pick out a few data sets from the entire data material according to the principle of "theoretical sampling".

For example, if we want to analyze recordings of mathematics instruction at different schools, we might have to expect differences depending on the grade level, school type, and teaching experience of the teachers observed. Using these criteria, we would select a few records and use them to develop an initial orientation. However, we might have observed from the outset in classes where teaching and learning is done according to different methods. Then we would use data sets (also) according to this characteristic for orientation.

This first phase quite obviously modifies Miles and Huberman's (1994; see above) recommendation to settle on a general, non-content-specific orientation framework for finding units of meaning even before reading the texts or reviewing the data files. We recommend here that this frame of orientation, which may be available in outline in advance because of the research question, be checked and differentiated with the help of a data sample.

(2) If we select data files according to the principle of "theoretical sampling", we already have to think in detail about general characteristics of the data sets. We consider what determines the profile of each data set. In the case of the data from "classroom observations", for example, by the age of the students, type of school, professional experience of the teachers, perhaps also by class size, didactic orientation of the teachers, and so on. We should record these characteristics in a memo and later use them as "profile codes" (or "singular" categories, i.e., used only once per data set for its overall characterization) for coding. It is best to insert these profile codes at the very beginning of the file.

If, on the other hand, we simply pick out a few data sets at random for orientation, it would be a good idea to think in advance now about which characteristics of the interviewed/observed persons and the respective situation in connection with the research question of the study could become significant as "profile characteristics" for further analysis.

(3) We first read/view the data selection (see point 1) without elaborate coding and try to understand the overall context, distinguish essential sections, find out commonalities and contrasts, and thus delineate our frame of orientation. It is crucial that we record our interpretations and ideas in memos.

At the end of this step it is advisable to write a short (!) thematic summary, preferably again as a memo. In it we note our "first impression" of what this data set (interview, recording, etc.) is all about, what we consider to be the central meaning of this data set at this early stage of the analysis. In addition, we should outline in a separate memo the - always preliminary - orientation framework.

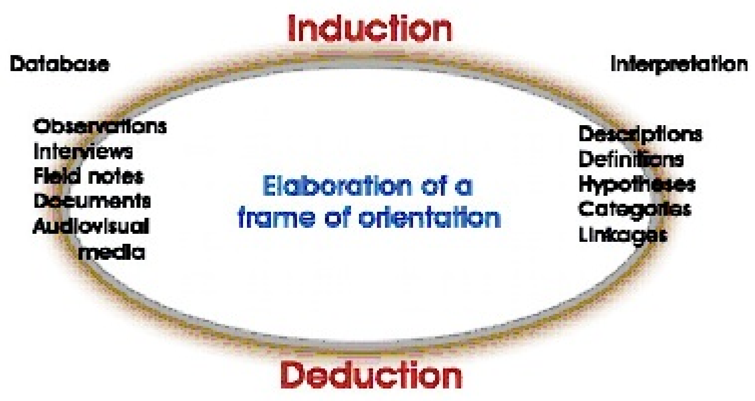
(4) Then we need to make an important decision about our basic interpretive strategy: Do we want to search for the most comprehensive units of meaning possible, as already suggested in point 3 by the thematic summary, code them, and then differentiate them step by step and in repeated passes? Or do we want to pay attention to all details right at the beginning, limited at most by the general orientation framework, mark them with the help of codes, and then summarize the many details in general categories in further coding passes to make the data sets comparable? The first strategy moves from general to particular ("differentiation"), the second strategy moves from particular to general in the data sets ("generalization").

Of course, it also depends on the research question and the researcher's experience with the subject area which strategy is preferable. In general, however, we would like to advise beginners in qualitative data analysis against the generalization strategy, in which they first pay attention to all specific features and then summarize them in further steps into more abstract, general categories. In doing so, one usually successfully keeps oneself from deeper, at the beginning difficult, theory-oriented analytical considerations and runs the risk of indulging in often superficial busyness for a long time. One produces codes, one is active and thus calmed down - even if one then has to realize later that the activities were not necessarily purposeful. In one seminar, for example, a group of students assigned codes for keywords (according to the students) in the text, basically duplicating the data. In another case, a beginner generated about 1500 (no typo!) codes for the interpretation of his interviews and then, of course, couldn't see the forest for the trees.

Therefore, it seems more promising to us, especially for beginners, to take the principle of "constant comparison" in the approach of "grounded theory" (Glaser & Strauss, 1979) very seriously from the beginning and to look for similarities and differences first within the individual, then between the data sets. With this orientation, one is largely assured not to get lost in surface details, but to work single-mindedly toward insight into meaning structures. The following table compares the approaches again:

Strategy:	Differentiation	Generalization
Coding starts with	Search for general categories	Search for spezific aspects
further procedure	Differentiation to uncover specific differences	Generalization to identify commonalities

However, the two approaches are not mutually exclusive, but rather mutually dependent. The interpretation of qualitative data is not a linear but a cyclical process. Shelly & Sibert (1992) have already described this cycle by referring to Dewey's remarks on the connection between inductive and deductive reasoning:

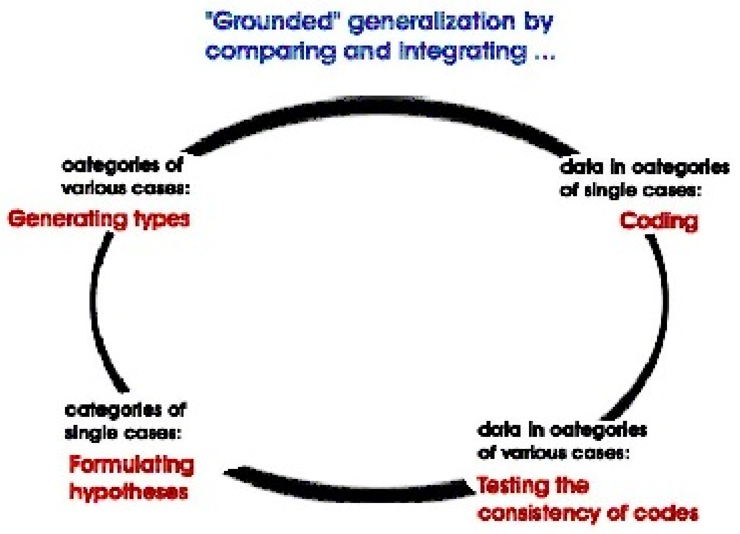


However, the recommendation for beginners to start from general categories has to be countered from this perspective that one has more of a chance to discover something new in the data if one interprets small data segments, line by line, so to speak - but at the risk of overlooking the general and essential.

If one follows our recommendation to approach the data with a strategy of gradual differentiation, the accent is initially on the "interpretative knowledge base." This interpretative knowledge may, of course, have developed inductively from experience in the field or in the initial sifting of the data material, for example, in the form of some very general categories (study in a kindergarten: "The kindergarten teacher directs the children's attention"; "She supports them in difficulties"; "She encourages verbalizations", etc.) or in the form of working hypotheses that were already decisive in the formulation of the research question. Necessarily, the data base differentiates when we analyze recordings from several people and perhaps from different situations.

If we apply the strategy of generalization, i.e., if we start the analysis with specific meanings that appear in the data from an interpretive perspective, signposts from specific interpretations to general interpretive knowledge become particularly valuable.

The basic rule of "grounded theory", the "constant comparison" has been characterized as a guide to generalization. These processes of comparison are also cyclical, as Shelly & Sibert (1992) have pointed out. Adapted from their classification, constant comparison and its goals can be described as follows as the analysis proceeds:


"Grounded" generalization by comparing and integrating ...
data in categories of single cases:
Coding
data in categories of various cases:
Testing the consistency of codes
categories of single cases:
Formulating hypotheses
categories of various cases:
Generating types

(5) After deciding in principle how to proceed with the analysis, one's deliberations are concretized: one begins to code a sample of texts. In doing so, one will necessarily switch again and again between comparing and integrating individual data in the single data set and comparing several data sets on the categories generated in the process. One thus moves in smaller, less expansive cycles of analysis within the cyclical process, initially to ensure consistency of coding.

(6) Finally, one applies the coding rules obtained in this process to the total available files. In the process, one will regularly encounter contradictions and exceptions that prompt modification of the coding rules. In some cases, such insights may make it necessary to leave the cycle of data reduction/interpretation and re-enter the phase of data collection in a broad cyclic movement.

(7) Memos to support hypothesis generation and typing will emerge throughout the process. Grounded in such incursions and insights gained in the course of coding, analysis at advanced stages focuses primarily on elucidating systematic (meaning) relationships among critical categories. This was described in point 4 as integrating categories into "hypotheses"; e.g., "When educator E observes event A or B, she applies action option X." Or, "When teacher L talks about lack of motivation of his students, he mentions the influence of mass media and the role of parents."

Experiences in this phase, for example of contradictions in interpretations, can also be the occasion for an analytic loop back through the stages of coding and consistency checking (see point 4). Repeated data collection with modified questions, observation situations, etc. may also prove necessary.

(8) Assuming a sufficient number of cases, we can finally try to summarize the individual cases into types on the basis of comparison and integration (theoretically justified) of selected categories. As a result, we obtain a distinction of cases according to typical combinations of characteristics.

(9) A general note should not be missed here, which applies to the process of qualitative analysis as a whole: qualitative analysis has something to do with the "quality" of the events and states captured in the data. Thus, as a rule, the codings must not only neutrally capture the fact of the occurrence of certain events, but must also represent the quality of the events. For example, for hypothesizing and typecasting in a biographical interview study with adolescents, it would not be sufficient to simply use a code "parents" to mark the file segments in which they talk about their parents. Which adolescents do not talk about their parents?

A characteristic that occurs in the same way in all of them without variation cannot contribute anything to "constant comparison" and is therefore not suitable for further distinctions and explanations. So we have to qualify the talk about parents from the beginning: How do young people talk about their parents? Few differentiated gradations, e.g. positive / indifferent / negative, are usually sufficient at the beginning of the analysis for the purpose of searching for commonalities and differences.

It is a humanly sympathetic trait if beginners of qualitative analysis hold back with such evaluations, especially of characteristics or statements close to the person, if they avoid rash value attributions ("positive"; "negative") - but it is exactly about qualifications that qualitative analysis is about. Thus, at the latest when checking the consistency of codings and establishing coding rules, "neutral", non-qualifying codes should be examined to see if they need to be cause for going through a backward loop in the interpretation process and comparing within and between files to see if quality differences in the data segments they mark become apparent.

4.3 Developing codes for units of meaning

The following hints should be understood as heuristics, i.e., general hints on how to identify units of meaning and find appropriate codes, but not as rules that lead step by step to the goal. Three strategies suggest themselves. In particular, AQUAD offers valuable support when using the first two options and their variants presented below.

1. when searching for categories, one goes through a file to see whether events or statements about events, situations, persons as well as expressed opinions, ideas etc. contained therein can be attributed to a superordinate concept.
2. in the search for sequences one pays attention to the representation of connections in the events or statements, looks for thus substantially more comprehensive units of meaning than with categorial search.
3. in the search for topics, the greatest abstraction is required; under certain circumstances, an entire data set is reduced to which topic is addressed in it.

Cresswell (2012) provides very instructive examples of qualitative studies against different theoretical backgrounds, namely analyses along the lines of the narrative, phenomenological, and ethnographic approaches, as well as the grounded theory and case study approaches. Saldaña (2016) provides an overview and concrete guidance for developing content-specific codes in analyses according to the coding paradigm.

4.3.1 Development of categorical codes

Should the categories or corresponding codes describe, interpret, or explain (see Miles & Huberman, 1994, p. 56)? Of course, computer assistance is not expected in this decision. Methodologically, the researcher has to decide on his or her own responsibility. The computer only helps to document all decisions, to arrange them, to revise them if necessary without much effort. For the solution of the first problem there are three possibilities:

Use of predefined category systems

In the case that a quantitative study does not require object-based theory construction, i.e. one does not want to develop subjective explanatory systems from the available data, the principle is quickly described: One can then fall back on already available category systems and reduce one's own data according to the interpretive schemes contained in these systems. "Available" are such category systems, for example, from one's own earlier investigations; one also finds them in the empirical literature or in theoretical analyses of the content area of interest.

If one makes use of given category systems, one must decide which sections in the available data correspond to which definition examples for the categories of the system, and then record the location and coding with the help of the computer. Since some utterances cannot be unambiguously assigned, and since inconsistencies occasionally occur in the interpretation process, a search function for specifically coded file sections ("Retrieval"; "Specific Codes") provides great help in controlling data reduction. After entering a critical code, this function quickly and effortlessly returns all assigned file sections in all data sets analyzed up to that point.

However, this form of reliability control of codes initially finds only the data segments erroneously assigned to one category, but not the passages erroneously assigned not to this category, but to another. To detect this error, one could immediately go through at least the set of data segments of similar categories "susceptible" to misassignments. Furthermore, one can - only in the analysis of texts (!) - take advantage of the possibilities of mutual complementation of the definition of manifest and latent units of meaning. Three strategies of varying complexity are used:

(1) One defines keywords as manifest indicators of a critical meaning content and searches for their occurrence throughout the text. One can then judge whether the passage in which a keyword is found should not be assigned to the category of interest.
(2) One summarizes keywords in an analysis lexicon or word catalog, i.e. one enters a list of keywords to search for critical text passages. This gives one the information one is looking for in one pass through the texts.
(3) In AQUAD, the first two options are provided, but are automatically linked to the third option, the key-words-in-context function (Popko 1980). This allows to search for one or more keywords in a row automatically in all texts of a study. The result is a printout of all lines in which the searched word occurs.

Hypothesis-based categorization

When determining and specifying the content of meaning units in the files to be analyzed, one is guided by hypotheses about the content area. On the basis of hypotheses one tries to define categories and coding rules for the data.

For example, Marcelo's (see above) approach to categorizing interviews with beginning teachers was hypothesis-guided. A theoretical model of professional socialization served as a scaffold for designing a preliminary category system. Growing familiarity with the interview texts and increasing insight into the subjective worldview of novice teachers led to some of these categories being discarded as less appropriate, while other categories were first discovered and newly incorporated into the system. As the analysis progressed, some of these categories were linked into larger units in terms of subjective theories of the entrants (cf. Huber & Marcelo, 1992), which in turn had to be tested on the whole material.

It is best to determine concrete categories for each of the hypotheses already when planning the data collection, for example when formulating guiding questions for an interview, as well as before the data analysis (deductive part). One then proceeds with these categories as described above. On the other hand, during the data analysis one will certainly discover segments in the data files that cannot be assigned to the predefined categories. One then marks these units of meaning and inductively develops categories that link them to the subject-specific hypotheses. Of course, modifications of the original orientation framework cannot be ruled out in this procedure. The degrees of freedom and the demands on one's own interpretative performance increase with this strategy, as do the possibilities to do justice to the specific perspectives of the research subjects in the analysis.

Even with this strategy, one cannot do without the "method of constant comparison" that characterizes the approach of object-based theory building (Glaser & Strauss, 1967, 1979; Strauss & Corbin, 1990). As described above, the central process in "constant comparison" is to test each inductive inference from the specifics of the data base to more general principles, here now categories of data reduction, using deductive inferences about the data base.

As the demands on the researcher's interpretive skills grow, so does the importance of computer assistance in this procedure. When checking correlations between categories, one comes up against the limits of simple search functions. From this point on, qualitative analyses require AQUAD, which allows deductive reasoning based on logical programming (Tesch 1990; Shelly & Sibert 1992).

(1) The possibilities for searching coded file segments and, in the case of texts as a database, keywords (including search lexicons and KWIC function) are to be used for consistency checking of codings within individual files and across all files here as well as when using given category systems (see above).
(2) Hypotheses that frame the development of categories provide clues to specific relations of the categories. AQUAD provides functions for hypothesis-based data reduction to test such relations. This can be used, for example, to determine the super/subordinate relationship of individual categories, sequences of certain categories, or clusters of certain categories. The common factor here is that one must keep in mind not only one category or the file segments represented by it, but two or more categories and the defined relationship between them - which can also include negations! Also, of course, the non-satisfaction of the presumption that certain relations between units of meaning exist must be registered.

Theory-constructing categorization

The most demanding form of qualitative data reduction renounces all pre-established reduction rules in the form of category systems and structuring of the analysis process by hypothetical framework concepts. In addition, the researcher must pay attention to his or her own subjective experiences, presuppositions, or prejudices with regard to the experience and behavior of the research subjects and try to avoid premature solidifications by constantly comparing the reduction principles that emerge in the analysis process with the events or statements in the data files. Sensitivity to readings that one puts into the data oneself is particularly necessary because content analytic procedures are generally applied to data that are often evaluated at a great distance from the time and context of their production (Fischer, 1982).

Information about the data producers (acting persons, speakers, or writers) and their context must be obtained from additional data or from the data file itself. This approach most likely promises to bring the researcher closer to the goal of seeing the world of the research subjects through their own eyes and understanding it from their own perspective.

If, in this process, one attempts not only to describe the subjective views of the world but also to organize them with the help of appropriate terms and to reconstruct systematic relationships, then one is striving for what Glaser and Strauss (1967; 1979) have called the discovery of a "grounded theory." Contained in the word "discovery" is the essential difference from methods that seek to confirm given theories. Strauss and Corbin (1990) point out that object-based theories are developed in analytical engagement with the object, so one does not start with the theory to prove it using object-specific data, but with the observed, recorded phenomena that one wants to better understand and explain.

This approach requires the highest interpretive skills - which is why it presents the greatest difficulties for novices. Given category systems contain clear interpretive rules implicitly in the choice of examples for the categories or explicitly in the definitions of the categories; hypothesis-guided categorization can at least rely on general guidance about what to look for in the data. Theory-constructing categorization, on the other hand, must first discover what the data are about, what the texts are talking about. Beginners tend to take as little risk of error as possible and to interpret as little as possible (see above). In extreme cases, this tendency leads to reading through (the mostly used) text data for relevant words, comparable to the use of keywords when categorizing with ready-made category systems. The text passages, in which the critical word is contained, are then marked as finding place of a "meaning unit" and designated with a code. In this procedure, however, the code can hardly represent subjective meaning, but mainly just the fact that a certain word has occurred in the designated text passage.

This tendency is especially pronounced when one does not yet know about the possibilities of tentative interpretation and effortless revision in computer-assisted content analysis. With AQUAD one is not punished for "playing" with one's own ideas by unreasonable revision work, if one finds contradictions later, on the contrary, one is encouraged to interpret, since codes are needed from the beginning to designate text segments and since changes, summaries, differentiations can be made without much effort (cf. Tesch 1992).

As a rule of thumb for theory-discovering categorization one can therefore adopt: search in the data for units of meaning that are on the one hand as large as possible, so that there is something to interpret at all, and on the other hand as small as necessary, so that inconsistent contents are not represented by the same code. This paraphrases the differentiation strategy recommended above. AQUAD supports this strategy because it does not impose any restrictions on the definition of meaning units; in particular, multiple units may overlap and, conversely, one may code the same unit multiple times.

If one is familiar with the subject area addressed in the data and with the method, one can also turn this rule on its head, i.e., start from the smallest units of meaning and follow a generalization strategy. With these assumptions, first, one does not get stuck so easily on details and does not miss the essence of the data. Secondly, one then knows that detail coding can be automatically linked to larger units with the help of functions of meta-coding with computer support. By constantly comparing the coded file segments, one then inductively determines the feature dimensions they circumscribe and, finally, the overarching categories to which they belong. AQUAD also supports this strategy, as one can very quickly search from a specific coding for contrasting, similar, dependent, or otherwise defined relation to that coding data segments. Based on these relation(s), one can then attempt to subordinate the partial codings to a higher-level category.

4.3.2 Development of sequential codes

In categorical reduction, we attempt to distinguish segments of the files from one another in such a way that clearly distinguishable, mutually exclusive meanings can be ascribed to them. However, the clear assignment of meanings to categories does not imply that file segments must also be distinct from one another. Depending on the expressive habits of data producers and readings by researchers, file segments assigned to different categories often overlap. In extreme cases, the same segment may be assigned to more than one category.

It is at such linkages in the data that the strategy of sequential reduction comes in. It looks for a specific connection of meanings, marks the occurrence of such connections and marks appropriate file passages with a code - as with categorial reduction. However, the code here represents a defined sequence of meanings. The unit of meaning in the file in this case is the entire passage in which this link is described.

Now, beyond the differentiation of categories and subcategories, what are the strategies for discovering meaning contexts in the data? In the following, we distinguish between strategies of searching for simple sequences and defining and searching for complex sequence patterns.

Searching for simple sequences

In addition to the search for hierarchical sequences of superordinate and subordinate categories, which Strauss and Corbin (1990) recommend for sequential reduction, a whole range of other simple sequences suggest themselves with regard to grammatical-linguistic systematics. Which of these are useful for data reduction depends, of course, on the research question. Texts are often searched for causal sequences (also with keywords like "due to", "because", "for", etc.), temporal sequences ("while", "then", "before", etc.), concessive sequences (positive constraints like "not ... however" or negative constraints of the type "indeed ... but"), conditional sequences ("if ... then"), final sequences ("so that", "for the purpose of", "in order to", etc.), comparative sequences, modal sequences, and precis sequences.

Neither the enumeration of these sequence types nor the possible keywords listed here claim to be complete. They are only meant to stimulate own attempts to reduce the data according to meaning sequences according to the research question. In this context, the function of the computer or software should be emphasized as a useful tool for interpretation, but not as an agent of qualitative analysis. As helpful as word search functions can prove to be in this approach, their applications are limited. Unless one is analyzing carefully worded documents, the yield of search lexicons for specific sequences of meaning is usually limited. In a free interview, it is often difficult to determine sentence structures. Where is the main clause, where the subordinate clause? Accordingly, the conjunctions critical for defining the sequence may be thought of, but not spoken. Moreover, many speakers' use of critical conjunctions and phrases is not grammatically sound. Thus, searching for key words or whole lexicons of such words does not replace empathetic interpretation, but it can provide a useful heuristic in this work.

Searching for complex sequence patterns

When interpreting data, one can be inspired by structural or processual characteristics of the object under study and look for more complex sequences of meaning than outlined above. For example, for the reconstruction of subjective theories of action of the data producers, one could look for sequences of situation assessments, reflection on alternative actions, attribution of expected outcomes, and assessment of personal consequences such as satisfaction or disappointment.

4.3.3 Development of thematic codes

The most radical approach is taken when one attempts to reduce the coding of a data file to a central category, i.e., its message or central theme. Because quantitative analyses proceed as cyclical processes, one cannot assign a fixed location to thematic reduction, such as at the end of the analysis to summarize findings. While condensing the data into a central category serves an important function at this point, the approach is also very useful in other placements:

(1) For relatively short and homogeneous files, thematic reduction can be at the beginning of the data analysis process.
(2) For long, heterogeneous, cluttered files, the thematic approach can be used to begin the data reduction process. In this use, thematic reduction performs important heuristic functions. One first tries not to be confused by the multitude of perhaps contradictory details and to filter out the main idea(s).
(3) Usually, thematic reduction comes at the end of the sequence of analytic steps, often only at the end of several cycles of data reduction, reconstruction, and comparison of meaning systems in which the "leitmotifs" of different files have been gradually worked out (cf. Shelly & Sibert 1992; Strauss & Corbin 1990).

Strauss and Corbin (1990) recommend five steps for thematic reduction. This sequence is compatible with each of the three described placements of the approach in the analytic cycle, and it helps to better assess the contribution of software features:

(1) one begins by identifying a central idea, a central category;
(2) then one looks for subordinate categories;
(3) often it will be necessary to differentiate or link these categories;
(4) hypothesized relations between subcategories and between them and the subject are checked against the data;
(5) inconsistencies, e.g., inconsistent categories, give rise to further cycles of analysis.

4.4 Reconstruction of meaning systems

Theory-constructing qualitative analysis is about inductively generating the theory about the subject matter from the events or statements in the data that is subjectively available to the originators of the data. We speak here of the reconstruction of systems of meaning. The problem of how to find the subjective links can be attempted to be solved inductively, deductively, or by combining inductive and deductive strategies. Which approach one prefers depends primarily on the research question.

For example, when analyzing recordings or transcriptions of little-structured interviews or free-form texts, e.g., diary entries, one will generally begin by identifying individual units of meaning and assigning the corresponding file segments to specific categories. Then one will look for typical sequences of these categories. Finally, one tries to subordinate such sequences to more abstract categories, to the speaker's theme or themes. In doing so, one starts inductively, but alternates between inductive and deductive inferences at the latest at the thematic reduction.

If one approaches the data with a specific epistemological interest or a certain theoretical orientation, one will look for certain correlations from the beginning. Thus, one starts from hypotheses about possible relationships and tries to prove them deductively on the basis of the data. Discrepancies and failures give rise to inductive analysis sections, with which again the hypothetical orientation system can be modified.

In the inductive procedure, one strives to generalize categories and their systematic relationships from the data. In the deductive approach, one looks for specific file segments that can confirm general presuppositions or hypotheses about the relationship of the categories. The following overview of the computer-assisted reconstruction techniques available in AQUAD is structured according to this scheme.

4.4.1 Reconstruction of simple conditions

Depending on the progress of the analysis process, two variants can again be distinguished:

(1) Search for simple sequences of meanings. One defines a data range for the search before and after the data segments of a selected category, i.e. one determines how many lines or counter units (frames for videos, tenths of seconds for audios) are to be searched before and after the data segments of the selected encoding. Within this data range, the occurrence of other encodings is then to be registered. If certain encodings occur frequently in the neighborhood of the critical code, a possible systematic behind these combinations can be checked. In AQUAD, the Search for "Coding Sequences" option in the Search module supports this reconstruction approach.

(2) Search for conditional meaning relationships. The "condition" for the possible presence of systematic associations is defined by (at least) a second category (in at least two meaning variants, e.g., "gender") to which a data segment must be simultaneously assigned. For computer-aided retrieval, one constructs coding matrices (Miles & Huberman, 1994) or coding tables (Shelly & Sibert, 1992). The twofold determination of the contents (data segments) of the cells of such a table determines the interpretation of the findings more strongly than the mere finding of an accumulation of spatiotemporal proximity of initially independent categories. On the other hand, the construction of an analysis matrix requires more conceptual preliminary work, and thus greater progress in the analysis process. The module "Tables" is available in AQUAD for this reconstruction approach.

4.4.2 Reconstruction of conditional relationships

Again, two variants can be distinguished, corresponding to the simple and complex sequential coding strategies described above:

(1) Testing simple coding sequences. Suppose that from a parent interview about family child-rearing practices the impression emerges that a father makes a special effort to justify his actions, then one could use the "linkages" module to search specifically for sequences of relevant categories, e.g., for final or causal coding, and thus test the assumption.

(2) Testing complex coding patterns. For checking coding relations that go beyond the level of complexity of the linkage structures specified in the "Linkages" module, an "open" program component is available in AQUAD for formulating complex, thus case-specific, linkages of codings.

4.5 Comparison of meaning contexts

Constant comparisons of interpretations within a data set (text, video, audio recording, image) and between different data sets form the core of qualitative analysis procedures (cf. Shelly & Sibert 1992). Already in the case of data reduction within a data file, one does not achieve reliable coding without comparisons of the categories or the data segments both among themselves and between the other data files. Now, in most research one wants not only to capture the uniqueness of each data set or the opinions, competencies, subjective world views, etc. expressed in it, but to capture it with a category system valid for all files. At some point in the research process, one wants to tease out general configurations across files. Thus, in qualitative analyses, one must deal with the proverbial danger of often not seeing the forest for the trees. Above the differences and their specification, one must not lose sight of what is common. To do this, one must recognize and compare systematic connections across files.

However, the problem of comparing correlations in social phenomena is that a multitude of conditions are often found in diverse, sometimes even contradictory combinations. Finally, in many studies, one faces the additional task of having to compare one's own findings with other studies, i.e., basically having to conduct a qualitative meta-analysis. Even if other researchers have worked on comparable questions, however, they will have found only partially consistent answers in the process. In the search for conditional relationships for a given phenomenon, a wide variety of constellations are demonstrated across different studies in all sciences that deal with the complexity of natural systems.

Let's take a question often bitterly debated by opponents in everyday life as an example: "Is there a connection between lung cancer and smoking?" Those who want to answer this in the affirmative are regularly confronted with the reference to their 80-year-old grandfather who had enjoyed the blue haze since early youth or with the reference to a victim of lung cancer who had never smoked. Obviously, many conditions play a role. Empirical studies provide a variety of empirical arguments from which one can pick out the appropriate ones for argumentative immunization, depending on personal interest or one's own consternation. Only a comparison of the findings of all relevant studies or at least a representative sample of them can provide clarity about relevant constellations of conditions.

Ragin (1987) has opened up an explicit comparative procedure by applying Boolean algebra to qualitative data. The procedure is based on the Quine-McClusky algorithm of "logical minimization." This procedure fulfills the requirements of the sought comparative method to a great extent, at least the application of Boolean algebra creates ideal conditions. According to Ragin (1987, p. 121) it is possible to fulfill the requirements

- to compare a large number of texts or individual cases;
- to capture complex links of conditions;
- if desired, to produce "parsimonious" reconstructions;
- to examine individual texts both in relevant parts and as a whole (cf. thematic reduction);
- to compare competing reconstructions.

If we compare this procedure with approaches oriented to variables, which start from the basic assumption of additive aggregation of individual variables, we can highlight that case-oriented comparison is designed to do this (cf. Ragin 1987, pp. 51 f.),

- to bring out invariances or constant relations of meaning by careful comparisons of individual cases;
- to pay more attention to the variability of meaningful conditional configurations than to mere frequency distributions of typical cases, from which it follows that even a single contradictory case must be taken into account;

- to consider cases as wholes, i.e. to see the conditions of the case in interdependence that constitutes this particular case, but not the individual conditions in dependence on a population distribution;
- to examine precisely how the overall conditions tapped lead to different findings in different contexts and in different configurations.

In order to apply the rules of Boolean algebra to data interpretations, one radically reduces the meanings found in each file to truth values. Thus one is content with the binary notation "condition is true" (i.e. given) or "condition is false" (i.e. not given). It does not matter whether one is dealing with genuinely qualitative conditions, i.e. interpretations and the assignment of suitable categories, or with measured values of a quantitative characteristic.

AQUAD offers the possibilities of logical minimization in the module "Implicants". When comparative operations become necessary in the process of qualitative analysis - and this can be at any stage due to the cycles of analysis - one should also think of the possibilities of logical minimization.

As a heuristic, it already performs well in the generation of categories for interpretation, even if only a few files have been analyzed, so the basis for comparison is still narrow. Towards the end of the analysis, when summarizing the results or when one wants to group individual findings, distinguish types of files or speakers, highlight key texts, etc., the procedure of logical minimization seems indispensable. This is because, as Ragin (1987, p. 51) notes, with each additional file the amount of work increases geometrically, but with each additional category or condition to be considered in the comparison it increases exponentially! The grouping or clustering of individual cases using logical minimization has been particularly extended in AQUAD.

Finally, the logical minimization procedure can be used when one intends to compare several qualitative studies. This is the methodological step to which the term "meta-analysis" is usually applied. When the research results have been expressed quantitatively, one can resort to a number of highly formalized procedures for meta-analysis. In the field of qualitative research, the logical minimization approach can bring the desirable simplification, transparency, reliability, and documentability to meta-analytic comparisons.

Chapter 5: Text analysis as sequence analysis

The goal of sequence analysis is to establish a case structure hypothesis by hypothesizing and testing text segment by text segment about their meaning. The case structure enables general statements beyond the individual sensitivities of the actors, since they have to orientate themselves for their communication and interaction on a linguistic-cultural common system of rules. Explanations of content in this chapter and two examples are taken from Gürtler, Studer, and Scholz (2010) and Studer (1988), respectively. The module "Objective Hermeneutics" for sequence analyses in AQUAD divides work with texts into three sections:

(1) As a prerequisite for sequential analysis, texts must be divided into text segments, to which subsequently
(2) hypotheses about all their conceivable meanings are developed, before
(3) these are later subjected to a critical examination at further text passages.

5.1 Preparation of the texts

In sequence analysis, the interpretation follows the sequence of events or interactions exactly as they are recorded in the text. Wernet (2009, p. 27) emphasizes the importance of the "basic interpretive attitude" according to which the "text is to be taken seriously as a text" and one must not "take it as a quarry of information or as a fairground of offers of meaning" or even "cannibalize" it. This procedure is conditioned by the goal of sequence analysis, namely, according to Oevermann et al. (1979), to identify in texts as protocols "of real, symbolically mediated social actions or interactions"

(p. 378) the *latent meaning structures* of the interaction and to distinguish them from the manifest content: "interaction texts constitute objective meaning structures on the basis of reconstructible rules, and these *objective meaning structures* represent the *latent meaning structures* of the interaction itself" [italics in original]. They "are reality (and endure) analytically (if not empirically) independent of the respective concrete intentional representation of the interaction meanings on the part of the subjects involved in the interaction" (ibid., p. 379). That is, we know nothing about the intentions of other subjects, but we can reconstruct their motives from textual protocols.

According to this position, the general meaning of a text, i.e., an interaction protocol, is not revealed by trying to fathom the motives, intentions to act, normative ideas, etc., of the participants, "the inner-psychic reality of subjects of action" (Oevermann et al., 1979, p. 379), but by trying to reconstruct its objective structure of meaning. This structure of meaning exists independently of the states of mind, perspectives, and intentions of the agents as a "social reality," as a "reality of possibilities"(Oevermann et al., 1979, p. 368) that are available through the given social system of norms and rules - or can also be inferred from a violation of the rules. It is therefore essential in sequence-analytic text interpretation to distinguish clearly between the level of objective meaning structure and the level of subjective meanings. Most importantly, the objective meaning structure or the latent meaning structure of interactions must be examined independently of and before taking into account the subjective meanings that are accessible to the participants. Oevermann et al. (1979, p. 380), referring to George Herbert Mead, specify that they start from a notion of meaning as an objective social structure that appears in interaction-and "which in turn must be regarded as a precondition for the intentionality [of the participants; author's addition]." On the other hand, this means that the self-understanding of the interacting subjects, their intentions and motives do play a role for interpretation, but only against the background of the latent structure of meaning of their interaction - i.e., after the analysis.

Oevermann et al. (1979, p. 354) consider this approach "a very simple perspective, arguing almost with trivial assumptions." However, strict methodological consequences follow from these assumptions. In particular, it should be noted "that no information from and observations of subsequent interactants may be used to interpret a preceding interactant" (Oevermann et al., 1979, p. 414). Or, as Wernet (2009, p. 28) explicitly points out, "One does not wander in the text looking for useful passages, but follows the textual protocol step by step." And further, "For sequence analysis, it is decidedly important *not to pay attention* to the text that follows a passage to be interpreted" [italics in original]. This does not imply that one must interpret a text from the first sentence or omit any passages, but passages selected as relevant to the research question may only be interpreted sequentially. For better understanding, a repetition: the purpose of sequential analysis is to examine a structure that unfolds naturally in exactly the same way; but for this purpose, the chronological order of events in the textual record is central.

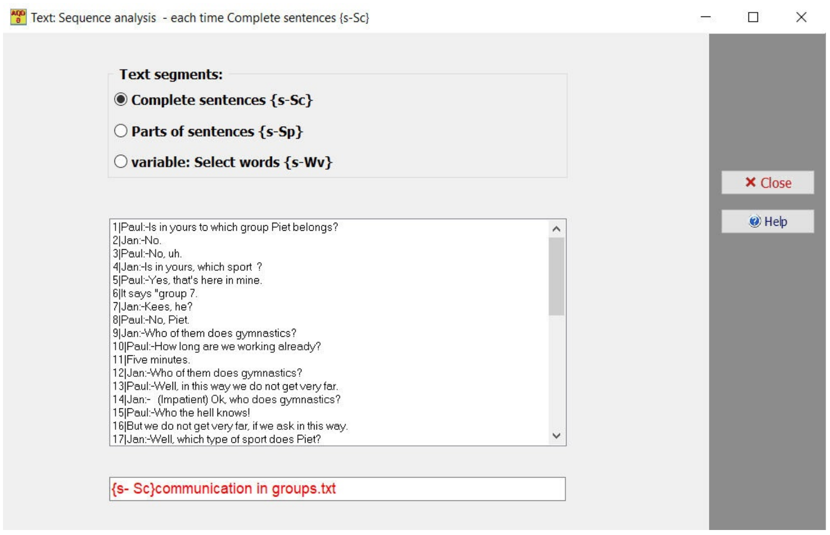
As a first step, therefore, the text must be divided into sequence units, into successive text segments, for the interpretation work on the computer. In AQUAD one can

- the text according to sentence parts (up to the next punctuation mark (,.;:?!) or
- by complete sentences.

It is also possible to

- to divide the text into text segments word by word.

In the following example the text was divided into complete sentences. In the renaming of the sequentially structured texts, the name of the original text (example in AQUAD) "Communication in groups.txt" is therefore preceded by "{s-cS}", which is intended to signal that here is a text sequentially structured according to complete sentences: "{s-cS}Communication protocol.txt". Correspondingly "{s-Sp}" stands for the structure after dividing the text into sentence parts, "{s-Wv}" for text segments just composed of variable number of words.


Text: Sequence analysis - each time Complete sentences (s-Sc)
Close
Help

Text segments:
[✓ Complete sentences {s-Sc}]
Parts of sentences {s-Sp}
variable: Select words {s-Wv}

```
1|Paul:-Is in yours to which group Piet belongs?
2|Jan:-No.
3|Paul:-No, uh.
4|Jan:-Is in yours, which sport ?
5|Paul:-Yes, that's here in mine.
6|It says "group 7.
7|Jan:-Kees, he?
8|Paul:-No, Piet.
9|Jan:-Who of them does gymnastics?
10|Paul:-How long are we working already?
11|Five minutes.
12|Jan:-Who of them does gymnastics?
13|Paul:-Well, in this way we do not get very far.
14|Jan:-  (Impatient) Ok, who does gymnastics?
15|Paul:-Who the hell knows!
16|But we do not get very far, if we ask in this way.
17|Jan:-Well, which type of sport does Piet?
```

{s- Sc}communication in groups.txt

5.2 Generation of hypothesis

5.2.1 On the theoretical background

"The concrete object of the procedures of Objective Hermeneutics are protocols of real, symbolically mediated social actions or interactions, be they written, acoustic, visual fixations combined in different media or archived in other ways" (Oevermann et al. 1979, p. 378). In these protocols the latent meaning is to be found - this is the goal of Objective Hermeneutics. In the introduction it has already been outlined that, according to Wernet (2009, p. 39 ff.), this is done by following a sequence of "telling stories" (stories in which the text segment might occur), "forming readings" (i.e., comparatively ordering the stories in various versions according to similarities and differences), and "confronting these readings with the actual context" (systematically comparing them).

It is not only people's unique experiences that must be considered, but rather their objective realities, which can be tapped as individual, social, and culturally embedded patterns of action. The strict orientation to what is, to reality, guides the procedure in Objective Hermeneutics. The starting point is the general (normal) case, which becomes public and generally accessible through the reconstruction of the latent structure of meaning. After that, the concrete manifestation, the form of expression can be interpreted more precisely by the empirical person and his subjectivity.

To illustrate this, we would like to give an example taken from the application letter of a potential client for addiction therapy (see Studer, 1996). A common formulation of therapy motivation to be found there reads:

"I am still very interested in getting a grip on my addiction."

We can read this text segment as "I want to get free of addiction" or "I want to be able to live abstinently," but also "I want to be able to control my drug use" – because what you have "a handle on", you don't let go of. Further analysis of the application letter (see Studer, 1996) shows that in the case of this potential client, the initial motivation "to get a grip on addiction" is aimed at control, but not complete overcoming of the addiction. The aim of control is to prevent harm to the client, if possible. The knowledge of latent meaning structures reconstructed in this way brings the invaluable advantage for practice of being close to the reality of the client, which has an impact on her actions and determines her everyday life. First of all, their motivation expressed at the moment to change their previous way of life must be taken seriously and appreciated. However, this must be followed by realistic exploration of the scope of possibilities in order to be able to drop therapeutic illusions in good time and to adapt therapy offers individually. To illustrate this, we would like to cite another text segment from another letter of a client (see Studer, 1996):

"Also, however, we [my partner and I] had very good conversations ..."

We can read this segment as another example ("Also we had ...") of the writer's appreciation of her partner. However, we can also read ("Also, however, we had ...") that the writer has two different thoughts in her mind at the same time: "On the one hand, my partner annoyed me, but on the other hand, we could also have a good conversation". However, as the further analysis of the latent structure shows, the person writing does not clearly keep these two thoughts apart and mixes these separate action intentions of approaching versus distancing. The separation of the two thoughts, in turn, opens up the possibility space in the practical work with the client to discover new things about herself. But first it must be understood what is actually at stake in the moment before therapeutic interventions begin. What actionable reality lies behind the observed verbal expressions? What motivates action from moment to moment here? It is imperative that these questions be successfully addressed in order to be able to act flexibly in therapy and to expand the possibility space instead of shrinking it even further. What, in turn, clients do with this interpretation is another matter.

Oevermann (2002, p. 33) states: "Sequence analysis nestles up to the real human-social events in its basic structure and is therefore not, like the otherwise usual measuring and classification procedures, a method external to the object, but one corresponding to and appropriate for the object itself. In fact, in practical life, decisions must also be made in principle at each sequence point among the options still open to an open future." Sequence analysis "... leans on the sequentiality that is constitutive of human action" (ibid., p. 6). Sequential action, however, does not mean simply working from front to back. Rather, it is a matter of opening up new possibilities (the term reading is used) strictly on the basis of the sequence of the text to be analyzed, and closing them down again when they cannot stand up to scrutiny on the text. Sequence analysis is thus the interplay between possibility (of "open options into an open future," see above) and reality (or the attempt to find these options again in the text).

5.2.2 Principles of hypothesis generation

As cited earlier, Wernet (2009, p. 39) gives a simple "answer to the question: what do I have to do to perform a methodologically testable operation of meaning reconstruction along valid rules?" The answer consists of a methodological three-step process, of which the first two steps interest us for hypothesis generation here: "I have to (1) tell stories, (2) form readings ...". The third step, confronting the readings with the actual context, will concern us in hypothesis testing (see Ch. 5.2.3).

Before we start interpreting text segments, we need to clarify what the case is and in which context it is embedded. According to Wernet (2009), this includes, on the one hand, clarifying and disclosing the research interest, and on the other hand, clarifying what it is that the text protocol actually logs, what social reality it logs, or more specifically, what contribution to answering the research questions an interview, for example, can make. We start with the objective data of the case (birth, milieu, occupations, dates of death, etc.), and only then the actual text (classroom conversations, therapeutic conversations, biographical interviews, etc.) is subjected to analysis. The collection of the objective data corresponds, for example, to the drawing up of a genogram (family data).

Interpretation is guided by the five rules of contextual indepenence, literalness, sequentiality, extensiveness or totality, and parsimony (see Wernet, 2009, pp. 21-38):

Contextual independence:

Of course, a textual interpretation must take into account the context in which the protocol emerged – but only after possible meanings of a text segment have been tapped independently of that context. Thus, one first invents stories in which the critical text passage could meaningfully occur. Wernet speaks here of designing exclusively "thought-experimental contexts" for generating possible meanings of the text segment in the initial text access. Wernet warns that interpretation would be dependent on the interpreter's everyday understanding or non-scientifically obtained prior understanding, and thus a circular interpretive process could be set in motion if prior understanding is not methodically set aside for the time being.

Literalness:

Literalness means that the interpretation is guided precisely and exclusively by what actually and verifiably occurs in the text - and not by what the text might have wanted to say. The text as such is a piece of reality and is to be treated as such. Two important conditions follow from this:

(1) The text protocol must - actually as a matter of course - be the basis of the interpretation in its original form, e.g. as a word-for-word transcription of a social interaction, not in the form of a paraphrase alienated by everyday understanding and social conventions, at least reduced in its possibilities of meaning already in advance. Such paraphrases, as they are sometimes used in qualitative content analyses, are absolutely out of the question.

(2) The interpretation must be downright fussy with the text. It must not overlook bumpy or even inappropriate details, as we might do in everyday life. Wernet calls for "putting the text 'on the gold scale' in a way ... that in everyday life "... would seem petty."

In this way, which is inappropriate in everyday conversation, perhaps even outrageously Beckmesserian, the latent meaning behind the manifest formulation is revealed. In the examples above, "I want to get a grip on my addiction" and "Also, however, we had very good conversations," the literal interpretation encounters the divergence of what is meant from what is actually said, and thus the ambiguity of these statements and the discrepancies in the person's lifeworld.

Sequentiality:

This principle determines the core of the analysis, because the procedure of interpreting is strictly logical and step-by-step oriented. This makes the procedure scientific. Thus, evidence for possible interpretations (readings) is not searched across and unsystematically through the text (looking for positive confirmation). Rather, an alternation of formulating hypotheses, condensing them into readings, and testing them against the text is practiced in strict sequence.

Where exactly the interpretation begins, which text segment represents the opening sequence, must be justified in terms of content. It is not necessary to start with the first sentence or part of a sentence. But then one follows methodically strictly, step by step the sequence of the protocol. What follows in the text behind the segment that is currently in focus must - at first - be disregarded. Of course, each text segment is embedded in the "inner context" of the text meaning that has been developed so far. The disregard of the respective following text segments is methodologically and practically highly significant: "The thought-experimental continuation makes clear that the respective concrete case must make a 'decision' to be what it is ...". Wernet emphasizes that this "...points to the reconstruction of the 'becoming-so-and-not-other' of a life practice."

The principle of sequentiality does not prohibit skipping text segments and selecting relevant text passages depending on the research question and the progress of the reconstruction of the meaning structure of the text. However, their beginning must then be justified again, and the interpretation must again proceed sequentially step by step. The selection of later text passages is based on whether they can confront or perhaps even support the different readings.

Sequentiality also means not allowing prior knowledge about later developments in the text to enter into the interpretation, but rather to be guided by the step-by-step development in interpreting. It is important not only to look for confirmation of hypotheses, but rather precisely for counter-evidence in order to be able to invalidate them. This corresponds to the falsificationist approach of critical rationalism (Popper, 1943). The reconstruction of meaning must adhere exactly to the text and be verified exactly there. In practice, however, one can usually identify attempts to empirically support one's model ideas rather than to question them. A fair examination of competing hypotheses then takes place only to a limited extent.

Extensivity:

Extensivity or totality is a balancing principle that should prevent following subjective and arbitrary intentions when interpreting. Thus, apparently unimportant details should not be overlooked, but neither should the interpretation be limited to the apparently important parts of the text. "Important" and "unimportant" appear as a consequence of subjective prejudices. Rather, it is significant to proceed with the interpretation in such a way that the text is analyzed step by step, without favoring or disfavoring anything. It is essential to keep a consistent equal treatment of all text passages. In particular, also, "...the thought-experimental contexts must be fully illuminated typologically ...".

Parsimony:

This principle requires that normality, understood as the everyday with its routines, be assumed when constructing possible interpretations of life practice. The demand here is to ". admit only those readings that are textually verifiable." This restricts "storytelling" to variants that are compatible with the text as a whole. Thus, the principle of parsimony in the interpretation of life practices excludes science fiction stories, esoteric digressions, or especially the unfounded attributions of a pathological deviation from normal behavior.

A deviation from the norm, which is often also regarded as pathological, is very easy to notice in protocols, because here the room for interpretation narrows drastically. Routines and sustainable problem solutions are less conspicuous because the spectrum of possibilities is very large. Normality is therefore more difficult to reconstruct than deviation from the norm. "Parsimonious" is thus to be understood in two ways: On the one hand, normality is initially taken as given, and the assumption of deviation is subject to justification. It is a matter of discovering truly significant deviations from the normality of human action. On the other hand, the space of possibility is limited by the fact that interpretation is carried out only on the existing text and all conclusions are based only on the text, not on everyday experiences.

In summary, the maxim is: what is interpreted must be substantiated on the text and what is substantiated belongs in the interpretation.

5.2.3 Condensing hypotheses into versions of the text (grouping)

The totality of sequencing rules or meaning-generating rules, which generate anew a multiplicity of options at each sequence point, at each text segment, is a set of algorithmically-typologically different rules. As examples of these rules, Oevermann (1996b, p. 7) mentions linguistic syntax, the pragmatic rules of speech action, and the logical rules for formal and material-substantive reasoning (induction, abduction, deduction; cf. Reichertz, 2000). Oevermann speaks of well-formed utterances with regard to the interpretations that arise from possibilities. A well-formed utterance exists when an interpretation has been generated according to the rules and with the building blocks of a language, when it is a normal written or oral linguistic utterance that conforms in word choice, style, and grammar to all the rules of the language in question. Thus, it is a grammatically correct utterance and therefore a normal sentence.

The hypotheses are grouped to derive types. These types in their formulation form the readings of the text. Only after this step, readings are tested on the text and confronted with the reality of the text. In this way, the connection between the general (abstract, case-independent structure) and the particular (concrete life practice) becomes clearer and more precisely outlined.

5.3 Testing hypotheses

After the basic structure has been reconstructed, the hypotheses generated in the process are tested. In technical terms, the principle of falsification (Popper, 1934) is followed. That is, a hypothesis is not retained because it could be proven, but it is retained because its opposite, its inapplicability to the concrete case, could not be proven. Accordingly, there is no such thing as an always valid truth, but only hypotheses whose non-provability has not yet been proven. The testing of a hypothesis is thus equivalent to the attempt to prove and justify that the hypothesis is not compatible with the text.

There is now no need to proceed sequentially. In the search for evidence for the stories generated in the course of hypothesis generation, or for the readings of the text designed from them, it is necessary to "wander" in the text - quite specifically to find just such counter-evidence.

Possible difficulties in verification mostly result from violations of the rules for text interpretation. Thus, bold conjectures may not have been stated and truly pursued, although it is precisely the potential possibility of a conjecture's failure that constitutes its explanatory power. A hypothesis is strong if it takes the risk of being brought down by even a minor inconsistency. From this derives a larger scope of validity. A hypothesis that cannot be refuted in principle is scientifically worthless.

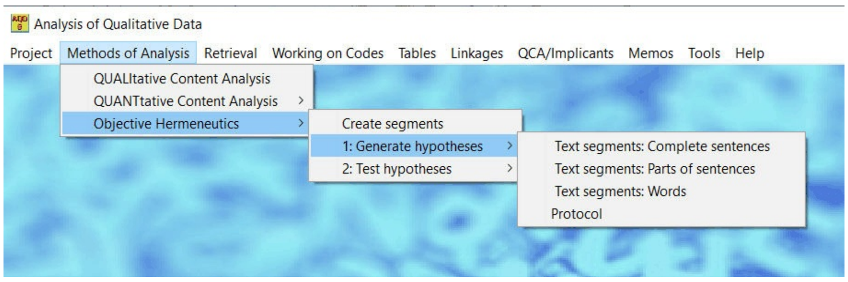
In recourse to the recommendations on hypothesis generation, it must therefore be added that every possibility, however small, of how a text is to be understood must be investigated. This requirement, of course, makes the procedure very laborious, but also leads to the fact that a very small sample is sufficient to cover a research field comprehensively. Oevermann himself goes so far as to claim that general statements can be made with a single well-analyzed case. Hildenbrand (2006) assumes about eight to ten cases are necessary to reach theoretical saturation in the sense of the research strategy of grounded theory according to Glaser (1998); after that, further cases do not promise any additional substantial gain in knowledge.

Analysis continues until the case structure is fully reproduced once (Oevermann, 1996). Since the internal logic reproduces and develops anew in each section, it is sufficient to examine a few passages in great detail to still obtain an overall view of the case. This stability of textual interpretation legitimizes analyzing (only) up to the (first) repetition of the structure and then looking for counter-evidence to invalidate it. This is emphasized by the fact that one "only [really] knows a case structure regularity or a case structure ... only [properly knows] when one has reconstructed, by sequence analysis, a complete phase in its reproduction or transformation" (ibid., p. 9). The analysis then gives rise to the manifest and obvious or the latent sense structures of daily routine in standard situations (ibid., p. 76 f.).

5.4 Generating and testing sequence analytic hypotheses with AQUAD

The preparation of texts has already been described above. In the following, we will go into the generation and verification of hypotheses using the example of the text excerpt "Cooperation protocol.txt". This text was saved as an example of a project "Communication" in the main directory of AQUAD (e.g. "C:\Aquad_8") on the hard disk when AQUAD was installed. We now let this text be structured (with the function "Preparation of texts") into complete sentences. These are then available for analysis in the subdirectory "C:\Aquad_8\cod" as the file "{s-kS}cooperation protocol.txt".

5.4.1 Hypothesis generation

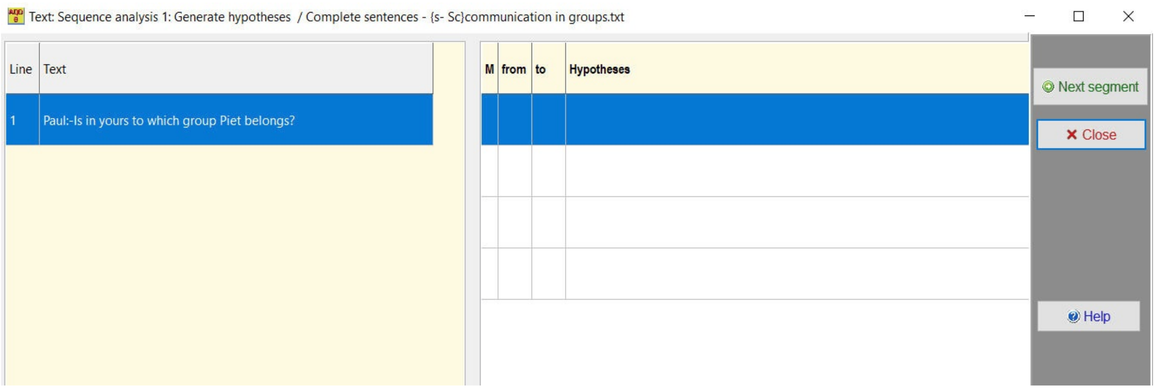


To start, we select "Objective Hermeneutics" -> "1: Generate hypotheses" -> "Text segments: Complete sentences" in the main menu.

The first thing that opens is – as in the beginning – the window where we select a file of the current project for analyzing. Now, the project "Communication" includes only one file "{s-kS}communication in groups.txt". We select it by clicking on it.

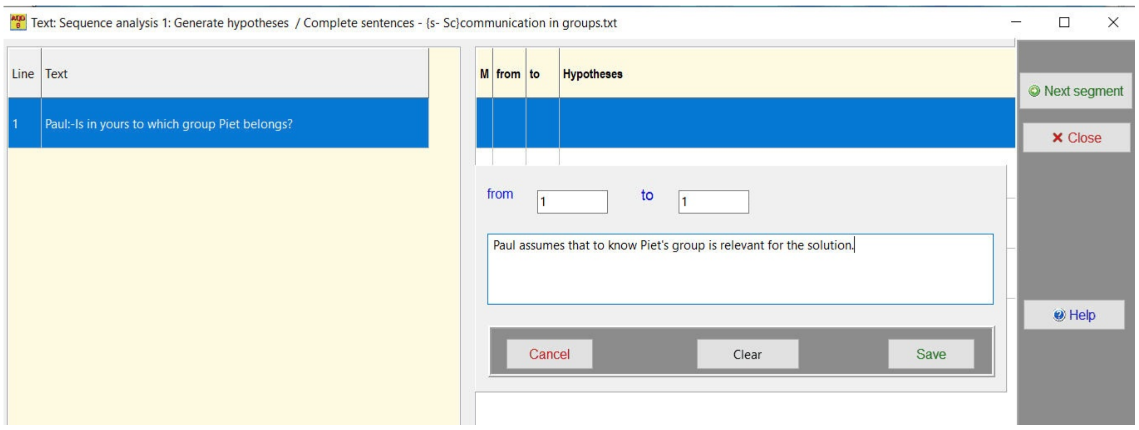
The window for entering hypotheses will open. It shows in the blue framed text field at first only the first text segment. The entire protocol comes from a study on cooperation by Van der Linden, Erkens, and Nieuwenhuysen (1995), in which two students each had a letter from boys at a summer camp and were asked to cooperatively figure out who was with whom at the camp:

P:- Do you say which group Piet is in?


Text: Sequence analysis 1: Generate hypotheses / Complete sentences - {s- Sc}communication in groups.txt
Line
Text
1
Paul:-Is in yours to which group Piet belongs?
M
from
to
Hypotheses
Next segment
Close
Help

The window on the right lists the hypotheses we are generating as we go along. Of course, this window is still empty, because we are just starting with the sequence analysis. To start, we click on the blue marked first text segment on the left, whereupon an input window for our first idea, the first hypothesis for the marked text segment, opens in the hypothesis pane on the right:

Here we enter our hypothesis (maximum 264 characters long) and then click on the button "Save", which transfers our hypothesis to the protocol window.

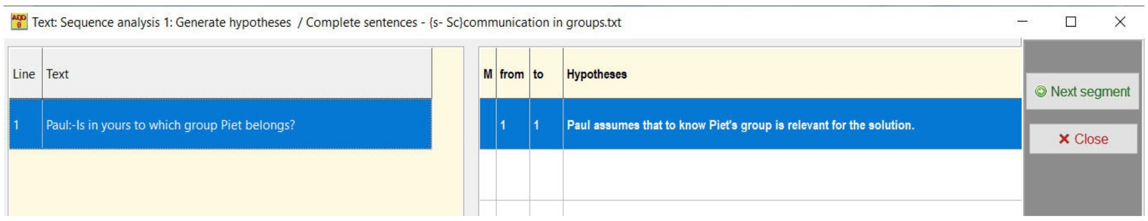

Text: Sequence analysis 1: Generate hypotheses / Complete sentences - {s- Sc}communication in groups.txt

Line	Text
1	Paul:-Is in yours to which group Piet belongs?

M	from	to	Hypotheses

from 1 to 1

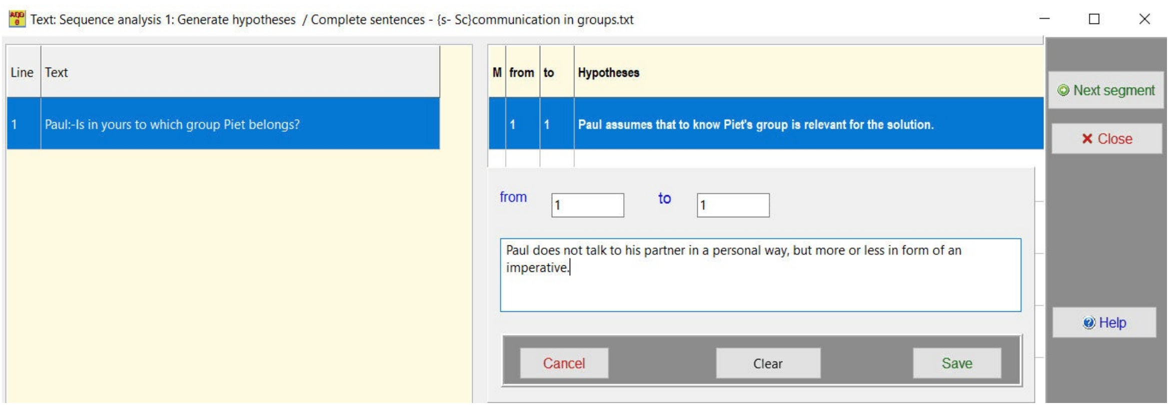
Paul assumes that to know Piet's group is relevant for the solution.


Text: Sequence analysis 1: Generate hypotheses / Complete sentences - {s- Sc}communication in groups.txt

Line	Text
1	Paul:-Is in yours to which group Piet belongs?

M	from	to	Hypotheses
	1	1	Paul assumes that to know Piet's group is relevant for the solution.

What else do we think of or notice about this segment? When we click on the text on the left again, the input window opens again on the right. We enter our hypothesis and save it again:


Text: Sequence analysis 1: Generate hypotheses / Complete sentences - {s- Sc}communication in groups.txt

Line	Text
1	Paul:-Is in yours to which group Piet belongs?

M	from	to	Hypotheses
	1	1	Paul assumes that to know Piet's group is relevant for the solution.

from 1 to 1

Paul does not talk to his partner in a personal way, but more or less in form of an imperative.

We continue in this way until we have no more ideas about this segment. Then we click on the "Next segment" button on the far right and the next text segment appears on the left, for which we again write down all the hypotheses that run through our heads. We continue in this way until we get the impression that the hypotheses are repeated. At this point, we stop "telling stories" about the next segment. Rather, we now continue in the text until new aspects emerge in the text that are relevant to the research question. From this segment on, we again work strictly sequentially! Once we have sifted through the entire text in this way and sequentially analyzed its relevant segments, we can move on to the second phase, hypothesis testing.

You can find the complete listing of hypotheses if you load the sample project "Communication", which was installed with AQUAD during setup, and activate "Phase 1: Hypothesis generation". You will then also see that hypothesis generation has ended after the 14th text segment, as the structural patterns repeat.

5.4.2 Hypothesis testing

If we click in the hypothesis window on the right on the number of a hypothesis (columns from or to) a window opens in the window of the text segments on the left for the selection of further activities. There we can accept the selected hypothesis (in the example hypothesis 1)

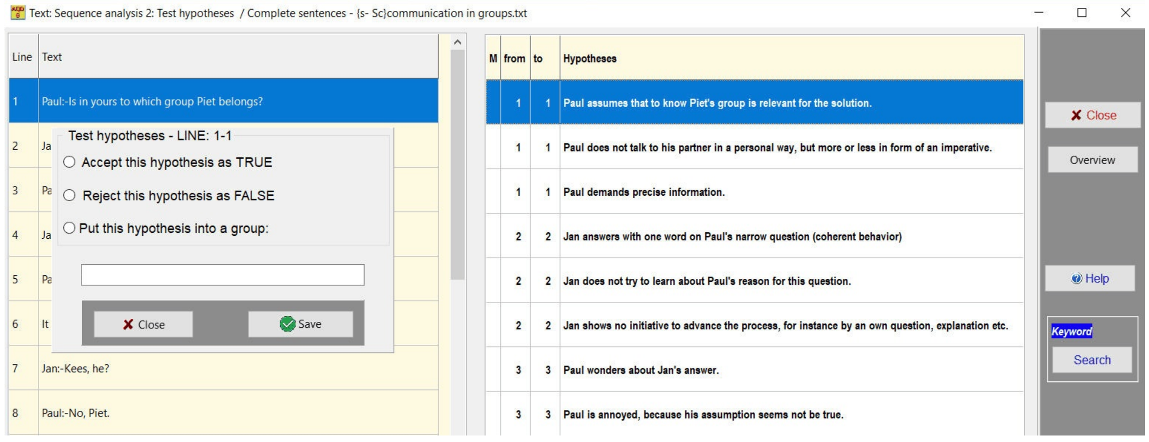
- as true. It will then be marked as true with the code "T-".
- Reject it as false. It will then be marked as false (False) with the code "-F".
- Or assign it first to a "reading", i.e. to a certain group. Which group it is and how it should be named is determined by entering a suitable name in the input window. This assignment is also prefixed as a code ("-G-:") to the hypothesis.

The ability to group can also be used to identify relationships between text segments, such as specific sequences, causes and consequences, etc. In version 7 of AQUAD, specific queries were built in for this purpose, e.g., by "type" (of hypothesis) or "reference" (to other hypotheses). The assignment to a specific group "-G-: " replaces these rigid classifications. Behind the marker "-G-:" we can note any typifications and correlations that make sense in the individual project. The search function "Keyword" helps to find the different groupings again and to save or print them.

We can now organize hypotheses according to "readings" of the text by assigning similar hypotheses to a common hypothesis and now also assigning the reading or version to a group. We could assign hypothesis 2 to segment 1: "Specific, narrow question (minimalist)" (see figure above) hypotheses 13 (segment 4), 23 (segment 7), 29 (segment 9), 40 (segment 12), and 46 (segment 14) to each other. As reading type/grouping one would enter "Narrow question".

Again: With the search function "Keyword" (at the right margin) we can get an overview, e.g. of all "true" or "false" hypotheses or of certain groupings. These overviews can of course be saved and printed.

And what is the result of the sequence analysis of the example project "Communication"? Why don't you run the hypothesis testing with the installed material or delete the files "{s-cS}communication in groups.shg" and "{s-cS}communication in groups.shc" in the subdirectory ..\cod or copy these files to another directory. You can then test the sequence analysis from the beginning with your own hypotheses. In any case, the result will be a case structure that illustrates how cooperation fails when the partners do not pool their resources, but one partner pursues his/her own ideas and wants to use the other only as an information provider.


Text: Sequence analysis 2: Test hypotheses / Complete sentences - {s- Sc}communication in groups.txt

Line	Text
1	Paul:-Is in yours to which group Piet belongs?
2	Ja
3	Pa
4	Ja
5	Pa
6	It
7	Jan:-Kees, he?
8	Paul:-No, Piet.

Test hypotheses - LINE: 1-1
- ○ Accept this hypothesis as TRUE
- ○ Reject this hypothesis as FALSE
- ○ Put this hypothesis into a group:

M	from	to	Hypotheses
	1	1	Paul assumes that to know Piet's group is relevant for the solution.
	1	1	Paul does not talk to his partner in a personal way, but more or less in form of an imperative.
	1	1	Paul demands precise information.
	2	2	Jan answers with one word on Paul's narrow question (coherent behavior)
	2	2	Jan does not try to learn about Paul's reason for this question.
	2	2	Jan shows no initiative to advance the process, for instance by an own question, explanation etc.
	3	3	Paul wonders about Jan's answer.
	3	3	Paul is annoyed, because his assumption seems not be true.

Chapter 6: Special questions about coding

6.1 Setting up a project

The text files to be encoded must be readable by AQUAD, i.e. the text files must be imported as plain text files (*.txt). From then on, AQUAD will no longer work with the original transcriptions, but with internal copies. Let's now go step by step in great detail to the work of coding.

When you start AQUAD from the file manager or from the desktop, the title page appears. If you first want to combine texts for analysis in a project, you now click on "Define project" in the "Project" module, write a project name and compile a list of texts to be analyzed in this project. The settings you define in this way are automatically activated when you start AQUAD again - until you enter a new project or open another project that you have already defined. Of course, if you want to (re)open a project you've already defined, click "Open Project" right away.

You may want to familiarize yourself with coding in AQUAD by working with the text examples provided. To give you an example of how to proceed with AQUAD, four interviews with beginning teachers (in Spanish schools) have been installed as the sample project "Interview" and a fairy tale by the Danish poet Hans Christian Andersen as the sample project "Poet". Simply because the texts are shorter, let's take a closer look at the "Poet" example here.

The text is divided into the files "poet001.txt", "poet002.txt" as well as "poet003.txt" and stored in the directory that was specified for the database when AQUAD was installed: The original files are always stored in the root directory of AQUAD, e.g. in "D:\aquad8".

Since AQUAD always activates the current project automatically at startup, you do not have to enter the settings again and again. You jump directly from the start window to the main menu. To view or continue working on the present coding, first open the default project "poet.prj" in the "Project" module with "Open project". Then select the option "Qualitative content analysis" in the module "Methods of analysis".

If the meaning of a menu option is not entirely clear to you, AQUAD offers general help for common problems in the main menu (see the last option in the main menu; details below in Section 6.2.7) and contextual help within various windows.

6.2 How to proceed with coding?

6.2.1 Types of codes

For working with AQUAD, it is useful to distinguish clearly between six types of codes:

(1) Conceptual codes are used as "labels for distinguishable events, occurrences, and other kinds of phenomena" (Strauss & Corbin 1990, p. 61). As an example, consider "deliberating." Wherever in a text one gets the impression that someone is given an advice or that someone is giving advice to others, one will attach an appropriate code, such as "advice." Conceptual codes may be up to 60 letters and/or digits long in AQUAD. It is not necessary to use cryptic abbreviations, because each code must be written only once. It is then automatically listed in a code register. From there, it can be easily selected again and again by clicking on it.

(2) Profile codes, mostly socio-demographic codes, characterize an entire file. They define its profile, so to speak. They can be qualitative or numerical. For example, the characteristic of a file may be that it is an interview with a "female" person, or that the field notes refer to a school "with more than 1000 students." Since each such characteristic is needed

only once to identify a data file - "gender" as a profile category of the interviewee cannot be both "male" and "female", "school size" cannot be both below and above 1000 students - these codes can appear only once in each data document. We therefore also speak of singular codes. To distinguish them from conceptual codes, it is agreed that profile codes always begin with a slash "/", so in our examples we would enter:

/gender: w
/school size > 1000

(3) Numeric codes are used when the phenomenon the researcher is interested in exists as a quantitative value, such as test scores, time markers in an interview recording, the age of the interlocutors, etc. Numeric codes are particularly useful when testing hypotheses or within tabular or matrix analysis. However, AQUAD must be able to distinguish numeric codes from other types of codes. Since numeric codes often also identify the profile of a file, such as when we record the age of interviewees or indicate the school size in the example above, it is agreed as a flag that numeric codes also always begin with the slash "/". After the slash, a word can express the meaning of the following digits. Finally, the actual quantitative date follows. For example, if we want to include the age of our interviewees, we use the notation e.g. "/age: 18" to record the information as available date that the person is 18 years old.

(4) Program internal control codes or "switches" occupy a special position in the files. They contain no information about the data or their producers. Therefore, they play no role in the process of interpretation. In the current version of AQUAD, only one control code is currently defined for processing text files:

$do not count

This control code excludes segments marked with "$not count" in text files from analysis at the level of individual words (count, keyword search, etc.).

(5) Speaker codes start with the two defined characters "/$" and behave like a mixture of profile codes and control codes. They virtually split a file into subfiles for certain analyses ("Retrieval" and "Table analysis"). We call them "speaker codes" because they were introduced to support the analysis of group discussions. When the parts of speech are assigned to individual group members with speaker codes, the text can be analyzed both as a whole and in the individually assigned parts (e.g., /$John, /$Mary, etc.). One can also use these codes to distinguish short answers or statements of many people in a single file (e.g., transcription of answers to some open-ended questions). In other words, speaker codes assign file segments to specific persons.

Attention: In the analysis speaker codes have an effect only
- when searching for codes (i.e. not when counting codes - this is handled for different speakers with table analysis);
- at the first level of table analyses, i.e. as top-level column definitions.

(6) Sequence codes correspond in principle to (1) conceptual codes in that they represent "labels for distinguishable events...". The difference with simple conceptual codes is that they define a data segment, which in turn is defined by two or more conceptual codes in a defined sequence. For example, when coding data segments in which someone is given advice (see (1), conceptual code "advice"), if we often find that the advice is not accepted but rejected, we could distinguish two typical advice sequences: "advice-acceptance" and "advice-rejection". We can then insert the code "Advice - Acceptance" (or just "Advice +") in the first case, and the code "Advice - Rejection" (or "Advice -") in the second. The coded, internally linked data segment then ranges from the beginning of the advice to the end of the accepting/rejecting response of the person being advised.

Back to the beginning of the process: In the data reduction phase, all files must be coded. However, in the following we have to consider differences in dealing with text, sound, video and image files. The basic rules of encoding with AQUAD will be presented in detail using text files as an example, and the specifics of the other file types will be presented afterwards.

6.2.2 Selecting text files for encoding

Let's find out a little more about coding. After clicking on "QUALitative Content Analysis", a box will open that contains the names of all the text files in your current project.

Before coding, you must decide which text file you want to work on. The names of the files are shown in the window. Select one of them by double-clicking on it. (The total text "Poet" has been divided into three parts by content sections poet001, poet002, poet03 to give you an opportunity to become familiar in AQUAD with the possibilities of working on multiple texts in one project).

An 'X' in front of the file names indicates that coding has already been done within these texts, i.e. that these texts have been interpreted before. If the box is empty when you open it, it means that you have not defined a project yet.

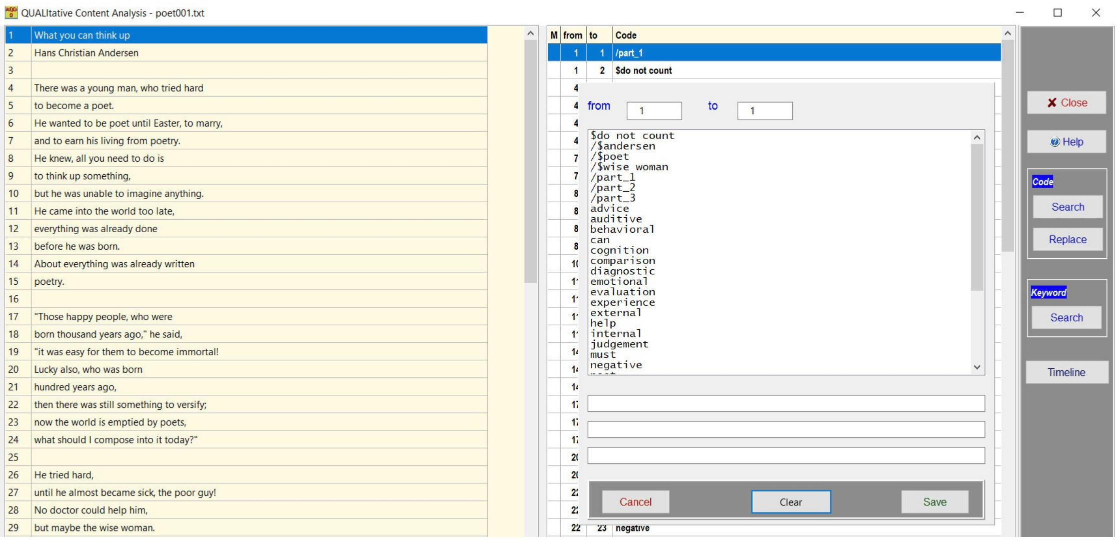
If you decide to select "poet001", AQUAD loads the encoding window with various function buttons. More about this later. Now, before you click on "poet001", let us remind you again that the text lines should not be too long.

6.2.3 Rules for inserting codes

When you have selected the file you want to code, a window will open showing an empty gray table on the right. If you have worked with this file before, the table shows the previous encodings. Please don't be surprised about the dummy codes 'xxx' and 'yyy': these codes were included to test different functions. You can make your own experiments with such codes at any time without changing the relevant conceptual codes.

The first column "M" in the code window will later indicate whether memos have been written for specific codes – and play back those memos by clicking on them. The "from" and "to" columns indicate which lines of text the coded text segment spans.

On the left you can see the text window highlighted in light yellow:



1	What you can think up
2	Hans Christian Andersen
3	
4	There was a young man, who tried hard
5	to become a poet.
6	He wanted to be poet until Easter, to marry,
7	and to earn his living from poetry.
8	He knew, all you need to do is
9	to think up something,
10	but he was unable to imagine anything.
11	He came into the world too late,
12	everything was already done
13	before he was born.
14	About everything was already written
15	poetry.
16	
17	"Those happy people, who were
18	born thousand years ago," he said,
19	"it was easy for them to become immortal!
20	Lucky also, who was born
21	hundred years ago,
22	then there was still something to versify;
23	now the world is emptied by poets,
24	what should I compose into it today?"
25	
26	He tried hard,
27	until he almost became sick, the poor guy!
28	No doctor could help him,
29	but maybe the wise woman.

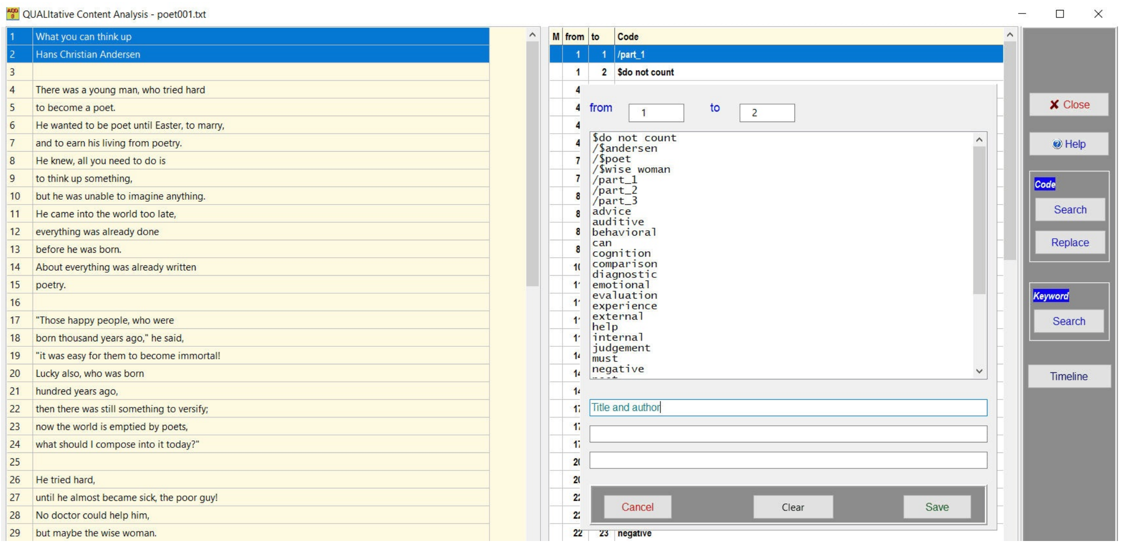
M	from	to	Code
	1	1	/part_1
	1	2	$do not count
	22	23	negative

from 1 to 1

```
$do not count
/$andersen
/$poet
/$wise woman
/part_1
/part_2
/part_3
advice
auditive
behavioral
can
cognition
comparison
diagnostic
emotional
evaluation
experience
external
help
internal
judgement
must
negative
```

To code any segment, move the mouse pointer to the left of the text window to the number of the first lines of the text segment you want to code. Now, while holding down the left mouse button, move the cursor to the number of the last line of that text segment and then release the mouse button.

Now two things happen: First, we see on the left side of the text window that the selected text segment (lines 1 and 2) has been marked. Secondly, the input window for codings opens on the right. We can see from the automatic entries that the text segment is in lines 1 and 2. From the code register we can now select a previously used code by clicking on it, or if no suitable code has been introduced yet, we can write a new code in one of the three input lines. You conclude correctly: Each text segment can be assigned up to three codes at a time in the same step. By clicking on "Save", the selected code or codes are transferred to the code table.



1	What you can think up
2	Hans Christian Andersen
3	
4	There was a young man, who tried hard
5	to become a poet.
6	He wanted to be poet until Easter, to marry,
7	and to earn his living from poetry.
8	He knew, all you need to do is
9	to think up something,
10	but he was unable to imagine anything.
11	He came into the world too late,
12	everything was already done
13	before he was born.
14	About everything was already written
15	poetry.
16	
17	"Those happy people, who were
18	born thousand years ago," he said,
19	"it was easy for them to become immortal!
20	Lucky also, who was born
21	hundred years ago,
22	then there was still something to versify;
23	now the world is emptied by poets,
24	what should I compose into it today?"
25	
26	He tried hard,
27	until he almost became sick, the poor guy!
28	No doctor could help him,
29	but maybe the wise woman.

M	from	to	Code
	1	1	/part_1
	1	2	$do not count
	22	23	negative

```
$do not count
/$andersen
/$poet
/$wise woman
/part_1
/part_2
/part_3
advice
auditive
behavioral
can
cognition
comparison
diagnostic
emotional
evaluation
experience
external
help
internal
judgement
must
negative
```

Title and author

Conversely, if you click on one of the codes in column 4 ("Code:") in the code window, the text segment coded in this way is highlighted on the left of the text window. At the same time, this text segment is displayed separately in a window with a green background. There, by pressing the right mouse button, you can call the text editor (notepad) and copy the selected segment (for subsequent pasting into other text files), print it, etc.

You can also call up the code register on the right for copying or printing in the text editor by right-clicking anywhere in the register.

It has proven useful to provide each new code name with a short definition. In this way, you will always be able to quickly recall the meaning or later present your coding system very easily with code, definition and text example. To do this, use the "Memo" function; it is activated when you click in the first column of this code entry (see above).

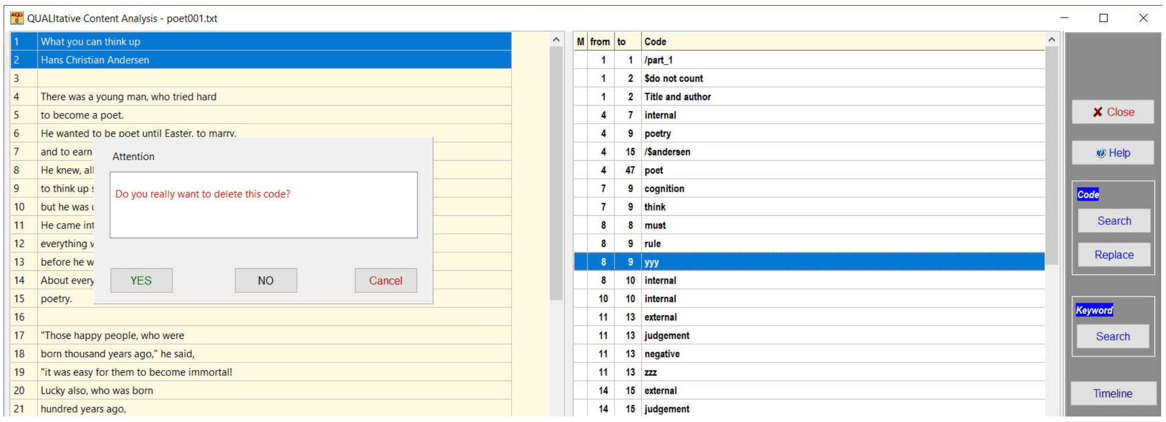
Please, remember when you mark something in the text field: The text is your database, you cannot change it during the analysis! AQUAD only *shows* the text, but does not allow *editing*. However, you can copy parts of the text (or the whole text) and move it to a memo: To copy, simply right-click in the text window. Then the whole text will appear in the text editor (notepad), where you can select critical parts of the text, copy them (and then paste them into a memo, for example), print them.

6.2.4 How to remove or replace codes

If you press "Cancel" (instead of "Save") in the coding window, your last entry will simply be ignored. If you discover a typo before "Save", simply double-click in the small code name entry window. The name will disappear and you can enter a new name. Of course, you can move around in this box with the cursor, delete or overwrite characters - just like in a text program.

But what do you do if later – after you have attached the code to the text segment by pressing the "Save" button – you find out that individual codes contain typos, are inappropriate or even misleading? Of course, then you want to remove them. Here is the suggestion how you can proceed. Later we will show you a more elegant, but also more dangerous way to search and automatically replace codes.

- Find the critical code in the coding table on the right and click in front of it in the assigned table column "from" or "to". A gray window (see figure below) will appear on the left side of the text window: "Do you really want to delete this code?
- To delete the selected code, click "YES!"
- Afterwards you can re-enter the code corrected.


QUALItative Content Analysis - poet001.txt

1	What you can think up
2	Hans Christian Andersen
3	
4	There was a young man, who tried hard
5	to become a poet.
6	He wanted to be poet until Easter, to marry,
7	and to earn
8	He knew, all
9	to think up
10	but he was
11	He came int
12	everything
13	before he w
14	About every
15	poetry.
16	
17	"Those happy people, who were
18	born thousand years ago," he said,
19	"it was easy for them to become immortal!
20	Lucky also, who was born
21	hundred years ago,

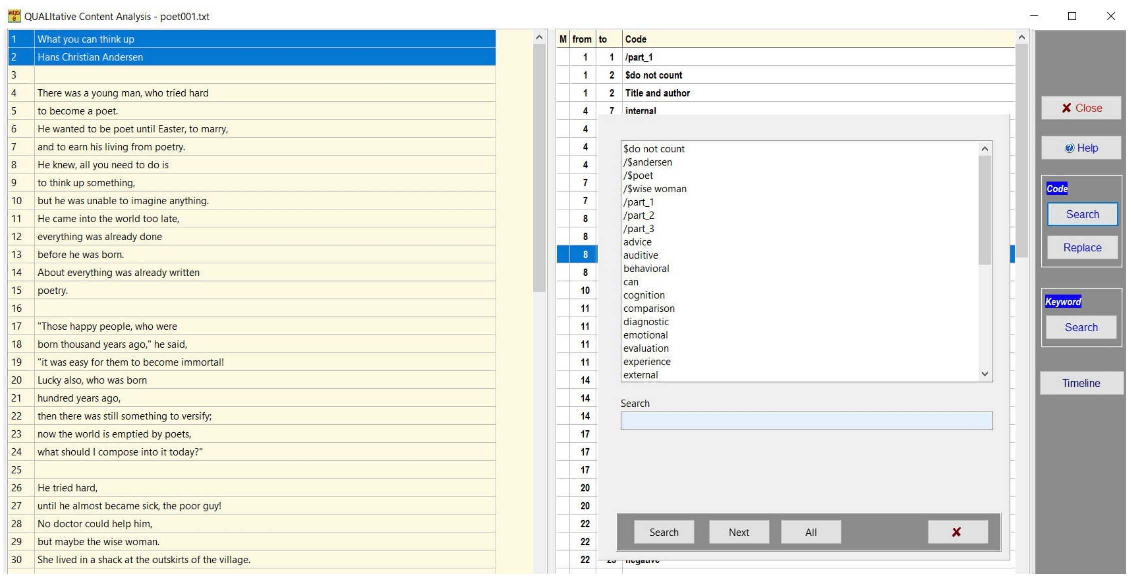
Attention

Do you really want to delete this code?

M	from	to	Code
	1	1	/part_1
	1	2	$do not count
	1	2	Title and author
	4	7	internal
	4	9	poetry
	4	15	/$andersen
	4	47	poet
	7	9	cognition
	7	9	think
	8	8	must
	8	9	rule
	8	9	yyy
	8	10	internal
	10	10	internal
	11	13	external
	11	13	judgement
	11	13	negative
	11	13	zzz
	14	15	external
	14	15	judgement

If you only want to make small corrections, it makes sense to first select the coded text segment again, select the incorrect code in the input window for codes, correct it in the input line and save it. Then delete the incorrect code as described. This way you do not have to retype the whole code.

However, deleting with this function becomes a chore if you want to delete more than one code entry. That's why there's a "Code" box on the right that contains two buttons, "Search" and "Replace". As you already guessed, the "Search" button is helpful in this case:


QUALItative Content Analysis - poet001.txt

1	What you can think up
2	Hans Christian Andersen
3	
4	There was a young man, who tried hard
5	to become a poet.
6	He wanted to be poet until Easter, to marry,
7	and to earn his living from poetry.
8	He knew, all you need to do is
9	to think up something,
10	but he was unable to imagine anything.
11	He came into the world too late,
12	everything was already done
13	before he was born.
14	About everything was already written
15	poetry.
16	
17	"Those happy people, who were
18	born thousand years ago," he said,
19	"it was easy for them to become immortal!
20	Lucky also, who was born
21	hundred years ago,
22	then there was still something to versify;
23	now the world is emptied by poets,
24	what should I compose into it today?"
25	
26	He tried hard,
27	until he almost became sick, the poor guy!
28	No doctor could help him,
29	but maybe the wise woman.
30	She lived in a shack at the outskirts of the village.

M	from	to	Code
	1	1	/part_1
	1	2	$do not count
	1	2	Title and author
	4	7	internal
	4		
	4		
	4		
	7		
	7		
	8		
	8		
	8		
	8		
	10		
	11		
	11		
	11		
	11		
	14		
	14		
	14		
	17		
	17		
	17		
	20		
	20		
	22		
	22		
	22		negative

$do not count
/$andersen
/$poet
/$wise woman
/part_1
/part_2
/part_3
advice
auditive
behavioral
can
cognition
comparison
diagnostic
emotional
evaluation
experience
external

Search

You do not have to write the searched code into the input field here either, but you mark it by clicking on it in the code register and transfer it – the search criterion – thus into the small input field "Search". All that remains is to start the retrieval by clicking on the "Search" button.

This will take you from one text segment to the next for which this particular code was used. It is best to click on "All" to get an overview in a list of the places where the code was found. With the routine "Delete code in all files" (select under "Edit codes" in the main menu) you can then decide whether you want to delete all instances of the critical code or whether you prefer to make the decision on a case-by-case basis. Caution: This routine does exactly what its name says - it is best to always put a backup copy of your codes in another directory.

You can proceed analogously to replace codes with others, either individually after selecting "Replace" in the "Code" button field or with the routine "Replace code in all files" (select under "Edit codes" in the main menu).

6.2.5 How to get an overview of codes

There are many reasons why a larger overview than the screen provides is necessary. AQUAD offers three ways to get an overview of the codes:

Overview of all codes used in the project

As mentioned earlier, AQUAD displays the code register to give you an overview of the codes used in your analysis at that time. During coding, AQUAD keeps a record and registers all new code names introduced. Each new code name is automatically sorted alphabetically into the code register. If you want to reuse a code, you can select it by clicking on it.

If you right-click in the code register, it will be transferred to the editor and can be copied from there (for pasting into other texts) or printed.

Overview of codes attached to a specific line

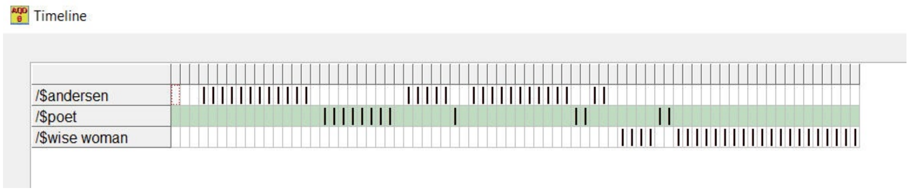
The "Text" column shows the current text file. Segments to which you have already attached one or more codes are displayed highlighted when you click in one of the code columns (see above). The line numbers where the text segments start are listed in the "From" column of the code table. All codes with the same segment start number were assigned to the same text segment.

Overview of codes of a file in sequential context

A selection of codes of interest is summarized in a code list (see under -> "Retrieval" ->"Code list" -> "Create code list") and then made visible in their mutual sequence within the coding windows of text projects, audio and video projects after clicking the "Timeline" button.

Since the purpose of the function is to make sequences of codings clearly manageable, the number of codes in the display is limited to 50. However, you should try to settle on a smaller number of critical codes if you want to take advantage of this feature.

As an example we try to show the sequence of speakers in the text "poet001.txt" using a code list "Speakers", where the three speaker codes are stored:

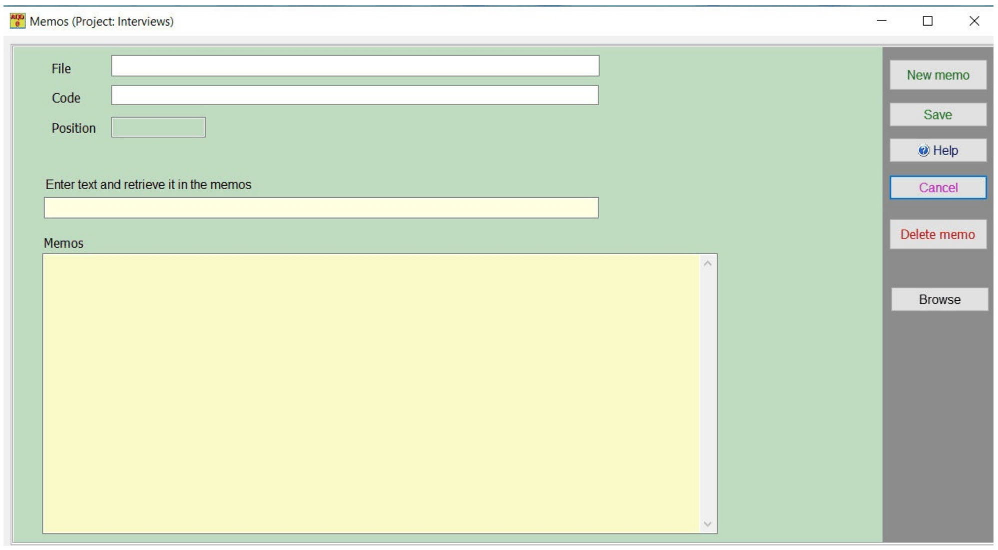


Above you can see the result of selecting codes from the Poet example, which have been combined into a code list (*.cco) as described above: When you click on the name of the critical code list, AQUAD creates a table row for each code in the list, in which its occurrence is marked by black blocks in relation to the use of the other codes. Representation units here are lines of text. In the case of audio recordings, the units of representation are seconds, and in the case of video recordings, 25 frames each (also equivalent to one second). This and the rounding errors in converting the units naturally cause overlapping of adjacent sections of different codes, but it also makes the position of the individual codes in the sequence of encodings easy to understand.

6.2.6 How to add comments to the codes

If you want, you can add a comment to each code. Theoretically, comments can be as long as you want; practically, there is a limit due to your computer's memory capacity. Comments are added in the form of "memos". They are useful, for example, when you start coding with very general codes and already expect that differentiation will become necessary as you work on the texts. At a later stage, you can easily turn these comments into differentiated codes. Now the first column of the code window becomes interesting, which is headed "M" for "Memo".

Clicking directly in this column will activate the memo function for that line. An "X" indicates that there is already at least one memo for this coding that starts in this line. When you click on it, the memo window opens – an empty window if there was no "X" yet.


Memos (Project: Interviews)
File
Code
Position
Enter text and retrieve it in the memos
Memos
New memo
Save
Help
Cancel
Delete memo
Browse

For later search, and also when you select the "Memo" option from the main menu, three characteristics can be defined for each memo:

- the name of the file to which the memo belongs;
- the code that belongs to it;
- the text line to which the memo refers.

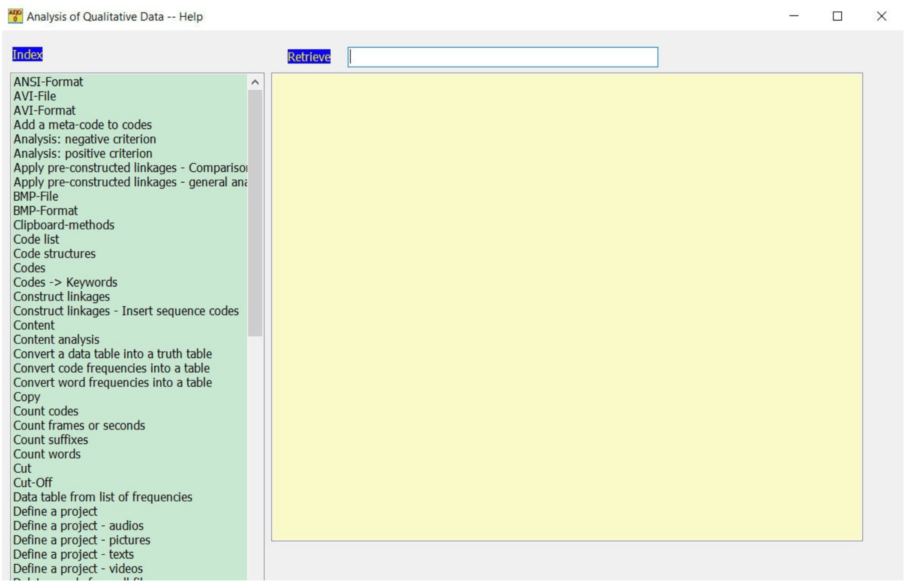
In the window under the yellow heading "Enter text and search in memos" you can enter a keyword or a related part of text, which will be searched for in all memos. For example, when coding, we enter definitions for the codes as memos, prefixing each time with the heading "Code definition". With this keyword, we can compile all these definitions in a flash by selecting the "Browse" function (right).

The large yellowish window needs no explanation: this is where you write or import your memo. The right mouse button gives you access to copy, cut, paste, etc. The two buttons at the top right do what they indicate: you can use them to enter new memos and delete existing ones.

The buttons on the right are self-explanatory - only "Browse" may need additional explanation: Scrolling in combination with features - such as searching for a specific text related to a specific code - is usually done using the memo function from the main menu. You can use the feature to get quick insight into memos with specific content.

6.2.7 When a little help is needed

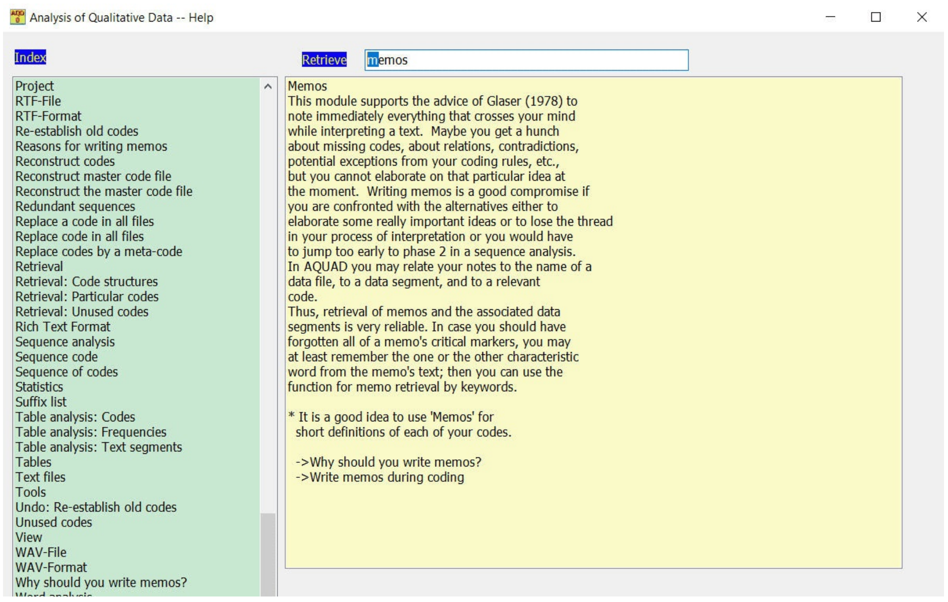
You may not always want to search through the entire manual if the meaning of a menu option is not entirely clear to you. No problem - AQUAD offers general help for common problems in the main menu (see the last option in the main menu) and contextual help within different windows. Let's say you want to get more specific about what types of text files AQUAD accepts. Let's click on "Help" in the main menu, then on "Contents" in the submenu and see what happens:



ANSI-Format
AVI-File
AVI-Format
Add a meta-code to codes
Analysis: negative criterion
Analysis: positive criterion
Apply pre-constructed linkages - Compariso
Apply pre-constructed linkages - general ana
BMP-File
BMP-Format
Clipboard-methods
Code list
Code structures
Codes
Codes -> Keywords
Construct linkages
Construct linkages - Insert sequence codes
Content
Content analysis
Convert a data table into a truth table
Convert code frequencies into a table
Convert word frequencies into a table
Copy
Count codes
Count frames or seconds
Count suffixes
Count words
Cut
Cut-Off
Data table from list of frequencies
Define a project
Define a project - audios
Define a project - pictures
Define a project - texts
Define a project - videos

On the left of the figure, a green window will show the list of available keywords. The assigned help text will appear on the right when we click on a keyword. To get the information on "Memos", we can drag the scrollbar on the left down until the desired keyword appears, or we can write the keyword in the "Retrieve" input line at the top. You will notice that even as we type our word "Memos" the list scrolls down and the word we are looking for is highlighted. Now we click on the highlighted word and the help text appears on the right (see next page).

Hints to additional information are shown when you press on sections of the help text marked by arrows. In this example, such a link is available with the keyword "-> Text format (*.txt)". It is best to call up the "Help" function in the main menu and simply explore the possibilities.



Project
RTF-File
RTF-Format
Re-establish old codes
Reasons for writing memos
Reconstruct codes
Reconstruct master code file
Reconstruct the master code file
Redundant sequences
Replace a code in all files
Replace code in all files
Replace codes by a meta-code
Retrieval
Retrieval: Code structures
Retrieval: Particular codes
Retrieval: Unused codes
Rich Text Format
Sequence analysis
Sequence code
Sequence of codes
Statistics
Suffix list
Table analysis: Codes
Table analysis: Frequencies
Table analysis: Text segments
Tables
Text files
Tools
Undo: Re-establish old codes
Unused codes
View
WAV-File
WAV-Format
Why should you write memos?

Memos
This module supports the advice of Glaser (1978) to note immediately everything that crosses your mind while interpreting a text. Maybe you get a hunch about missing codes, about relations, contradictions, potential exceptions from your coding rules, etc., but you cannot elaborate on that particular idea at the moment. Writing memos is a good compromise if you are confronted with the alternatives either to elaborate some really important ideas or to lose the thread in your process of interpretation or you would have to jump too early to phase 2 in a sequence analysis.
In AQUAD you may relate your notes to the name of a data file, to a data segment, and to a relevant code.
Thus, retrieval of memos and the associated data segments is very reliable. In case you should have forgotten all of a memo's critical markers, you may at least remember the one or the other characteristic word from the memo's text; then you can use the function for memo retrieval by keywords.

* It is a good idea to use 'Memos' for
short definitions of each of your codes.

->Why should you write memos?
->Write memos during coding

6.2.8 Why not let AQUAD search: semi-automatic coding

AQUAD's search functions offer the possibility of computer-assisted searching for keywords in the text - a kind of semi-automatic coding. When coding, we click on the "Search" button in the "Keyword" field on the right, write a keyword (or a sequence of words or parts of words) in the window that appears afterwards, and then make AQUAD go through the text.

So, in our example project "Poet", we might be interested in where something is said about the hero's intentions in "poet001.txt". We might enter the keyword "poet" and click on "Search". Since this search function is not sensitive to upper and lower case, we will get as a result in the text poet001 all occurrences of "poet", if we have chosen the first of two additional settings. There are two settings:

- Whole words
- Part of words

If we choose the second setting and enter "poet" as a criterion again, we will find the noun "poetry" in addition to two places where a poet is mentioned. Often such a simple procedure helps to quickly find critical text segments.

In addition, the following applies: We must always check and interpret all found passages!

What can we do if several keywords are relevant, or if several keywords define a field of meaning that is important for the analysis? Let us assume that in the text poet001 we want to find out if there are more perceptions than thoughts and ideas expressed in it. We can compile a list of relevant words or word stems such as hear, ear, see, eye, etc. Unfortunately, the keyword option in this part of the program forces us to enter one keyword at a time and have it checked.

It is quicker to find all the words together using the keyword function in the "Retrieval" module. If you want to have a quick overview of keywords in all your files (without loading one after the other), you can use the "Count words" option in the same module (more about this in Section 7.5 within this chapter and in Chapter 9).

6.2.9 If you want to print code files

The code list on the right side of the "Qualitative Content Analysis" window always gives you an overview of your codings, but only on the screen when you have AQUAD open. How do you proceed if you want a printout of your codings?

- Activate the "Qualitative Content Analysis" function;
- select the text file for which you want to print the codes;
- click anywhere in the "Code" column - right-click. The whole list will be immediately loaded into the editor (notepad) and displayed there.
- There, in the "File" menu group, select the "Print" option.

6.2.10 What is the code register - and why would you want to delete it?

First some important terminological distinctions:

- Codings are composed of text number, number of the beginning and end line of a text segment, and the actual code.
- The codes are stored in code files.
- When we speak of a code register, we mean the alphabetical listing of codes used in all files of a project.

On several occasions, AQUAD wants to know which of the many codes you want to refer to for analysis. This happens every time you attach codes to text files. If you prefer to select code names by clicking on them rather than writing them each time, AQUAD needs to keep a record of the codes used. Another case is when you want to get an overview of which codes were used to reduce which texts and how often. As a prerequisite for such a frequency analysis, AQUAD first wants to know which codes it should look for and count - all of them or just some critical ones for your question? Even if you want to count everything, AQUAD needs to know which codes you have used - otherwise it will not be possible to determine whether some of the codes may never have been used in some of the texts.

So you need a breakdown of all the code names. During coding, therefore, AQUAD does a kind of accounting of the code names you introduce as you work. Each time you use a new code name, it is automatically entered in an alphabetically sorted code register. If only a few codes are significant from the point of view of the type of analysis, you can simply select the codes from this code list created by AQUAD by clicking on them. If almost all codes are relevant for the analysis, such as counting, then delete only the few codes from the list that you do not want to use. Of course, your code register remains undamaged when you do this; you always work with a copy of the original directory.

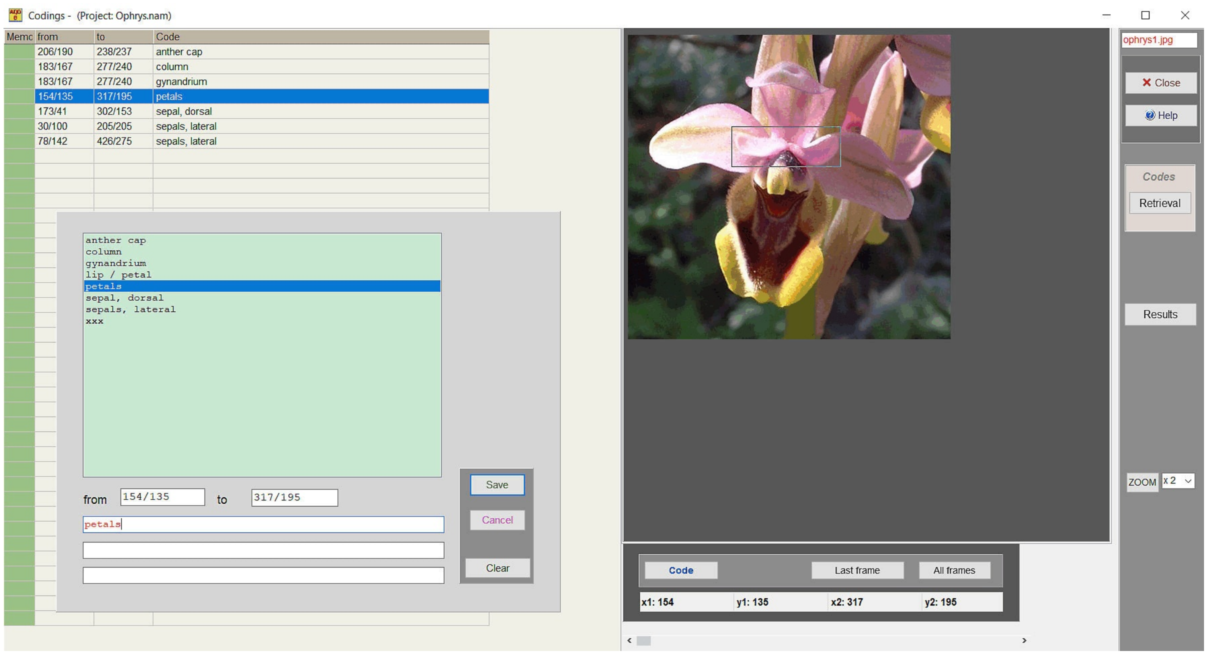
AQUAD creates the list automatically and also keeps you up to date automatically - so why would you want to delete the list? Well, you might run into the following difficulty: If you work with the possibility to replace codes with metacodes - and you should play with this possibility - it could happen that afterwards some code entries appear twice or more in the code register. Something similar can happen if you modify code files with a text editor outside of AQUAD.

While this does not cause any problems in terms of functionality, it goes against the logic and aesthetics of a code register. To clean up the code register, simply delete it (option in the "Edit codes" submenu) and then have it restored (last option in the same submenu). The new code register is created from all the codes associated with the files that are now read by AQUAD for control.

6.2.11 Encoding image files

The program windows and functions for encoding image files are the same as described above for encoding text.

On the screen shot below, we see the encoding list on the left, and the image to be encoded on the right. File segments are marked with the mouse (press the left mouse button at the upper left or lower right corner of the area to be marked, hold it down, drag the frame, release the button): In the photo of an Ophrys flower, the petals have just been marked (see figure); AQUAD enters the image coordinates and, after clicking the "Code" button, opens the coding window where the appropriate code can be selected from the register or newly entered. Clicking on "Save" saves the entire coding (coordinates of the image section and code) in the code file.


Codings - (Project: Ophrys.nam)

Memo	from	to	Code
	206/190	238/237	anther cap
	183/167	277/240	column
	183/167	277/240	gynandrium
	154/135	317/195	petals
	173/41	302/153	sepal, dorsal
	30/100	205/205	sepals, lateral
	78/142	426/275	sepals, lateral

```
anther cap
column
gynandrium
lip / petal
petals
sepal, dorsal
sepals, lateral
xxx
```

from 154/135 to 317/195

petals

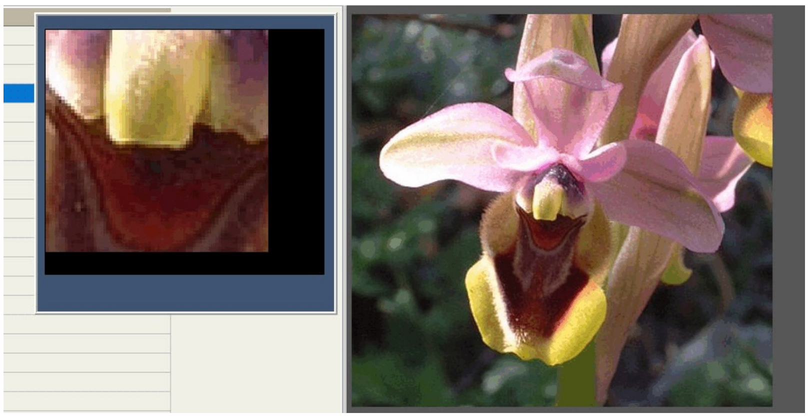
x1: 154 y1: 135 x2: 317 y2: 195

ophrys1.jpg

Conversely, if you click on a code name in the code catalog, the corresponding image segment is highlighted on the right. Deleting code entries is also possible after clicking on the coordinates of a code, as when working with texts.

Unlike the other file types, however, it is not possible to search for specific code structures and links of codes in the codes for image files. The reason for this is the somewhat complicated localization of data segments by specifying the x/y coordinates of the upper left and lower right corner points. So far, users have also not expressed a need for such functions when analyzing image data. However, coding structures and linkage hypotheses for texts that are available as image files (e.g., scanned handwritten texts) can be checked after all with a little trick (see next section below).

Since all images in AQUAD are reduced in size to the window format if they are larger in the original, a magnifying glass function is available for viewing details (see below). Moving the mouse pointer over the image selects the area to be magnified. The "Zoom" button switches the magnifying glass function on, and the second press switches it off again.

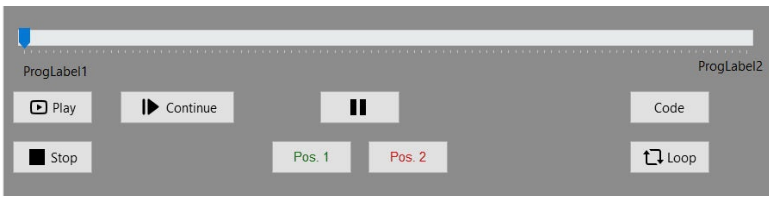


A rarely used function that provided quantitative information about image details was removed in version 8, but can be quickly reinstalled if needed: For some research questions, quantitative information from images can be interesting: For example, in children's drawings, how large was the father drawn compared to the mother and siblings? What percentage of area does a particular area of the picture occupy in the drawings of different children? To answer such and similar questions, in AQUAD 7 (with the function "Calculate height/width" in the function group "Tools" of the main menu under "Image codes"), the height and width of selected image segments (selection based on the encodings) could be calculated as the difference of the vertical and horizontal pixel coordinates, respectively, and inserted into encodings.

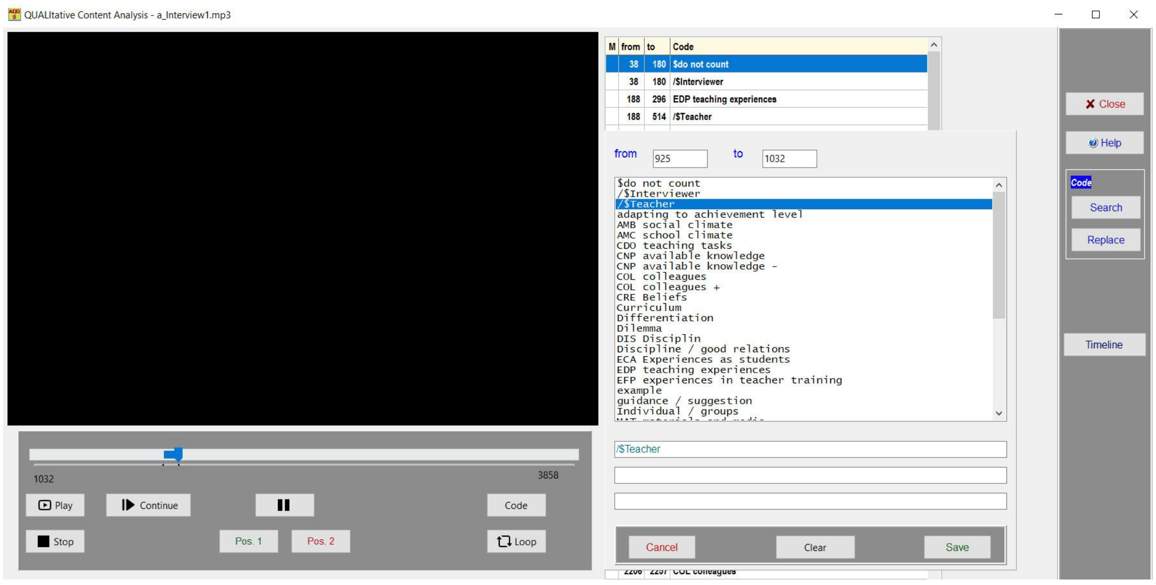
6.2.12 Encoding audio files

Working with the Mediaplayer

The program windows and functions for encoding audio recordings are the same as described above for encoding text. However, when encoding, you must use the media player control to move through the data.



In the figure below, we see on the right the columns for the encodings: "Memo" for memos ("X" signals that memos have been added here), "from" - "to" and finally the codes themselves. The unit of the audio recording is tenths of seconds; internally, the media player works with milliseconds as its unit, so there may be tiny deviations between the marker and the start/end of the recording sequence, but these are not disturbing in standard interview recordings. The screen capture shows the marked speaker code "/$teacher", the assigned data segment ranges from "925" to "1131".



M	from	to	Code
	38	180	$do not count
	38	180	/$Interviewer
	188	296	EDP teaching experiences
	188	514	/$Teacher
	2206	2207	COL colleagues

from 925 to 1032

```
$do not count
/$Interviewer
/$Teacher
adapting to achievement level
AMB social climate
AMC school climate
CDO teaching tasks
CNP available knowledge
CNP available knowledge -
COL colleagues
COL colleagues +
CRE Beliefs
Curriculum
Differentiation
Dilemma
DIS Disciplin
Discipline / good relations
ECA Experiences as students
EDP teaching experiences
EFP experiences in teacher training
example
guidance / suggestion
Individual / groups
```

/$Teacher

When the teacher started speaking in this passage, the buttons "Pause" , then "Pos1" (-> position 1) were pressed to mark the beginning (925), then "Next" and at the end of this interview segment, when the teacher stopped speaking (1132), the buttons "Pause" and then "Pos 2" were pressed. The current position of the player in the file is shown on the left under the slider and is transferred to the coding window - here we see the stop position "1132" after clicking on (end) position 2.

Entering single or multiple codes for the marked data segment is exactly the same as when working with text data.

Marked segments between positions 1 and 2 (see above) can be played again and again with the "Loop" button. Fine tuning is possible directly by changing the numbers in the "from" and "to" fields.

By clicking on a code in the corresponding column of the coding window, you can directly play back the coded section.

Fast movements through the recording are best done using the slider at the top of the control panel. Click on the tongue of the slider, hold down the mouse button and move the slider left or right with the mouse. The counter value changes analogously. The "Pos1" and "Pos2" buttons hold the beginning and end, respectively, of a data segment to be coded; the counter reading is transferred to the coding window. Clicking on a position (columns "from" and "to") in the coding table, a small window opens on the left side, in which you can remove (with the button "Delete") the entry again. Unlike when working with texts, in this window you can manually change the start and/or end position of the coded segment in the two input lines "from" and "to".

When you search for codes (cf. Working with text files), the location of each code is displayed in the coding list on the left in a green window. Then close the search window and click on the code displayed on the left in the coding window on the right - the coded passage will be played immediately.

The best thing to do is to open the sample project "a_Interview1" (post-interview audio recording of the transcriptions of the project "Interview") and play with the possibilities.

General strategy for encoding audio files

Unlike transcribed texts, one cannot quickly skim the content of audio files. Therefore, unless you are well acquainted with the content, you will need a lot of time if you want to find certain passages. We therefore recommend an encoding strategy that has proven successful in our own work:

First, organize the file formally, for example by speaker changes. In the project "a-interview1.mp3", therefore, the file segments of the interviewed teacher and the interviewer were first marked with speaker codes (/$teacher, /$interviewer).

Within the file segments, thematic emphases should already be recorded during this formal structuring - either in memos or in preliminary codings.

Therefore, it is recommended to immediately assign the appropriate speaker coding at the beginning of a new file segment – even if the end boundary of the segment is still unknown at this point. Otherwise, if a thematic-content encoding is applied within this segment, the initial boundary will change – and you will then have to laboriously search for the beginning of the segment again. So proceed like this:

- At the beginning of a new segment (speaker change, new question, etc.), immediately press the stop button and then set "Pos 1" (beginning boundary) and "Pos 2" (preliminary, false ending boundary). Click on "Coding" and set an appropriate (speaker) code.
- If you then know the exact end marker at the end of the segment, i.e. at the next speaker or topic change (press the stop key, read the number), click on the current (speaker) code with the incorrect end boundary in the coding table. Now you can transfer the numerical value in the "to" input field by deleting and typing or by copying and pasting from the control panel below (highlight there, click right mouse button, select "Copy").

Based on feedback from an AQUAD user, the following note may be useful: If pressing the "from" or "to" or "pause" or "stop" buttons on the audio player does not immediately interrupt playback - lag times in the range of seconds have been reported - please look for the cause not in AQUAD, but in your computer's system settings. The most likely cause, especially for computers with "sound on board" is that the driver software supplied or installed by the dealer and the hardware do not match. The simplest solution is to check the manual of the computer to find out the manufacturer of the "sound on board" component and then download the latest driver from the manufacturer's website on the Internet.

Transcribing important file segments

Direct coding of audio recordings without transcription saves considerable time, but important passages cannot simply be copied into the research report. Therefore, it is recommended to transcribe critical or significant passages, indicating the count in the media player, and collect them in an additional text file (e.g., the simple editor notepad.exe).

6.2.13 Encoding video files

When working with video recordings, everything applies mutatis mutandis what was described above for audio recordings. The crucial difference is that you will now find a video display above the control panel where the recording is played.

The recommendation for encoding audio data to first make a formal encoding pass (see above) in order to structure the recording according to speaker and/or scene changes applies analogously to video data.

6.3 How to edit texts and encodings?

6.3.1 How to edit texts

This is quickly clarified: Within AQUAD, one does not edit texts at all! AQUAD is designed as a program for analyzing texts. Changes to the texts, which are the basis of the analyses, may not be made in the process of analyzing them.

However, there may be reasons not to be quite so orthodox with the database. One may discover one or the other typing error, which one would rather improve than document until the publication of the research results. Sometimes you find that additional information would be useful for interpretation, such as bracketed references to nonverbal behavior in transcripts of video recordings of a discussion. (However, with AQUAD 8 of videos, you are likely to transcribe only selected scenes, but overall analyze the video directly!) Finally, in word-level analyses, one may want to exclude certain passages of text, such as the interviewer's questions or information identifying the speakers, time, location, and so on. In all such cases, the textual basis of the analysis must be edited.

Of course, this is always possible outside of AQUAD before coding begins. When doing so, please follow the guidelines given in Preparing Texts for Analysis with AQUAD and do not forget to save your texts back to ANSI format after editing using your word processing program.

However, if you have already coded the texts, you will have problems after editing the text. Editing may have changed the length of the text and the initial positions of the text segments in the encodings may no longer match the changed text. The codings are then incorrectly placed in the text from the first change. Therefore, the preparation of texts for content analysis should be thoroughly considered and planned before working with AQUAD, i.e., before coding.

6.3.2 How to edit codings

Remember, codings are composed of line numbers (beginning and ending line) of the coded text segment and the code words. You may have made a mistake when coding, or you may want to differentiate a very general code like "emotion" after the fact and enter "joy" or "fear" etc. instead. These codes can be edited when encoding text files as well as when encoding audio, video or image files:

- Within the encoding of "text" as well as of "image", "audio" or "video" you can delete encodings and add new codes.

- Within the coding of "Text" as well as of "Image", "Audio" or "Video" there is a special button "Search" in the "Code" field on the right side of the window. This button gives access to search and replace functions that help you edit specific codes in your file at once.

Please remember: AQUAD only allows you to edit codes associated with a data segment, it does not allow you to edit the segment contents!

Remember: If you want to check a code entry in the encoding list that is on the screen when encoding "Text", "Image", "Audio", or "Video", click on the code name to open a window for confirming or deleting the encoding or changing the segment boundaries of the code when encoding "Image", "Audio", or "Video".

When working with audio or video files, click the "Loop" button at the bottom of the audio or video player control panel. This allows you to listen to or watch the encoded data segment over and over again. When working with image files, clicking on the code in the encoding table marks the encoded image segment with a frame.

6.4 What is the purpose of metacodes?

If one follows the generalization strategy of coding, then in the first pass through the data of a project one invents a great many different codes in order to adequately capture the variety of content relevant to the research question. Soon, however, one reaches the point where it becomes necessary to look for commonalities within the multitude of codings, and then proceed to summarize and structure codes. However, one can also face this necessity if one proceeds quite un-strategically: When one views files for the first time (reads through, listens to, looks at), one usually finds many interesting segments that can be marked with many meaningful codes. Soon one runs the risk of losing the overview and therefore using different codes for similar units of meaning. One cannot avoid bringing order into the long list of different codes. In doing so, one quickly gets into trouble with the possibilities of manual editing, because now it is a matter of reorganizing the entire code system.

The principle of "constant comparison" contains a solution approach also for these problems. The approach consists of comparing the codes to see which of them represent similar meanings - and can therefore perhaps be understood as a group of sub-concepts that can be assigned to a more abstract superordinate concept. Such higher-level codes are called metacodes in AQUAD. If you think metacodes could provide clarity and structure in your text analysis, we recommend that you print out the code register, i.e. the list of all codes used so far (right-click in the code register window!) and mark with different colors all codes that belong together according to their meaning. Next, you should look for the name of a superordinate concept for each of the code groups, i.e. the metacode.

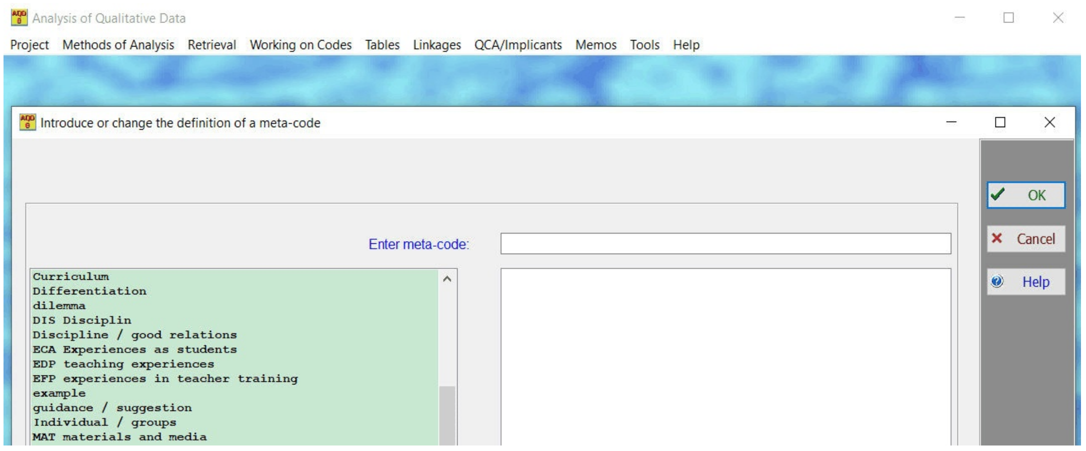
After this preliminary work, you can start defining the new metacodes to be introduced one by one. To do this, select the item "Metacodes" in the "Edit codes" submenu and then "Create/change definition". Of course, you answer the question whether you want to apply an already available definition with "NO" when creating a new definition. Now a new window appears: First enter the name of the new metacode in the line "Enter metacode". On the left you see the code register from which you can select all codes to be subsumed under a new metacode by clicking on the code name(s) in the light green box (code register) on the left. You can either transfer codes from the code register into the definition of your metacode by clicking on them, or put them back into the code register in the opposite direction by clicking on them.

When you click "OK", the definition of your new metacode, i.e. the metacode name and the list of child codes, will be saved for later use. Therefore, you will then be prompted to write a file name (without extension) under which the metacode definition will be stored. It has proven useful to store these definition lists under the name of the respective defined metacode with a prefixed extension, e.g. the definition of a metacode "Advice" (in which the codes "Advice", "Hint", "Help", "Suggestion", "Note" are to be combined) as "M_Advice".

6.4.1 Modify files: Add metacodes

If you select the item "Metacodes" in the submenu "Edit codes" and there the option "Add metacode", the new metacode will be added to the file in addition to each of the old codes with the same meaning. This way, the original codes will be preserved. However, your code file will get longer and longer, especially if you want to play with the possibility of generalization by introducing meta-codes – which you should definitely try. If you use this option frequently, you'd better choose the "Replace codes with metacodes" function, which, however, ensures that your original code files are preserved for further analysis.

To add metacodes you first have to select a metacode definition (see above) from the list of saved definitions (left in the figure), for example "Problem-Meta.mcd", in which all codes were summarized that refer to interview segments in which teachers talked about a wide variety of problems. Next to it, the new (meta)code name is shown at the top for control, as well as the codes subordinated to the new metacode. If we click on "OK", this metacode will be added to the codes for the critical file segments:



Enter meta-code:

```
Curriculum
Differentiation
dilemma
DIS Disciplin
Discipline / good relations
ECA Experiences as students
EDP teaching experiences
EFP experiences in teacher training
example
guidance / suggestion
Individual / groups
MAT materials and media
```

6.4.2 Modify files: Replace codes with metacodes

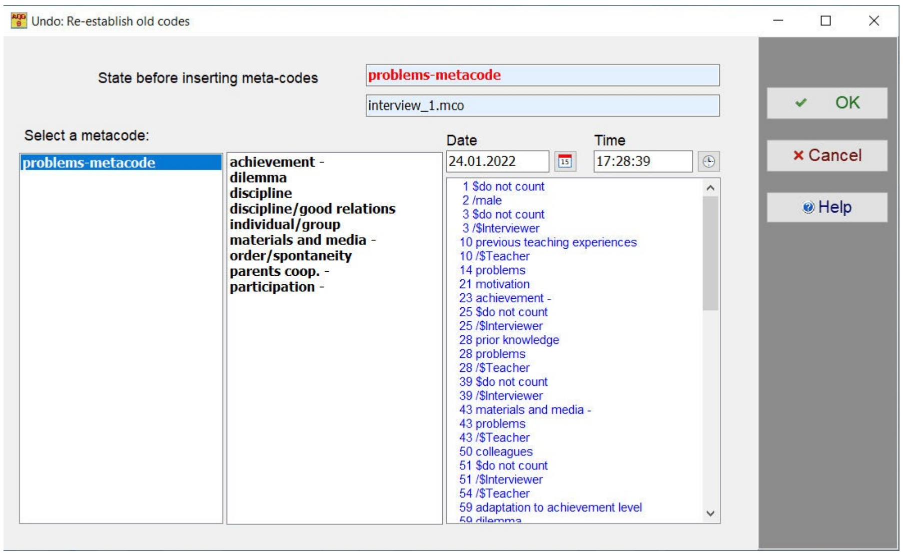
The procedure is the same as adding metacodes. However, after pressing "OK", the original codes disappear from the coding table and are replaced by the new metacode. However, the old coding files are still stored in the ...\mco\ path with the date and time of the change. Thus, your original codes remain untouched and can be reactivated for later analysis if needed.

However, AQUAD only stores up to 100 coding variants. From the 101st change by replacement, the old files - starting with the oldest - are overwritten in sequence. AQUAD will draw your attention to this so that you can still manually save the old codings in another directory.

6.4.3 Restoring previous codes

If you want to work with an earlier coding state again, select the item "Metacodes" in the submenu "Edit codes" and there the option "Undo: Re-establish old codes".

In the window that opens, select a metacode definition whose components are then shown in the middle window. Now you can restore the state before applying this metacode. In the window on the right you can see the state of the encoding of the first file in your project, above it the date and time of the modification with the selected metacode. Clicking "OK" loads the code files valid up to that time for all files in the project:



State before inserting meta-codes: problems-metacode

interview_1.mco

Select a metacode:

problems-metacode

achievement -
dilemma
discipline
discipline/good relations
individual/group
materials and media -
order/spontaneity
parents coop. -
participation -

Date: 24.01.2022
Time: 17:28:39

```
 1 $do not count
 2 /male
 3 $do not count
 3 /$Interviewer
10 previous teaching experiences
10 /$Teacher
14 problems
21 motivation
23 achievement -
25 $do not count
25 /$Interviewer
28 prior knowledge
28 problems
28 /$Teacher
39 $do not count
39 /$Interviewer
43 materials and media -
43 problems
43 /$Teacher
50 colleagues
51 $do not count
51 /$Interviewer
54 /$Teacher
59 adaptation to achievement level
```

It is a good idea to use the memo function when working with metacodes and keep a record in a research diary of exactly when which metacodes were introduced and used. With these records, it should not be difficult to recover the correct code files among the maximum of 100 stored coding states.

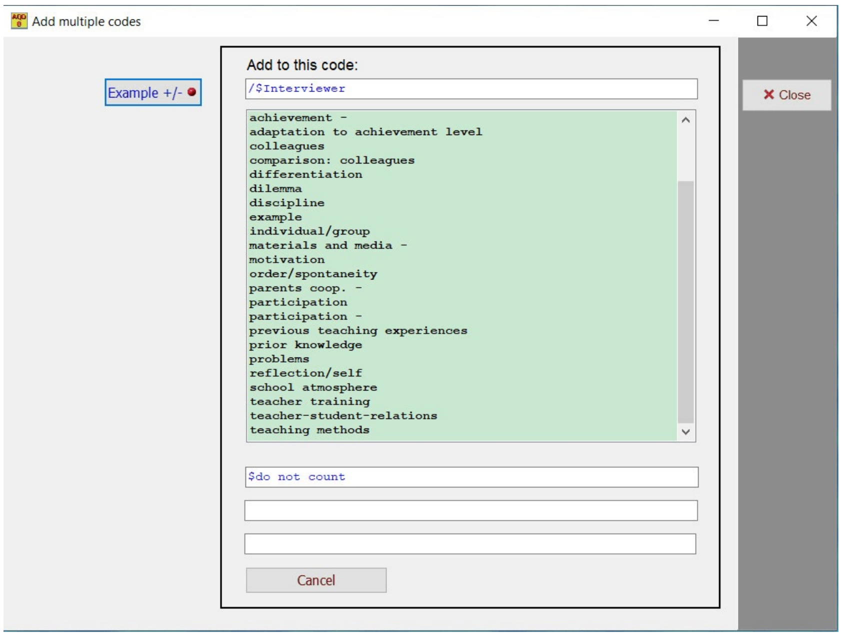
6.5 How to insert multiple codes?

During encoding, one can mark a selected file segment (text segment, video sequence, etc.) with up to three encodings at a time. This is useful, for example, if you want to mark the text segments of a certain speaker with his specific speaker code (/$...) and also with a conceptual code and additionally exclude them from counting ($not count).

Occasionally, however, the need to additionally encode certain already encoded file segments, i.e. to insert multiple encodings at the already encoded positions, emerges only later in the course of encoding. The coded file segments could be found quickly by "Search codes", but then one would have to start the function of the code input at each finding place individually again. Of course, one could also use the function "Add metacodes to codes" for this purpose in a slightly modified way and would then have the effect of automatic code insertion at previously already defined positions.

The situation is clearer with the special function "Insert multiple codes" (to be found in the function group "Edit codes"), because it does exactly what it promises: We first select from the code register by double-clicking the code to whose file segments (e.g. text sections) additional, multiple codes are to be added. Then we select the additional code(s) or write new ones in the input columns below, if we want to add codes not used yet. By clicking the "Insert code/s" button the function starts its work in all files of the active project or file directory.

In the screenshot we can see the master code list from the "Interview" project, where during coding it was decided to assign the control code "$do not count" to all the text segments that were previously only marked with the speaker code "/$Interviewer".



```
/$Interviewer

achievement -
adaptation to achievement level
colleagues
comparison: colleagues
differentiation
dilemma
discipline
example
individual/group
materials and media -
motivation
order/spontaneity
parents coop. -
participation
participation -
previous teaching experiences
prior knowledge
problems
reflection/self
school atmosphere
teacher training
teacher-student-relations
teaching methods

$do not count
```

If you press the button "Example +/-" (top left in blue font), the program enters an example code (/$Interviewer), which is supplemented at each occurrence by the code "$do not count" entered below. This way you could also add conceptual codes to the code "/$interviewer". Of course, a second or third code can also be inserted at once. When you click the button a second time, the example entries are deleted again and you can insert your own codes.

6.6 How to use keywords?

This has already been explained above. Remember the "Search" button in the "Keyword" field on the right side of the window when coding "Text". While you are reading your text files, you can use it to enable keyword searching.

You click on "Search" and a special window appears where you enter the critical word.

You start the search by pressing the "Search" button at the bottom of the window that appears and then using "Next" to jump from occurrence to occurrence. Or you click on "All" from the start; this will show you all the results of a search run at once. Assuming you proceed step by step ("Search" and repeatedly "Next"): As soon as the searched word is found, the corresponding line in the text window is highlighted. Maybe you are occasionally surprised about the result? It is important to remember a few rules:

- The search does not distinguish between upper and lower case letters when it searches for keywords in the text files.

- The search function also cannot put separated words back together. If you want to use the word search systematically, you should turn off automatic separation when transcribing the texts.

- If you have checked the "Part of words" option in the search window, the function will report the entered string regardless of its position at the beginning, within, at the end of a word or as an independent word. For example, if you let the program search for the word "for", it will then also find, for instance, the words "forbidden", "before" or "afford". Here's a hint: Let's assume you want to search for all text passages that mention "advantage" and "disadvantage". If you now simply enter the common component "advantage" as a search word, AQUAD will find what you are looking for in one pass – if the words occur in the text. However, you would also get compounds like "advantageous" or "advantageouslyt" displayed. You can exclude this by entering an additional space after the search word "advantage ". However, this again excludes that the critical word is found at the end of the sentence; because there a punctuation mark immediately follows, no space!

- If you accept the default option "Whole words" in the search window, the function will of course only find whole words that match your search input, but regardless of their position in the sentence.

Chapter 7: How to work with codes

7.1 How to find codes again

7.1.1 Prerequisite: Create a code catalog

As a prerequisite for searching for codes in all files (not only the currently opened file) of a project, the codes to be searched for must be grouped in a catalog. If you click on the "Code catalog" option in the "Retrieval" module, a window opens that shows the project's code register with a green background on the left, and an empty window for the new code catalog on the right. Above this is an input window where you enter a name to save the new catalog.

To select critical codes as contents of the catalog, simply click on the code name on the left, which is then displayed as catalog contents on the right. Of course, you can add multiple code names to the catalog. If selected by mistake, simply click on the unwanted code name in the catalog window on the right. The code will be deleted from the catalog and reinserted into the list on the left. Don't worry: the code register itself remains unchanged, AQUAD just uses a copy of the register here.

7.1.2 How to find coded text segments and particular codes

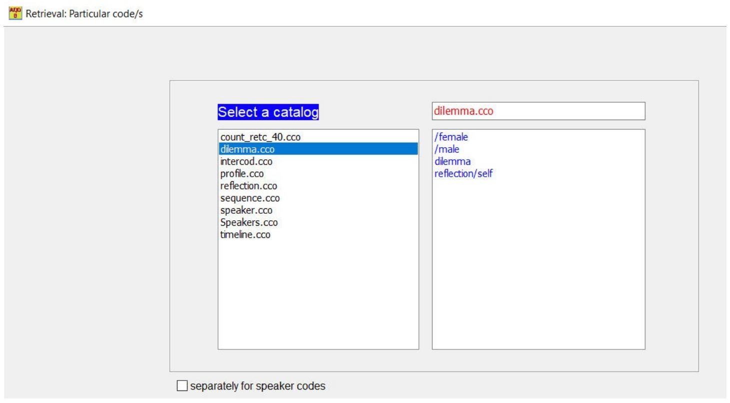
There are many reasons why a researcher might want to survey all text passages that deal with the same topic, i.e., text passages that fall under the same category, every now and then during the course of a project. One of these reasons, which plays a major role in many descriptive-interpretive studies (see already Tesch, 1990), is to detect commonalities across the entire data set. Another reason can be seen in the effort to control coding consistency, i.e., to ensure that codes have always been used according to the same principles. Other reasons can be found in looking for evidence of where a code has been used correctly and where it has been used incorrectly, where there are parallels between certain texts, and so on.

AQUAD extracts the text segments it is looking for from all texts. This is done in the order in which the texts are listed in the file directory. Because one text is searched after another, we also call this search for relevant text segments one-dimensional or linear analysis. It differs from two-dimensional analyses, called table or matrix analyses in AQUAD, and from analyses of complex links. The results of one-dimensional analyses can be read on screen, printed on paper, or initially saved to disk.

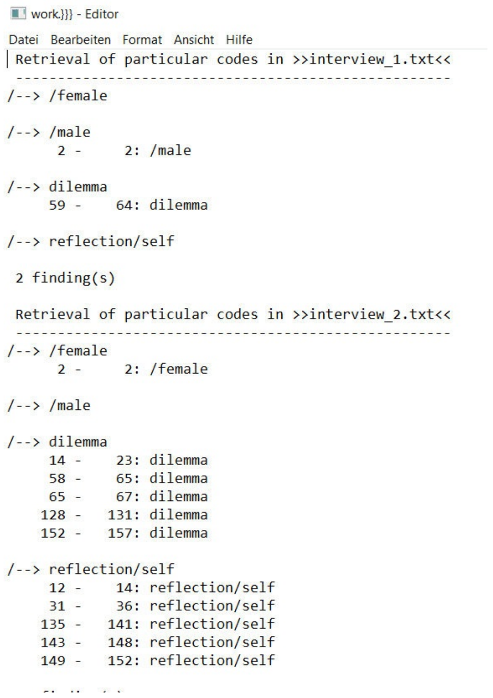
Search for particular codes

The linear search for coded text segments is started in the main menu item "Search" with the option "Particular code". A window opens where you can decide what to search for by selecting one of the available code catalogs.

In the example on the following page we worked in the project "Interviews" which includes the files "interview1.txt" to "interview4.txt". For this, among other things, a catalog with four code names was created, containing the profile codes "/male" and "/female" and two conceptual codes "dilemma" and reflection/self". This catalog "dilemma.cco" ("cco" for "catalog of codes") has been selected for the example by clicking on it:



A click of the button "OK" immediately returns the search result:



```
Retrieval of particular codes in >>interview_1.txt<<
-----------------------------------------------------
/--> /female

/--> /male
      2 -     2: /male

/--> dilemma
     59 -    64: dilemma

/--> reflection/self

 2 finding(s)

 Retrieval of particular codes in >>interview_2.txt<<
-----------------------------------------------------
/--> /female
      2 -     2: /female

/--> /male

/--> dilemma
     14 -    23: dilemma
     58 -    65: dilemma
     65 -    67: dilemma
    128 -   131: dilemma
    152 -   157: dilemma

/--> reflection/self
     12 -    14: reflection/self
     31 -    36: reflection/self
    135 -   141: reflection/self
    143 -   148: reflection/self
    149 -   152: reflection/self
```

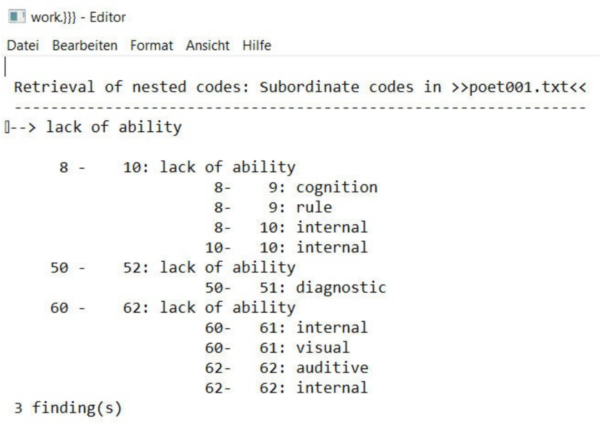
We do not want to discuss the findings at this point, but are only concerned with the way they are presented: We see that Interview 1 was conducted with a teacher ("/male") who talks about a "dilemma" only once, but never on himself. Interview 2 is from a female teacher ("/female") who talks about "dilemma" and reflects about herself in five text segments. The result appears in the text editor and can therefore be printed, saved, copied completely or in excerpts in the module "File" (top right in the menu bar of the editor). The original result contains, of course, findings from all four interviews, but for reasons of space only two have been copied here.

7.1.3 How to search for code structures

Search for code structures: embedded codes

This option searches for whether sub-codes are still enclosed within the line boundaries of a code. Or in other words, the option captures all codes that include other codes within the assigned text segment. This menu option is therefore also suitable for searching for hierarchically structured coding sequences. Codes with identical start and/or end lines are displayed as well, as long as their text segments do not overlap (see below).

Example: In our example text "Poet" some segments were coded with "lack of ability". At a later stage, when coding, we paid attention to details and noted what the lack of ability was, for example, visual or auditory deficiencies. Now we want to relate the two. For example, a search might show that particular segments of text were coded as "lack of ability," while within these segments the codes "cognition", "diagnostic", "visual" and "auditive" had been assigned to a smaller section. AQUAD provides information about this along with the text of the segment as follows:


work.}}} - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
 Retrieval of nested codes: Subordinate codes in >>poet001.txt<<
 --------------------------------------------------------------
▯--> lack of ability

      8 -    10: lack of ability
                  8-    9: cognition
                  8-    9: rule
                  8-   10: internal
                 10-   10: internal
     50 -    52: lack of ability
                 50-   51: diagnostic
     60 -    62: lack of ability
                 60-   61: internal
                 60-   61: visual
                 62-   62: auditive
                 62-   62: internal
 3 finding(s)
```

We see that in the first text deficient ability is diagnosed in general (lines 50-52) and differentiated as cognitive, visual and auditive deficiencies. (*Not copied here:* In the second text, the critical code does not occur at all, while in the third text, the protagonist himself (speaker code "/$Poet") attests to auditory and visual deficiencies.)

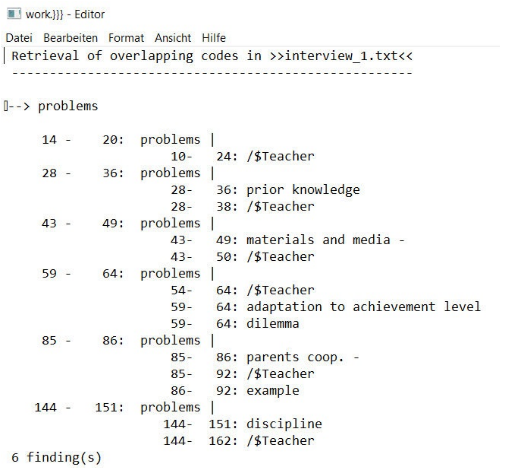
Moreover, AQUAD can reveal hierarchical coding structures according to two directions:

- from higher-level codes, the lower-level encodings are searched (as in the example above);
- from subordinate codes, the higher-level codes are searched, whose text segments thus enclose the segments of the subordinate codes.

Searching for code structures: overlapping codes

As a result of this search strategy, all codes (as always with information about the "location") are displayed, with which that were used to interpret overlapping text segments. Here we are informed about the text segments as a whole, not only about the narrower part where codes overlap.

Example: We again take our example "interview" texts and look for text segments that intersect with those coded "problems".


work.}}} - Editor

Datei Bearbeiten Format Ansicht Hilfe

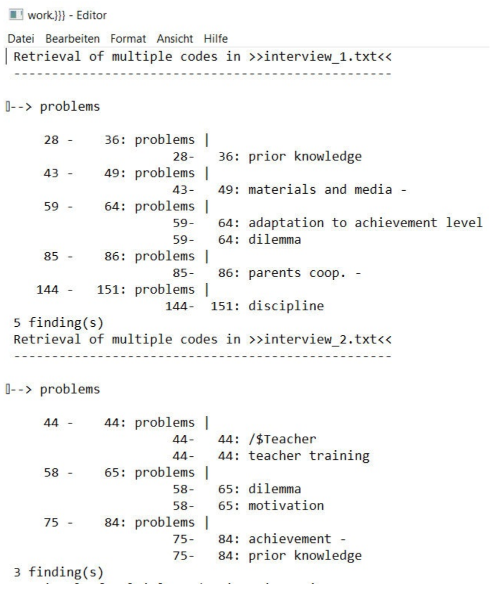
```
 Retrieval of overlapping codes in >>interview_1.txt<<
 -----------------------------------------------------

▯--> problems

     14 -    20:  problems |
                       10-    24: /$Teacher
     28 -    36:  problems |
                       28-    36: prior knowledge
                       28-    38: /$Teacher
     43 -    49:  problems |
                       43-    49: materials and media -
                       43-    50: /$Teacher
     59 -    64:  problems |
                       54-    64: /$Teacher
                       59-    64: adaptation to achievement level
                       59-    64: dilemma
     85 -    86:  problems |
                       85-    86: parents coop. -
                       85-    92: /$Teacher
                       86-    92: example
    144 -   151:  problems |
                      144-   151: discipline
                      144-   162: /$Teacher
 6 finding(s)
```

In the text "interview_1.txt" this code was used 6 times (the remaining findings are not copied because of space), first in general terms, then referring to (students' lack of) prior knowledge, lack of materials and media, difficulties to adapt teaching to students' achievement level, low cooperation of parents and discipline.

Please, note: "overlapping" is nothing more than a physical overlapping of text passages - without necessarily semantic references.

One more note on profile codes (see above): profile codes characterize the whole text, for example, the gender of the speaker ("/female"). Nevertheless they should be inserted if possible only in one line, expediently at the text beginning, otherwise they would overlap with all codes when searching for overlaps without providing relevant information.

Search for code structures: Multiple codes



```
Retrieval of multiple codes in >>interview_1.txt<<
--------------------------------------------------

▯--> problems

    28 -    36: problems |
                    28-   36: prior knowledge
    43 -    49: problems |
                    43-   49: materials and media -
    59 -    64: problems |
                    59-   64: adaptation to achievement level
                    59-   64: dilemma
    85 -    86: problems |
                    85-   86: parents coop. -
   144 -   151: problems |
                   144-  151: discipline
 5 finding(s)
 Retrieval of multiple codes in >>interview_2.txt<<
 --------------------------------------------------

▯--> problems

    44 -    44: problems |
                    44-   44: /$Teacher
                    44-   44: teacher training
    58 -    65: problems |
                    58-   65: dilemma
                    58-   65: motivation
    75 -    84: problems |
                    75-   84: achievement -
                    75-   84: prior knowledge
 3 finding(s)
```

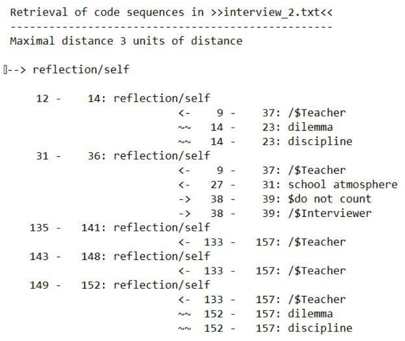
This search strategy is used to find all file locations associated with more than one code. Let's assume that it is not completely certain, we always set the code "problems" when coding teacher interviews if the interviewee talked about difficulties in a text segment. However, since we were not sure whether certain foci of difficulty would not emerge in further interviews, we tried to code each time the same text segment with a code indicating the type of difficulty, e.g., "problems" and also, for example, "students' lack of knowledge".

At a later stage, we realize that instead of stating "problems" in general, we should better differentiate specific problem areas. Now we look for which reported difficulties we have labeled additionaly with "problem" and with which other codes at the same time (Results for "interview-1.txt" and "interview_2.txt"):

Search for code structures: Code sequences

A helpful first step in approaching the reconstruction of meaning contexts in texts is to find out which statements frequently occur in the neighborhood of other statements. We determine a critical code as a reference point for finding code sequences. AQUAD informs about all combinations of codings for file segments that occur at a maximum distance (which we define in addition) from the code that is the reference point. "Embedding" and "overlapping" are displayed as well.

Example: We want to quickly find out what other codes are in the vicinity of each file segment coded with "Reflection/Self". We define the maximum distance to be 3 lines (distance units) before or after the critical file segments. This means that AQUAD will inform about all codes associated with file segments that end at most 3 lines before the content marked with the critical code "Reflection/Self", and about all codes associated with file segments that start at most 3 distance units after the last line/time unit/image of the critical segment (in the following figure you can see only results for "interview_2.txt"):



```
 Retrieval of code sequences in >>interview_2.txt<<
 --------------------------------------------------
 Maximal distance 3 units of distance

▯--> reflection/self

    12 -    14: reflection/self
                        <-     9 -    37: /$Teacher
                        ~~    14 -    23: dilemma
                        ~~    14 -    23: discipline
    31 -    36: reflection/self
                        <-     9 -    37: /$Teacher
                        <-    27 -    31: school atmosphere
                        ->    38 -    39: $do not count
                        ->    38 -    39: /$Interviewer
   135 -   141: reflection/self
                        <-   133 -   157: /$Teacher
   143 -   148: reflection/self
                        <-   133 -   157: /$Teacher
   149 -   152: reflection/self
                        <-   133 -   157: /$Teacher
                        ~~   152 -   157: dilemma
                        ~~   152 -   157: discipline
```

• Codes marked with "<-" start at a distance of at most three units (here: lines) before the critical segment (but: embedding is also signaled),
• codes marked with "~~" are located within the critical segment,
• codes marked with "->" denote segments starting at most three units after the critical segment.

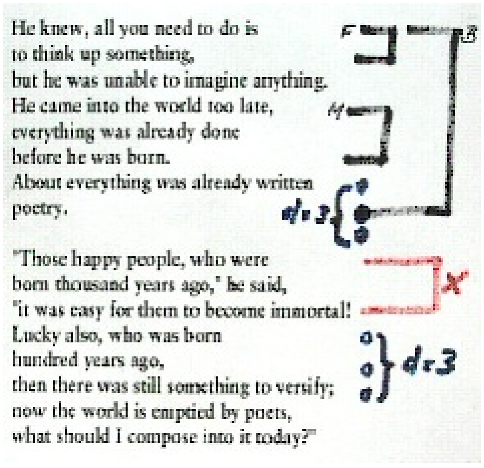
An important function of AQUAD is to find relationships between significant events or categories in the data, or to check whether such connections exist. The program uses a number of algorithms in the "Connections" module that are based on the principle of backward deduction. That is, AQUAD checks assumptions, how codes could be related to each other and searches in all files, whether accordingly critical codes can be found in certain sequences and distances.

Each coded file segment can become evidence of a corresponding category within your file. As you read your texts, listen to audios, watch videos, structure them, compare them, and glance at your memos now and then, you will develop assumptions about relationships between some categories that you use in your analysis. Such assumptions about relationships can lead to hypotheses about latent file content. Searching for coding sequences assists in generating such hypotheses.

7.1.4 Excursus: Distance between file segments

For hypotheses about linkages of Codes or the encoded file segments, the distance between these segments plays an important role. Staying with the examples of interview texts we have worked with so far in this chapter, let us assume that we have reason to believe that some speakers start thinking about themselves whenever they talk about a problem in their everyday practice. Specifically, we note that a "problem" is talked about in a certain text segment and now we expect that (in the example project "Interview") it is followed by a segment coded as "Reflection/Self".

Since the program cannot perform a semantic analysis of the content context, the only approximate solution is to narrow it down by specifying the maximum permissible distance between the segments in the text flow. Talking about "oneself" five minutes (or three pages further down in the transcription) after an interviewee came to talk about contradictory behavioral possibilities (e.g., spontaneity of students vs. rules in the classroom) will most likely not be directly related to the problem mentioned. If it is, then the interlocutors usually also explicitly pick up a past thought - and we would have to mark this reference here again with the code "problem". As a default value, three "units" are specified in the analyses, here lines as the maximum distance. File segments, which start with this setting only four lines after the end of the first segment in the presumed sequence "Problem - Reflection/Self", are not considered as finds - even if they should be coded with "Reflection/Self". So we have to experiment with the set maximum distance to make it fit the characteristics of our data and/or check the context at the original text.



He knew, all you need to do is
to think up something,
but he was unable to imagine anything.
He came into the world too late,
everything was already done
before he was born.
About everything was already written
poetry.

"Those happy people, who were
born thousand years ago," he said,
"it was easy for them to become immortal!
Lucky also, who was born
hundred years ago,
then there was still something to versify;
now the world is emptied by poets,
what should I compose into it today?"

In the example from the file "poet001.txt" shown here, the file segment "He had come into the world too late, everything had been anticipated before he was born" is marked with "Code B". Three lines later begins the segment "The happy people who a thousand years ago..." marked with "Code X". With a distance setting of d=3, the program would still accept both segments or their codes as contiguous here.

With text files the meaning of the distance specification is clear: maximum tolerated distance in number of lines between the end of the first and the beginning of the searched second segment. But what do the numbers mean for audio or video files?

Audio files:

The media player counts the flow of audio in tenths of seconds. For the distance check, the count is multiplied by a factor of 10, i.e. the maximum distance is checked in units of seconds. Accordingly, the default setting d=3 causes all relevant encoded file segments that begin a maximum of 3 seconds after the first segment found to still be accepted as "associated" with it.

Video files:

The media player counts the video flow in frames, i.e. single frames. For the distance check, the count is multiplied by a factor of 25, i.e. the maximum distance is checked in units of seconds. The default setting d=3 therefore has the effect that all relevant coded file segments that start a maximum of 3 seconds (or 75 frames) after the first segment found are still accepted as "associated" with it.

7.1.5 Codes NOT used

This option in the "Retrieval" menu initiates a one-dimensional search. Sometimes, in a large project, it is quite helpful to know which codes are not used in conjunction with which files. AQUAD uses the current contents of your code register or a selection of codes as criteria for this search.

7.1.6 Counting codes and entering frequencies in tables

With information about the frequency of selected codes in a project, many quantitative analyses are possible that are quite useful, depending on the research question. Frequency tables for export to programs for quantitative analysis (e.g., spreadsheets and statistical calculations) can be created for all codes, including, for example, sequence codes, which usually arise only in the course of an analysis. In any case, the starting point is counting the frequency of codes with the "Count codes" function in the "Search" submenu. The result, a simple frequency table of the counted codes, is automatically saved under the name of the code catalog used in the subdirectory ..\res. In the case of the "Problem" code catalog, we can find the frequency list in this directory as "Problem.txt".

For further statistical processing of the frequency information, this list is automatically converted into a structured table and also stored in the "..\res" directory. The columns are defined by the codes, the rows contain the frequency values of these codes in the individual cases (files). Again in the case of the code catalog "Problem" we find in this directory the frequency table as "Problem_DT.csv". The name part "_DT" stands for "data table", the extension ".csv" for "comma separated values" (in fact the semicolon is used as separator). These CSV tables have a standardized structure, so that they can be exported without problems into programs for spreadsheet analysis (e.g. Microsoft Exel) or for statistical analysis (e.g. SPSS).

7.2 How to find correlations between codings

Helping researchers find and examine meaning relationships in data is the central function of AQUAD. The program does this by using algorithms that implement the logical principle of "backward deduction." With this, AQUAD searches the code files and checks whether the elements of a suspected connection can be confirmed in the correct order and distance. According to Glaser and Strauss, each of the many codes in a project represents an analytic category: "A category stands for itself as a conceptual element of theory" (Glaser & Strauss 1967, p. 36). Each coding or code-marked data segment is evidence for the occurrence of the corresponding category in the data. As you read, organize, compare the data, and look at the code list and perhaps related memos, you probably come up with conjectures early in the analysis process about relations that seem to exist among some of the categories in your analysis. "Whenever these people talk about critical life events, you come up with emotions ..." you may notice. Such linkages can become the occasion for formulating hypotheses or initial theorizing about what the data are actually about.

It has already been outlined above that in practice one always combines inductive and deductive thinking movements. For example, one can start from hypotheses about possible correlations and try to prove them on the basis of the data, i.e. deduce evidence for these hypotheses from the data. If nothing can be found to confirm the general assumptions or hypotheses about the relationship between the categories, or if contradictions are encountered, then an inductive effort is made to develop categories and their systematic relationships from the data, i.e., to generalize basic relationships of meaning from distinctive data segments. This is, of course, an extremely abbreviated representation of the inductive-deductive procedure in the analysis of qualitative data.

In any case, however, one comes to the point where one formulates a slowly generated conjecture or linkage of categories sought from the outset as a statement about that very linkage. In other words, one formulates a hypothesis. In the analysis of qualitative data, such a hypothesis is basically an assertion that certain coded data segments or the categories to be recognized in them occur in combination. As an example, consider the conjecture that arose in the analysis of biographical interviews: "There is a link between critical life events and emotional reactions."

In principle, checking how codes are related is very simple. You enter the codes that you think are related in a certain way, and then the computer starts looking to see if there are segments in your files that are coded in that way. That is, the computer looks for codes that they assume are specifically linked and the computer looks for features that are indicative of such linkages. The advantage of the computer is that it doesn't miss anything in the process. It can be relied upon to detect and consider any relevant coding in the data.

To adapt this search principle to your needs, AQUAD offers a number of variations. More generally, there are four ways to check links between codes or test your hypotheses. These four possibilities are presented below, organized according to different research strategies. Here AQUAD looks for

• Connections in the context of unconditional categories, i.e., in a line range to be specified before and/or after the text segments of a critical category. Within this text area, the occurrence of any other encodings is registered without requiring them to satisfy any other additional condition.

• Connections in the context of conditional categories. The "condition" for the possible presence of systematic correlations is defined by a second category to which a data segment must be assigned simultaneously.

• Connections in the form of simple coding sequences. Such sequences have been built into AQUAD during its development on the occasion of concrete investigations. You can simply insert your codes into the abstract linkage structures.

• Interrelationships in the form of complex coding relations. For verification, you can put the relations in the form of testable hypotheses about coding sequences you suspect in your data.

7.2.1 Context of unconditional categories: Searching for codes

You have already become acquainted with this simplest form of non-linear search or testing in data texts in the presentation of the various ways in which coded data segments or the codings used in them can be retrieved with the help of general or specific search structures.

The computer receives as default for the search work only the order to register all codings in certain distance around the code just found ("distance analysis"). Somewhat more complex is the order to search for specific coding patterns, e.g. multiple coded data segments. In specific search, the restriction is still made that only those patterns are reported in which a specific code is represented. So the "category" in the general case is a specific coding structure.

What is not required here for search success is that the data segment belongs to the scope of (at least) one other category. For example, the search algorithms described in chapter 7.1.2 could not distinguish between overlaps of other data segments with the scope of, say, the category "critical life event" in the interviews of "younger persons" and "older persons". This would be a search for "conditional" categories (here conditioned by the characteristic "age").

7.2.2 Context of conditional categories: Table Analysis

Miles and Huberman (1994) recommend and use in their book the matrix representation form as the main strategy for structuring textual content. In particular, they are concerned with making the sequential configuration of statements, thoughts, opinions in texts more manageable by transforming them into simultaneous configurations. According to Miles and Huberman (1994, p. 93 f.) matrices or tables help to

- to gain or keep an overview, since data and analysis are presented in a summarized way;
- quickly identify where additional, more sophisticated analysis is needed;
- to compare data as well as interpretations;
- to communicate the results of the investigation to others and make them understandable.

Further guidance on organizing verbal data in tables or matrices, along with numerous examples, can be found in Miles and Huberman (1994, chs. IV, V, and VI). For computer-aided retrieval, one constructs coding matrices (Miles & Huberman, 1994) or coding tables (Shelly & Sibert, 1992). The twofold determination of the contents (data segments) of the cells of such a table determines the interpretation of the findings more strongly than the mere finding of an accumulation of spatiotemporal proximity of initially independent categories. On the other hand, the construction of an analysis matrix requires more preliminary conceptual work, consequently greater progress in the analysis process.

Example: In the analysis of interviews with teachers, we compare interviews with primary and secondary teachers to demonstrate table analysis. Using school level as a "singular" characteristic, we label the columns of our table. We introduce the profile codes "/primary" and "/secondary" for this purpose. We want to test the assumption that in terms of school atmosphere (conceptual code "social climate"), view of one's own position ("role as a teacher"), problems in class ("problem"), and reflections on oneself ("reflection/self") there are differences between teachers of the two school levels. (Of course, we would have to conduct more and systemtically selected interviews for this; however, for the presentation of the principle, these will suffice here). The codes "social climate," "role as teacher," "problem," and "reflection/self" define the rows of the table. When we start the table analysis, AQUAD will gather all file segments in which primary teachers talk about school atmosphere; in the column next to it we read statements of secondary teachers about the same topic. In the row of the table below we first find statements of primary teachers about their role as teachers, in the column next to it the corresponding statements of secondary teachers, and so on.

Such matrices are particularly congenial to the demands of structuring data because they structure and present extensive information simultaneously (rather than sequentially as in the original files), allow rapid comparisons with results on other subjects, situations, time points, investigations, etc., and open up new, more refined analyses. If it is true that a picture is worth 1,000 words, then a tabular presentation supersedes at least 100 words, especially since it can visualize the data and the analytic process together.

However, a restrictive note seems to be appropriate right here: Even a highly differentiated representation of the reduced data, prepared in a table structure, cannot do more than provide good preconditions for further conclusions, but by no means replace necessary conclusions. The matrix representation also reaches its limits when not only singular characteristics such as school level or gender are used as ordering criteria, but when specific combinations of meaning units are searched for, each of which can occur several times in a data file. For such higher-order evaluations, complex patterns of conditions must be specified, usually involving multiple codes or units of meaning and their logical relations, sometimes including quantitative, e.g., socio-demographic, data. A two-dimensional matrix would be overwhelmed by the demands of such analyses, and an arrangement of more differentiated or specialized matrices would quickly become unmanageable.

The various functions of the table analysis can be found in the main menu item "Tables". You should carefully study the use of the functions "Create table" and "Edit table". The next three options "Analysis: Frequencies", "Analysis: Coding" and "Analysis: Text segments" determine in which form you get the results. You can choose between three options: simply specify the frequencies of the codings, specify additionally the locations of the codings, or produce even a complete printout of the coded text segments that meet the conditions of the table construction.

How to create a table

In order to construct a table for a meaningful two-dimensional data search in AQUAD, you need at least two files that differ in a profile feature (or "singular" feature). One can only search for or determine the frequency of segments or codes with a matrix if the file segments have been interpreted with at least two different codes.

One of these codes must be a socio-demographic or profile code of the entire text, i.e., a singular code that is assigned only once in a text and characterizes the entire text. In the example of the interview analysis, a category has been introduced whose characteristics could each be represented by a profile code: "/primary level" and "/secondary level". Both are characteristic for the interview partner respectively his text as a whole and can be used to define the columns. In our example project "Poet" we used "/Part_1", "/Part_2" and "/Part_3" as profile codes.

The other code(s) defining the rows of the table must belong to the conceptual codes of the study. They may and probably will occur more than once per code file. In the interview example, we selected the codes "Social Climate," "Role as Teacher," "Problem," and "Reflection/Self" as interesting for analysis. In our example project "Poet", for example, we could use the codes "acting", "auditory", "visual", "emotional", "taste", "thinking", etc. to define the table rows to see if there are differences between the parts of the story in terms of external activities vs. internal activities.

Now, every time AQUAD finds one of these codes in a code file, it will (if the "Analysis: Text Segments" option is selected) also output the associated text segment. First, however, we need to figure out how these codes are entered: After activating the "Create table" option, the following window will appear. You will see two round buttons in the upper left corner: "Columns" and "Rows".

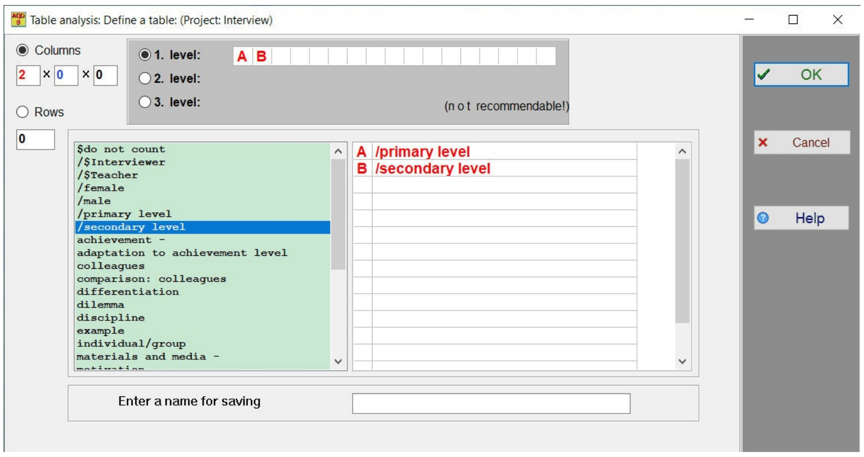


Table analysis: Define a table: (Project: Interview)

(● Columns) 2 × 0 × 0

○ Rows 0

(● 1. level:) A B
○ 2. level:
○ 3. level: (n o t recommendable!)

```
$do not count
/$Interviewer
/$Teacher
/female
/male
/primary level
/secondary level
achievement -
adaptation to achievement level
colleagues
comparison: colleagues
differentiation
dilemma
discipline
example
individual/group
materials and media -
```

A	/primary level
B	/secondary level

Enter a name for saving

The screen shot shows that "Columns" is already activated (there is a small dot in the button) and the profile codes "/primary level" and "/secondary level" have been set as column headings (by clicking on them in the green code register).

Please, keep in mind:
You can only use profile codes as column headers (indicator: "/"; also called singular codes here)!

In our example project "Interview" we clicked on the profile codes "/primary level" and "/secondary level". Both were then transferred to the white box on the right, and in the small cells that follow behind the label "1st level", the abbreviations "A" and "B" are automatically entered. As you can conclude from the number of cells, a maximum of 12 column headings are possible on the 1st level. At the same time, the first three count boxes on the left are automatically filled with numbers – first "1" appears, then "2": the column headings (= analysis conditions) are counted.

If we have a lot of text files, we can differentiate the column headings into a maximum of six "2nd level" headings. Thus, in the example of the interview analysis, we could additionally subdivide the categories of school level by gender or by professional experience (beginner, practical experience, expert) of the teachers. In this way, we create 2x2 or 2x3 columns. However, for such an analysis we would need a larger number of files that can be assigned to the determinants of the columns. In the example project "Interview" we have only four texts or cases, so differentiation would not make much sense here. Sometimes, when there are a lot of texts available, one can even think of a third level of differentiation - this is possible in AQUAD in principle, but usually not very advisable because of the difficulty of interpreting the results.

After we have defined the column headings, it is the turn of the rows. Click on the "Rows" button and then select appropriate conceptual codes, for example "problems", "school atmosphere", etc.

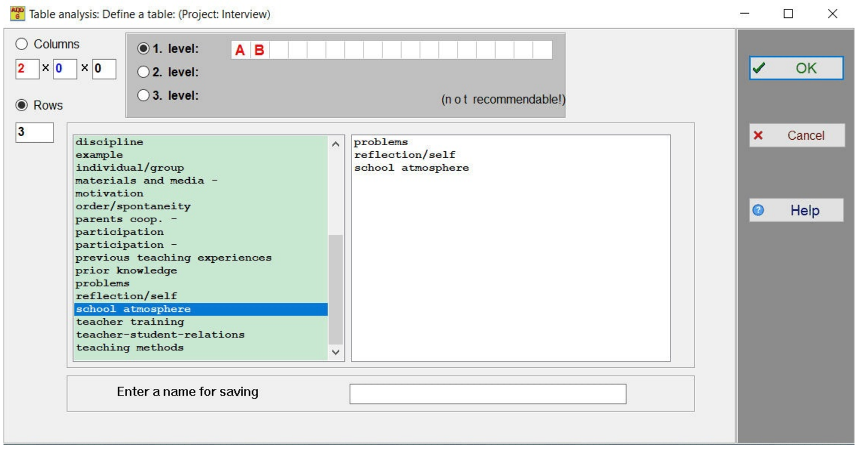


Table analysis: Define a table: (Project: Interview)

Columns: 2 × 0 × 0

1. level: A B

2. level:

3. level: (n o t recommendable!)

Rows: 3

```
discipline
example
individual/group
materials and media -
motivation
order/spontaneity
parents coop. -
participation
participation -
previous teaching experiences
prior knowledge
problems
reflection/self
school atmosphere
teacher training
teacher-student-relations
teaching methods
```

```
problems
reflection/self
school atmosphere
```

Enter a name for saving

The selection is again automatically transferred to the white box on the right, and the number of lines provided is counted in the small box below the "Lines" button.

Now we are almost ready to press the "OK" button. But at the bottom of the window there is a small box where we are asked to enter a name (without extension!) to save our table definition. After clicking "OK" all entries will be saved for later table analysis.

We suggest that you try all this out once. If you want to see what the window looks like after you have labeled columns and rows, first go to "Edit table" (in the "Tables" submenu) and select the table definition "example.ata", which was also copied to your hard disk when AQUAD was installed. Jump back and forth between the "Columns" and "Rows" options to take a look.

Summary:

① You click the "Columns" button and select the first level of column headers.
② You enter up to twelve socio-demographic (profile) codes as column headings.
③ If necessary for the analysis, add another level with corresponding column headings (subordinate profile codes).
④ You click on "Rows" and select conceptual codes as row headings.
⑤ Enter a name to save this table definition and then click "OK".

How to edit tables

In principle, the same sequence of steps applies as described above for making a table or matrix. However, you must first decide which of the already created table definitions you want to edit.

If you do *not* want to overwrite the table design you have just edited, please enter a new name to save it in the small input field below (before pressing "OK").

How to perform a table analysis

You have three options, which in principle work the same, but give different informative results.

- The most economical, but also the least informative is the "Analysis: Frequencies". For this, the result fits on the screen or a sheet of paper, and we can see at a glance how often the conceptual codes occur in the texts under the condition of the profile codes. However, where exactly and in which utterances remains hidden from us with this function. Mostly, frequency analysis is therefore used as a heuristic in the search for connections. Let's take a look at what a frequency analysis the table definition created above provides in our example project "Interview" (If needed, the results can be saved in csv format as described above):

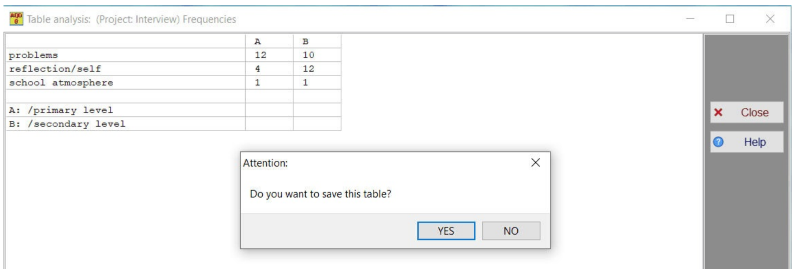


Table analysis: (Project: Interview) Frequencies

	A	B
problems	12	10
reflection/self	4	12
school atmosphere	1	1
A: /primary level		
B: /secondary level		

Attention:

Do you want to save this table?

- More detailed is the "Analysis: Codes". It provides a list in which the codings found there (i.e. line numbers and code names) are entered one after the other for each cell of the table definition.

- The most detailed results are obtained with the "Analysis: Text segments" option. This option returns all text segments that satisfy the dual conditions of the matrix design. Therefore, please note: If you want to output the results to the printer, the paper consumption can be considerable! Neither printer nor screen can display the columns of the table filled with text documents side by side. AQUAD therefore prints one cell at a time for each column. You can then assemble the paper printouts into a table on a large blackboard or other suitable surface if you want to see the result in its entirety.

7.3 Connections in the form of simple coding sequences: Checking linkages

For theory-generating or theory-constructing analyses, the program module "Linkages" is probably the most important in the AQUAD program package. It uses deductive logic to check hypothetical connections between units of meaning in the files. Each code is seen as representing a "category of analysis" in the sense of Glaser and Strauss: "A category stands for itself as a conceptual element of theory" (Glaser & Strauss, 1967, p. 36). Each coded segment is the current expression of the occurrence of that category in your data. As you work with your data and look at your conceptual categories, conjectures arise about relationships that seem to exist among the categories. Such relationships and linkages could be the beginning of theoretical observations about what is going on in your data – or in the minds of the people from whom the data come. Therefore, you can also say this module allows you to test hypotheses about relationships.

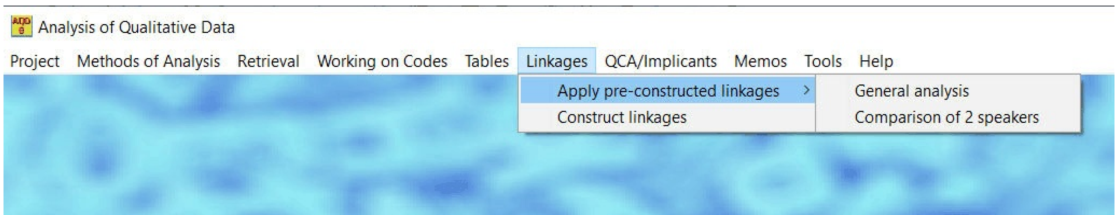
You formulate a presumed relationship between the meaning units of a file as an assertion that this very relationship is true, and then leave it to the computer to check all the entries in the code files to see if the conditions contained in the assertion are met. As a result, you will not only receive a message that the proof has succeeded or failed, but also a list of all "pieces of evidence", i.e. all codings that correspond to the assertion. This can then be used to make further comparisons.

Within AQUAD, one can check all types of code sequences in this way. Included in the software are some sequence patterns for immediate use as abstract linkage structures. When performing a linkage analysis, you enter some concrete codes of your study, i.e., when called, these abstract structures are converted with the given codes of your own study into concrete hypothesis formulations that AQUAD can check against the coding files.

To start the linkage analysis, select "Linkages" from the main menu and then "Apply linkage structures". There are two alternatives: You can use the "General Sequences" option to look for general coding sequences occurring throughout the file within the files of a project. However, this gives you the opportunity to search for the links separately for different "speakers" (generally: file segments marked by "speaker codes") if necessary. The "Compare 2 speakers" option, on the other hand, analyzes specific relationships in each case between the successive data segments of two defined "speakers" (or even questions in a questionnaire, for example), such as: "If speaker 1 says 'A', then speaker 2 claims 'B' ".

More complex correlations can also be checked with AQUAD; however, because of the variety of conceivable correlations, no abstract sequence structures are built into the program for this purpose. For this, you are given the opportunity with the "Construct links" option to enter your assumptions about complex links of codes by clicking on code names and logical conjunctions (operators "AND", "OR" and "NOT"). How to do this is explained in detail below. For pragmatic reasons, the number of codes that can be linked in this way is limited to five.

The following excerpt from the main menu shows again the possibilities AQUAD offers for searching or confirming links:

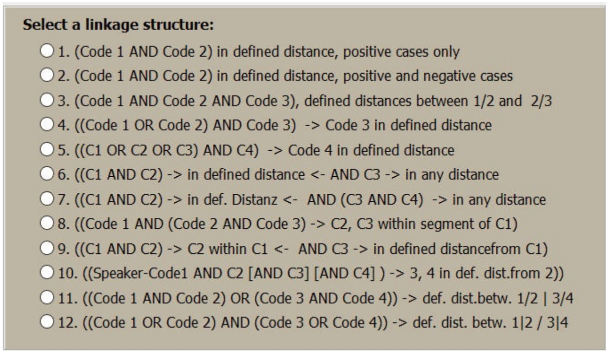


If your problem cannot be solved with the existing linkage structures or construction options, the author will be happy to create a custom algorithm.

7.3.1 Which linkage structures does AQUAD provide?

General coding sequences / examples

AQUAD provides 12 hypothetical "shortcuts" into which you can generally insert any of your codes. These linkage structures represent formulaically expressed relationships between units of meaning that you want to check:



Select a linkage structure:

- ○ 1. (Code 1 AND Code 2) in defined distance, positive cases only
- ○ 2. (Code 1 AND Code 2) in defined distance, positive and negative cases
- ○ 3. (Code 1 AND Code 2 AND Code 3), defined distances between 1/2 and 2/3
- ○ 4. ((Code 1 OR Code 2) AND Code 3) -> Code 3 in defined distance
- ○ 5. ((C1 OR C2 OR C3) AND C4) -> Code 4 in defined distance
- ○ 6. ((C1 AND C2) -> in defined distance <- AND C3 -> in any distance
- ○ 7. ((C1 AND C2) -> in def. Distanz <- AND (C3 AND C4) -> in any distance
- ○ 8. ((Code 1 AND (Code 2 AND Code 3) -> C2, C3 within segment of C1)
- ○ 9. ((C1 AND C2) -> C2 within C1 <- AND C3 -> in defined distancefrom C1)
- ○ 10. ((Speaker-Code1 AND C2 [AND C3] [AND C4]) -> 3, 4 in def. dist.from 2))
- ○ 11. ((Code 1 AND Code 2) OR (Code 3 AND Code 4)) -> def. dist.betw. 1/2 | 3/4
- ○ 12. ((Code 1 OR Code 2) AND (Code 3 OR Code 4)) -> def. dist. betw. 1|2 / 3|4

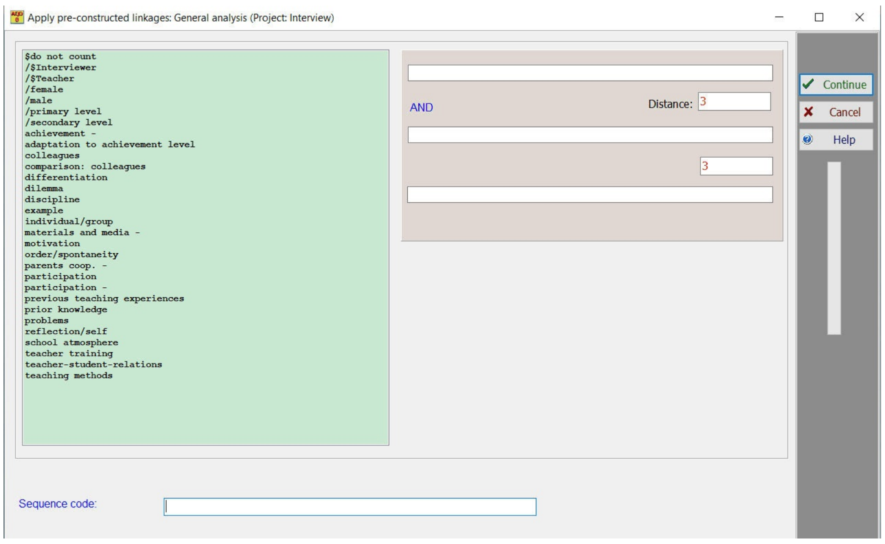
In doing so, you must not only specify the codes to be checked, but usually also a critical distance that must not be exceeded between the coded file segments.

These structures have not been didactically constructed, but have been developed while working with AQUAD in our own research, while supervising dissertations, and in doctoral courses introducing qualitative methods. Before we go into more detail about these linkages and examine some examples, here is a description of what these very abbreviated formulations mean on the screen (see figure). The first sentence repeats the hypothesized linkage, the second sentence describes what the computer tests when you activate this particular linkage hypothesis:

Hypothesis 1: Two codes occur in the same file within a certain distance of each other. In which cases is this assertion true?
Hypothesis 2: Two codes occur in the same file within a certain distance of each other. In which cases is this statement true or false?
Hypothesis 3: Three codes occur in the same file, code 2 is within a certain distance of code 1, code 3 is within another defined distance of code 1. In which cases is this statement true?
Hypothesis 4: One or both of two codes occur within a defined distance of code 3. In which cases is this assertion true?
Hypothesis 5: One, two, or all of three codes occur within a defined distance of code 4. In which cases is this assertion true?
Hypothesis 6: Three codes occur in the same file; two of them occur within a defined distance of each other, while a particular third code is located somewhere in that file - this code is simply a special condition that applies to that file as a whole (e.g., a profile code). In what cases is this assertion true?
Hypothesis 7: Four codes occur in the same file; two of them occur within a defined distance of each other, while a certain third and fourth codes (as additional general conditions) are located somewhere in this file. In which cases is this assertion true?
Hypothesis 8: Three codes occur in the same file, with code 2 and code 3 present within the file segment encoded by code 1. In which cases is this assertion true?
Hypothesis 9: Three codes occur in the same file, with code 2 occurring within the file segment encoded by code 1 and code 3 following code 1 within a certain distance. In which cases is this assertion true?
Hypoth. 10: One, two, or three codes occur in a file within the data segments of a given speaker (speaker code 1), where the distance of codes 3 and 4 is defined by code 2. In which cases is this assertion true?

Hypoth. 11: Two particular codes (1 and 2) or two other codes (3 and 4) occur in a data file within defined distance. In which cases is this statement true?
Hypoth. 12: A code 1 or an alternative code 2 occurs in a file within defined distance of a third or another alternative code (code 4). In which cases is this assertion true?

If you want to work with one of these linkage structures, choose the appropriate formulation, then enter the codes and in most cases also the maximum distances. Suppose you are interested in hypothesis 3, in which case a special input window will open:


Apply pre-constructed linkages: General analysis (Project: Interview)

```
$do not count
/$Interviewer
/$Teacher
/female
/male
/primary level
/secondary level
achievement -
adaptation to achievement level
colleagues
comparison: colleagues
differentiation
dilemma
discipline
example
individual/group
materials and media -
motivation
order/spontaneity
parents coop. -
participation
participation -
previous teaching experiences
prior knowledge
problems
reflection/self
school atmosphere
teacher training
teacher-student-relations
teaching methods
```

AND Distance: 3

3

Sequence code:

On this screen shot, both the distances between the text segments linked to code 1 and code 2 and the distances between the segments linked to code 2 and code 3 are already entered. You will see a default distance of maximum three lines in the small boxes on the right side. Just click in the boxes and enter a different distance if that makes sense for your project. Now we have to enter the code names. To do this, click on the corresponding codes in the code tab on the left. They will automatically be transferred to the next free field in the middle. If necessary, double-click to empty these fields again and transfer the contents back to the code register if you want to formulate the linkage hypothesis differently.

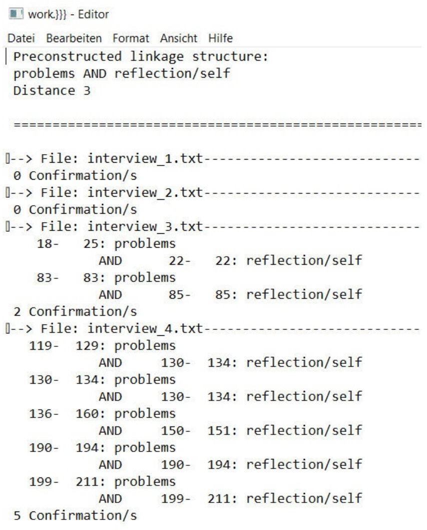
If it is necessary to enter a distance in a linkage formula, you can bypass this specification by entering a number equal to the total number of lines in your longest text - in doubt: 99999. This trick will capture any occurrences of two codes.

Also note that the order in which the codes are entered is crucial for the result, since the linkage structure is "directed", i.e. it determines the sequence of the coded segments. The first code entered must also appear first in the file. If the second appears first and then the first, in this particular case the hypothesis of a linkage of code 1 and code 2 in defined distance is rejected as false.

At the bottom of the window you will find the label "Sequence code" and behind it an input field. What is this all about? Findings of code links that are relevant to the question of your investigation can be automatically summarized in a single code, just a "sequence code", which ranges for the file segment from the beginning of the first code to the end of the last linked code. If we want to test the following hypothesis:

"*There are teachers in this sample whose experiences of teaching problems are linked to reflections on themselves.*"

we would, for example, use linkage hypothesis 1 ("Two codes occur in the same file within a certain distance of each other.") to search for links of "problem" and "reflection/self" in the "interview" sample. This search returns the following result:



```
Preconstructed linkage structure:
problems AND reflection/self
Distance 3

=====================================================

▯--> File: interview_1.txt---------------------------
 0 Confirmation/s
▯--> File: interview_2.txt---------------------------
 0 Confirmation/s
▯--> File: interview_3.txt---------------------------
    18-   25: problems
            AND      22-   22: reflection/self
    83-   83: problems
            AND      85-   85: reflection/self
 2 Confirmation/s
▯--> File: interview_4.txt---------------------------
   119-  129: problems
            AND     130-  134: reflection/self
   130-  134: problems
            AND     130-  134: reflection/self
   136-  160: problems
            AND     150-  151: reflection/self
   190-  194: problems
            AND     190-  194: reflection/self
   199-  211: problems
            AND     199-  211: reflection/self
 5 Confirmation/s
```

The list of references shows that
in interview 1 and in interview 2 the presumed linkage never occurs,
in interview 2 three times and
five times in interview 4.

In the case that the examined linkage yields significant findings for our investigation, we could enter "Reflection in case of problems" as sequence code in the input field below. After clicking "Next," a security question follows: "Do you want to insert the found links as sequence codes into the code files?" If you answer "YES" here, the code files will be expanded accordingly, otherwise you will only receive the list of results.

For each search for links, AQUAD offers the option of marking the finds with a sequence code, i.e. also for marking the findings with self-constructed link hypotheses. In the corresponding window there is an input mask labeled "Sequence code" for this purpose.

Comparison of coding between two speakers / examples

If the coding distinguishes between different speakers (e.g. in a group discussion) or the questions of a questionnaire as "quasi-speakers", the comparison of two speakers each can be interesting. In AQUAD, 13 hypothetical "links" are provided to test connections between the utterances of two speakers. You put concrete codes into the abstract specifications. Below is a list of all the hypotheses currently available in AQUAD.

Code A, code B, etc. represent the coded utterances of the speakers, where code A, B, etc., can possibly be identical for speaker 1 and speaker 2 (see hyp. 1):

1. S1: Code A -> S2: Code B.
   "Speaker 1 utters A, whereupon speaker 2 utters B" (A and B may be identical, i.e., speaker 2 repeats an utterance of speaker 1).
2. S1: Code A -> S2: Code B and Code C
   "Speaker 1 utters A, whereupon speaker 2 utters B and C".
3. S1: Code A -> S2: Code B or Code C
   "Speaker 1 expresses A, whereupon speaker 2 expresses B or C".
4. S1: Code A and Code B -> S2: Code C
   "Speaker 1 utters A and B, whereupon speaker 2 utters C."
5. S1: Code A and Code B -> S2: Code C and Code D.
   "Speaker 1 utters A and B, whereupon speaker 2 utters C and D."
6. S1: Code A and Code B -> S2: Code C or Code D.
   "Speaker 1 utters A and B, whereupon speaker 2 utters C or D."
7. S1: Code A or Code B -> S2: Code C.
   "Speaker 1 utters A or B, whereupon speaker 2 utters C."
8. S1: Code A or Code B -> S2: Code C and Code D.

"Speaker 1 utters A or B, whereupon speaker 2 utters C and D."

9. S1: Code A or Code B -> S2: Code C or Code D.

   "Speaker 1 utters A or B, whereupon speaker 2 utters C or D."

10. S1: Code A and (Code B or Code C) -> S2: Code D.

    "Speaker 1 utters A and (B or C), whereupon speaker 2 utters D."

11. S1: Code A and (Code B or Code C) -> S2: Code D and Code E.

    "Speaker 1 utters A and (B or C), whereupon speaker 2 utters D and E."

12. S1: code A and (code B or code C) -> S2: code D or code E.

    "Speaker 1 utters A and (B or C), whereupon speaker 2 utters D or E."

13. S1: code A and (code B or code C) -> S2: code D and (code E or code F).

    "Speaker 1 utters A and (B or C), whereupon speaker 2 utters D and (E or F)."

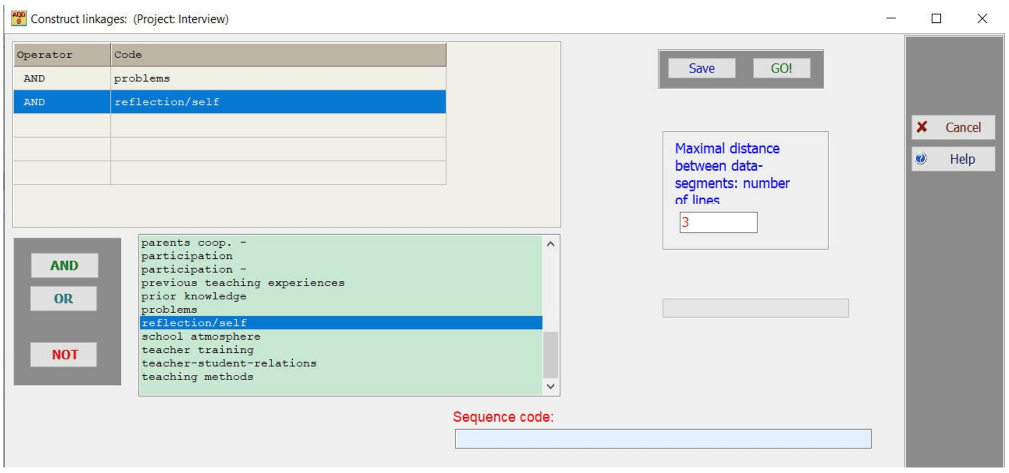
AQUAD communicates the results by indicating the locations in the files where appropriately coded segments can be found. Thus, the files themselves are not printed. At the hypothesis generation stage, we are no longer working with the original data; we are operating at a more abstract level.

In addition to the pre-formulated hypotheses, there is the possibility to create and test your own linkage hypotheses. This is presented in the next section.

7.3.2 How to construct linkages

As you have seen above, linkages are possible with fixed comparison elements. Thus, "linkage formulas" are available, so to speak, in which one inserts one's own codes as "variables". All these variables are linked with the logical "AND" or "OR". In general this is sufficient. However, if one comes to hypotheses, according to which, for example, an issue is addressed by a speaker, one must construct one's own linking hypotheses and occasionally also use "NOT" as a logical operator. In the following, Leo Gürtler gives examples and suggestions from his experiences with qualitative analyses in the context of his dissertation on how to construct one's own linking hypotheses in a meaningful way. The possibilities of the program option "construct linkages" and "sequence codes" (insert into code files) are combined.

Before we go into these complex linkages, we will look at the simple example of linking the codes "Problems" and "Reflection/Self" (example see above) to see how one can construct the hypothesis of their linkage itself and what comes out when checking it:

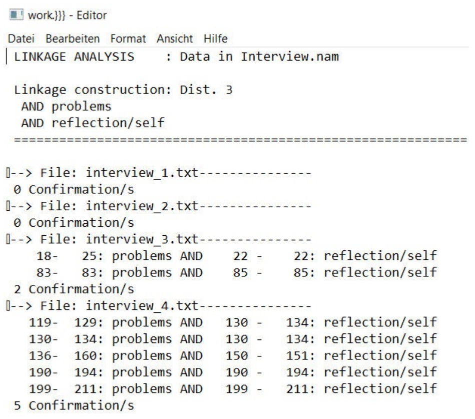


Operator	Code
AND	problems
AND	reflection/self

```
parents coop. -
participation
participation -
previous teaching experiences
prior knowledge
problems
reflection/self
school atmosphere
teacher training
teacher-student-relations
teaching methods
```

Maximal distance between data-segments: number of lines: 3

One could save this construction for later re-use by clicking on "Save". The button "Go" starts the analysis and shows the following result (cf. above result of the given hypothesis 2):



```
LINKAGE ANALYSIS    : Data in Interview.nam

 Linkage construction: Dist. 3
  AND problems
  AND reflection/self
 ============================================================

--> File: interview_1.txt---------------
 0 Confirmation/s
--> File: interview_2.txt---------------
 0 Confirmation/s
--> File: interview_3.txt---------------
    18-    25: problems AND    22 -    22: reflection/self
    83-    83: problems AND    85 -    85: reflection/self
 2 Confirmation/s
--> File: interview_4.txt---------------
   119-   129: problems AND   130 -   134: reflection/self
   130-   134: problems AND   130 -   134: reflection/self
   136-   160: problems AND   150 -   151: reflection/self
   190-   194: problems AND   190 -   194: reflection/self
   199-   211: problems AND   199 -   211: reflection/self
 5 Confirmation/s
```

Now for the use of own hypothesis constructions in a complex context: In his investigation of "Subjective Theories of Humor" (coded answers of students in a questionnaire; Gürtler, 2004) the hypothesis came up that there is a connection between answering two questions:

Question 7 - "Do you think there is enough humor in the classroom?" and
Question 8 - "If you could change teaching yourself, what would you do to make it more humorous?"

It was now interesting to learn whether students who responded with dissatisfaction to question 7, i.e., who are not satisfied with the quality or quantity of humor in the classroom, might still have no suggestions for improvement to offer to question 8. This would give an indication that dissatisfaction can occur with a tendency towards passivity in the response patterns and that the responses should be distinguished from those students who also express dissatisfaction in question 7, but who do offer suggestions for improvement in question 8 (e.g., changes in the classroom climate, the teaching style of the teachers, or the methods used). In order to qualitatively separate and identify these groups, which could be sub-elements of a type, the following sequence codes were established, given a unique name, and inserted into the code files:

```
SeqCode 1: "F7_noSatisfaction"
    AND Speaker Code:  /$Question7
    AND Code:          StatusQuo: not enough humor
    NOT Code:          StatusQuo: enough humor

SeqCode 2: "F7_satisfaction"
    AND Speaker Code:  /$Question7
    AND Code:          StatusQuo: enough humor
    NOT Code:          StatusQuo: not enough humor

SeqCode 3: "F8_noAnswer"
    AND Speaker Code:  /$Question8
    AND Code:          Missing Data
    OR Code:           don't know/ don't care

SeqCode 4: "F8_suggestions"
    AND Speaker Code:  /$Question8
    AND Code:          Change structures (teaching/ methods)
    OR Code:           Foster climate/ relationships
    OR Code:           Change at the institutional level
```

As you can see, each of these sequence codes is clearly linked to one of the questions (there were nine in total) in the questionnaire. All positive results of the subsequent search returned hits, which are thus also exclusively related to the answer to this question. In a second step, additional sequence codes were established. These looked like this:

```
SeqCode 5: "Vgl_F7<->F8_inconsistent1"
    AND Code:      F7_noSatisfaction
    AND code:      F8_noanswer

SeqCode 6: "Vgl_F7<->F8_inconsistent2"
    AND code:      F7_noSatisfaction
    NOT Code:      F8_suggestions

SeqCode 7: "Vgl_F7<->F8_inconsistent3"
    AND Code:      F7_satisfaction
    AND Code:      F8_suggestions

SeqCode 8: "Vgl_F7<->F8_consistent1"
    AND Code:      F7_noSatisfaction
    AND code:      F8_suggestions
```

This coding scheme allows to distinguish from each other those students who

- are dissatisfied with the school situation, but have no answer at all (= "missing data") to the question about possibilities for improvement. This is inconsistent and could give clues about the experienced potential for action in the context of school or the willingness or ability to express or contribute own designs and suggestions.
- are dissatisfied, but do not present any suggestions. This is also inconsistent like the previous example, but here apparently other expressions and thus an answer at all are given, which is qualitatively different from SeqCode 5.
- are satisfied with the school situation, which could mean they do not want to change anything, but still present suggestions for improvement (here the question is: are these students not really satisfied or are they rather creative and engaged, which would have to be investigated in a separate analysis).
- are dissatisfied, but have suggestions for improvement to offer. This is internally consistent with something being perceived as incongruent, but also with at least cognitive efforts being made to address that incongruence.

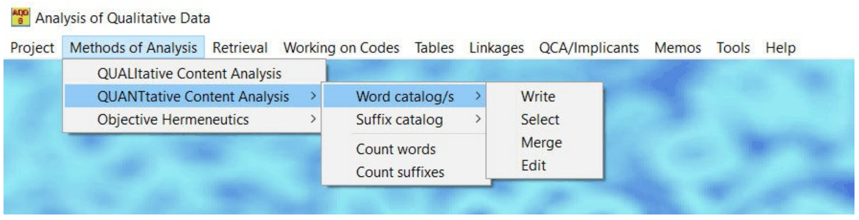
Combining sequence encoding with pre-existing sequence encoding is an excellent way to unambiguously answer more complicated questions. The example should additionally emphasize the intelligent planned use of speaker codes. Speaker codes have the useful property of neatly separating specific parts of the text, which is essential for comparing qualitative hypotheses in typing. Besides actual speakers, you can use speaker codes for the questions of a questionnaire (see example) or different areas like emotions, cognitions, actions. As a starting point for a type formation this procedure is perfectly suitable, as well as a basis for a later implicant analysis (cf. Ragin, 1987). For especially in implicant analysis (see below; QCA: Qualitative Comparative Analysis) you benefit from not only relating individual content codes to each other, but from using propositions (e.g., subject - predicate - object, i.e., part/actions) that have been operationalized by sequence coding.

Chapter 8: Quantitative Content Analysis

8.1 Counting words

We cannot address here the controversy regarding the use of quantitative approaches in the analysis of qualitative data. However, there is no doubt that for some research questions it can be quite useful to supplement qualitative with quantitative analyses. If one has identified key words as indicators of particular statements in the texts, then by counting these words and comparing their frequencies in the texts, one can at least obtain initial indications of foci of meaning. Other text analysts would go further and use the results of a word-level frequency analysis as the initial data of relevant statistical procedures, such as cluster analyses. A detailed discussion of word frequencies can already be found in Vorderer and Groeben (1987).

Access to the counting functions in texts can be found in the main menu of AQUAD in the function group "Methods of Analysis" -> "QUANtitative Content Analysis". The following figure gives an overview of how lists of critical words can be created. The "Count words" can be selected in the preceding submenu of the subfunctions.



Before you can start counting words, you must enter the words you want to count or enter the name of a corresponding word list that is already available. The program does not search for words as "text parts" here, but splits the whole text into single words. These are all converted to lower case. This means that the computer counts only whole words. If you want to count only certain words with word lists, the program considers only words that are contained in the list. For example, if you have entered the word "friend", the computer will not count "friendship" or "circle of friends", and so on.

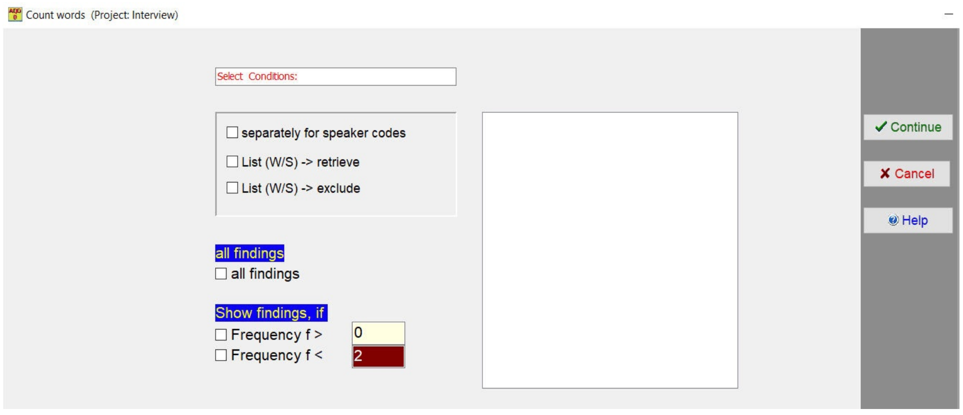
As a second option, you can have the program list all the words in a given text (or in several texts). From the alphabetically sorted list of single words on the screen one selects then those by clicking, which are to be arranged for limited frequency analyses in a word list. It is recommended to first create a word list for critical texts from the text by "Select".

If you have stored words of several texts in different lists in this way, you can "merge" exactly these (or some) word lists with the third subfunction.

As a last possibility you can "edit" word lists. To do this, one loads an already available list on the screen and then deletes those words that one does not want to include in a subsequent frequency analysis. Of course, you can also add more words.

For later re-use, all lists are saved under a freely selectable name. AQUAD adds the extension ".cwo" (catalog of words) to the file name for automatic recognition. Word lists in the form of word fields or "lexicons of meaning" help to quickly gain an overview of relevant content areas in the texts of a project.

When we finally click on "Count words", the first window that appears is the one below, in which the conditions of the analysis and the output of results are defined. In detail, AQUAD offers the following options:


Count words (Project: Interview)

Select Conditions:

☐ separately for speaker codes
☐ List (W/S) -> retrieve
☐ List (W/S) -> exclude

all findings
☐ all findings

Show findings, if
☐ Frequency f > 0
☐ Frequency f < 2

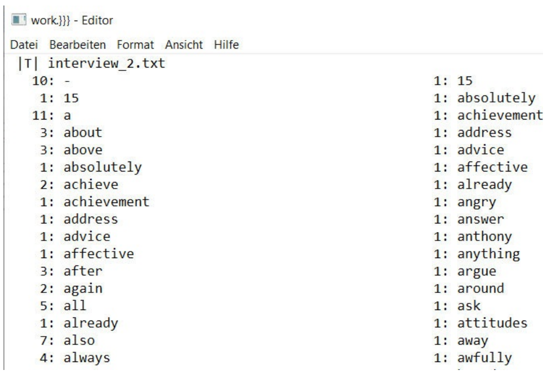
If the texts are subdivided with speaker codes, the counting of words can of course be performed separately by speaker. (We would like to remind you here that "speaker" can be interpreted creatively; for example, when analyzing open-ended answers in a questionnaire, one could mark the individual questions with speaker codes and then evaluate them separately).

However, individual speakers, e.g., the moderator of a group discussion, can be excluded from quantitative analysis by the control code "$do not count" already during coding.

If word lists have been created, these files will be displayed on the right side of the still empty window for selection, if one ticks one of the boxes "List (W/S) -> search" or "List (W/S) -> exclude". With this you decide whether only the words in the selected list are counted or exactly these words are excluded from counting. Without selecting a list, all words of the texts are counted.

For displaying/outputting the results we see three checkboxes in the bottom left corner: We can display all results - then the next options are omitted. Or we choose a lower limit (f>0) and/or an upper limit (f<2) for the output of finds, where the "0" and the "2" can be changed as needed.

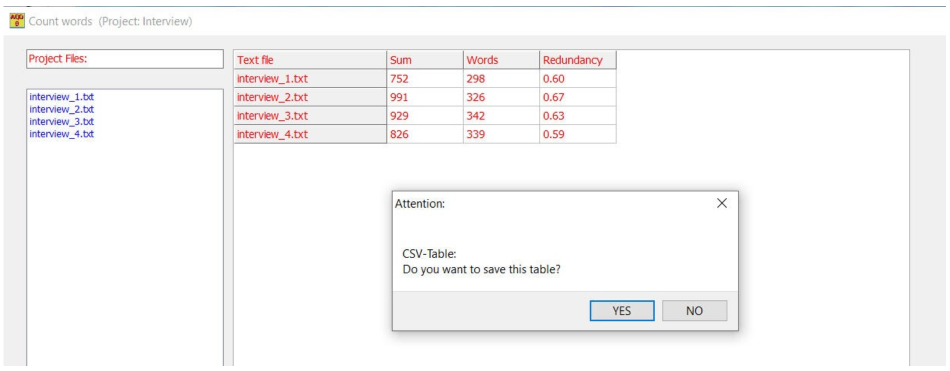
Counting is always done in all files of a project. here in the four interview transcriptions of the project "interview". If you want to consider only certain files, you can create a project definition in which only these files are listed.



```
|T| interview_2.txt
  10: -                                               1: 15
   1: 15                                              1: absolutely
  11: a                                               1: achievement
   3: about                                           1: address
   3: above                                           1: advice
   1: absolutely                                      1: affective
   2: achieve                                         1: already
   1: achievement                                     1: angry
   1: address                                         1: answer
   1: advice                                          1: anthony
   1: affective                                       1: anything
   3: after                                           1: argue
   2: again                                           1: around
   5: all                                             1: ask
   1: already                                         1: attitudes
   7: also                                            1: away
   4: always                                          1: awfully
```

What does the result look like if we let all words count? The first result we get is an alphabetically ordered list of all words on the left and again all wordson the right side, but ordered by frequencies. In front of each word its frequency in the respective text is entered. The screenshot shows the beginning of the results of counting the words in "interview_2.txt":

This is followed by a tabular overview of the total number of words in each text, the number of different words per text, and the redundancy of the text, i.e., the ratio of the total number of words to the number of different words:



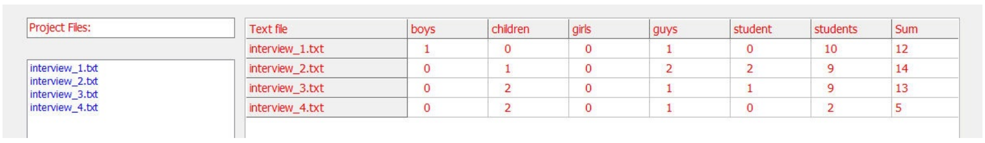
Text file	Sum	Words	Redundancy
interview_1.txt	752	298	0.60
interview_2.txt	991	326	0.67
interview_3.txt	929	342	0.63
interview_4.txt	826	339	0.59

Attention:

CSV-Table:
Do you want to save this table?

A note: If one wants to perform statistical calculations with the frequencies, one should correct the frequencies of the critical words as a function of the total number of words in a text. It could make a big difference for the inferences whether, for example, the word "I" occurs twelve times in a text that consists of 700 words in total or whether it consists of 1400 words.

As an example of the need to correct the frequency data, let us look at the frequencies of the words contained in the word list "Students.cwo" when counting into all four interview transcriptions. In this word list, the words teachers used to refer to their students were compiled from all four interviews:



Project Files:

interview_1.txt
interview_2.txt
interview_3.txt
interview_4.txt

Text file	boys	children	girls	guys	student	students	Sum
interview_1.txt	1	0	0	1	0	10	12
interview_2.txt	0	1	0	2	2	9	14
interview_3.txt	0	2	0	1	1	9	13
interview_4.txt	0	2	0	1	0	2	5

References to "students" (boys,children girls, guys,student, students) occurred 12 times in Interview 1 (752 words), 14 times in Interview 2 (991 words), 13 times in Interview 3 (928 words), and only 5 times in Interview 4 (826 words). After taking into account the interview length and correcting accordingly, the difference between the first three interviews and the fourth is confirmed as far as the focus on students is concerned. However, what this may mean now is a question that needs to be answered qualitatively....

At the end we should not forget to save the frequency list. To do this, we click on the "File" module at the top of the results window and then on "Save as". This will open the usual save dialog. Select as path/directory for saving the subdirectory ...\RES in the directory of the AQUAD files (the path for results). We recommend to use as file name for saving the frequency lists the name of the word list - here "students" - and as extension the letters ".frq" (for "frequency"). This is the easiest way to find the results later.

8.2 Counting suffixes

For special questions, e.g., the determination of cognitive styles from verbal productions of persons (cf. Günther, 1987), it seems appropriate to use a search and count algorithm which captures only word endings or suffixes (such as "-acy", "-ance", "-dom", "-ism", "-ment", "-ness", etc.). Again, you need to create a directory in which you enter all the suffixes you want to count. Then select "Count suffixes" and you will get the frequencies. When entering the suffixes, you should also remember the plural forms.

8.3 How to exclude parts of text from word analysis

Above it was mentioned how parts of a text can be excluded from the analysis with a control code. In the meantime it should have become clear why this possibility is often urgently necessary. In interview analyses, for example, all verbal utterances of the interviewer would be displayed or counted.

Here we show a small textual example of a teacher-student interaction in which we wanted to analyze how often the students refer to the critical problem objects, the spheres. We therefore masked out the teacher utterances with the control code "$do not count". In this way, the teacher utterances are excluded from the analysis, but the student utterances are not:

54 ...
55 TEACHER How many more bullets can we put on it *$do not count 55 - 56*
56 so that it makes sense?
57 STUDENT Huh?
58 TEACHER How many more bullets can we compare, *$do not count 58 - 59*

```
59 how many bullets on each side? ...
60 STUDENT Two -- three -- and four, or one bullet ...
61 TEACHER And which is the best solution?          $do not count 61 - 61
62 STUDENT Four
63 TEACHER Why?                                     $do not count 63 - 63
64 ...
```

Chapter 9: Working with Memos

9.1 What are memos good for?

All ideas, concerns, possible contradictions, etc. that come up during work, as well as other project notes, cross-references, literature references that you do not want to forget, can be recorded directly in AQUAD and specifically searched for again. We strongly recommend that you make extensive use of the memo function in AQUAD. Because of the importance of memos in processes of continuous comparison of texts and theory-constructing analysis, special search algorithms have also been built into AQUAD to help researchers later reconnect the memos to the texts, segments, codings, etc. that gave rise to the noted idea.

Loosely based on Miles and Huberman (1991, p. 69), we emphasize that collecting and analyzing texts is so exciting and coding often so exhausting that it is all too easy to get lost in the abundance of interesting details, as there are "...the trenchant phrasing, the intriguing personality of a key informant, the telling cartoon on the hallway bulletin board, the informal conversations after an important meeting. They forget to think, to seek the deeper and more general meaning of what is happening, to explain it in a conceptually coherent way. ...". It is therefore almost indispensable to write down everything that goes through one's mind about central categories and their possible connections during the interviews, the observations, the conversations, etc., and then during the coding.

Three decades ago, a dispute arose between Glaser (see Glaser 1992, p. 108 ff.) and Strauss (see Strauss & Corbin 1990, p. 197 ff.) about the importance of memos in the process of qualitative analysis. They agree that memos should accompany the research process from the first beginnings to the final report, and that one should not refrain from immediately jotting down ideas at any stage. Memos provide important supports for theoretical reflection. Abandonment of memos is evident in the final product of the project: "a theory whose concepts lack density and/or are only loosely related" (Strauss & Corbin, 1990, p. 199).

However, Strauss and Corbin then set out to canonize the spontaneous, creative, and perhaps at first wildly speculative memos; they begin to distinguish between memos at large, coding memos, theoretical memos, operational memos, diagrams, and logical diagrams, and to develop specific procedures for each. At this point, we recommend following Glaser's (1992, p. 109) objection that one cannot develop in advance and in general a system of whether and why one needs to pay attention to very specific features in any theory one is constructing – until one comes across those very features. On the other hand, such feature systems can have useful heuristic functions (cf. Huber 1992, p. 145 ff.), but by no means guarantee or force the construction of a particular theory building as an exhaustive description or blueprint. Or, to put it differently: according to the approach of generating a "grounded theory" it would be ideal if "... the analyst starts with no structural assumptions at all and lets the concepts structure themselves in the process of their emergence" (Glaser 1992, p. 110); but now, if nothing wants to emerge, some suggestions under which aspects one could still look at a text can be quite helpful.

As a conclusion we hold: write down all your ideas in all phases of the qualitative analysis as memos. In doing so, do not try to follow a rigid notational system developed independently of your texts.

9.2 How to write memos in AQUAD

AQUAD offers two possible ways to access its memo functions, depending on the modules you are currently working with:

1. There is a "Memos" option in the main menu, and
2. you will find a "Memo" column on the left side of the coding table (in the "Content Analysis" options in the main menu).

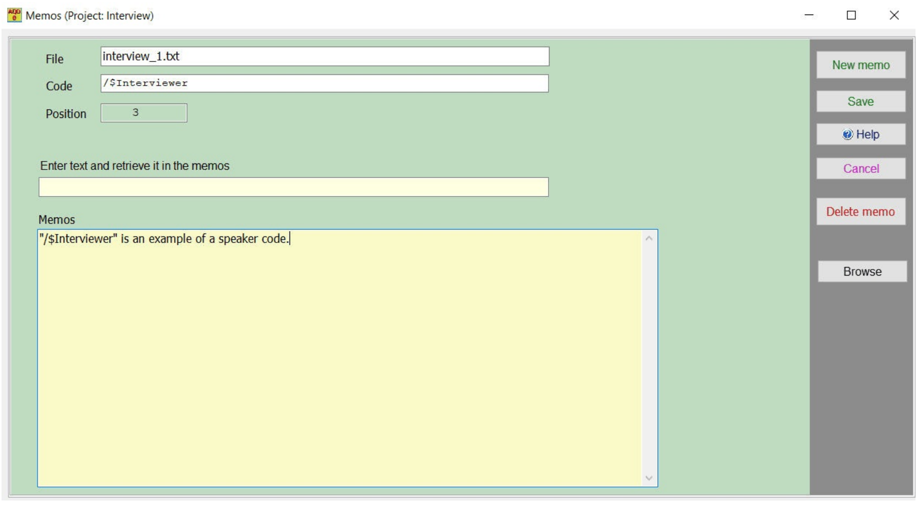
Especially when coding, you will often want to jot down an idea. Therefore, let's deal with this first.

9.2.1 Writing memos while coding

While encoding text, images, audio or video recordings, you create your own code files for your data files. At the same time, you can activate the memo function whenever necessary by clicking in the memo column ("M") on the left side of the coding window.

In the lines of the coding table where no memo has been attached yet, the "Memo" column is empty, in the other lines an "X" is entered in this column. Clicking on this character opens the memo window with the first of possibly many memos associated with this row.

Clicking on an empty cell in the "Memo" column opens an empty window in which a memo can now be written:



File: interview_1.txt
Code: /$Interviewer
Position: 3
Enter text and retrieve it in the memos
Memos
"/$Interviewer" is an example of a speaker code.

In this box, some entries are already made automatically: (1) the name of the file to which this memo will be linked, (2) the code linked to these rows, and (3) the number of the row in which a cell of the "Memo" column was clicked. Users have requested to be able to change the first two entries (double click on the text windows). The change of these criteria is facilitated by the fact that with double click into the fields for file name and code in each case an additional window (file directory of the project; code register) opens, from which you select the suitable contents by clicking. Normally, it would not occur to you to want to change the name of the file to which your memo refers, but if it should be necessary to close these windows again, just double-click. According to the logic of "writing memos while coding", the position/line number (as determinants of a file segment) should be changed while coding only by clicking in the "Memo" column in another line.

In the input line under the yellow heading "Enter text and search in the memos" you can enter a text part or just a keyword, if you later want to find those memos containing this text part or keyword when "browsing" through the memos.

What about the memo text? You have a yellowish input window in front of you, but the real length of a memo is practically unlimited. For example, you can copy a full text file into the memo box - if that makes sense. In other words, there are no limitations when writing memos. Copying text passages and copying text parts from other memos is done by right-clicking anywhere in the text. The entire text then appears in the editor, where the desired passage can be marked and copied using the editing functions. In the memo window you then right-click in the memo box and select the "Paste" function. For sure you will see here also the possibility to print your memos.

!! *Don't forget to save the memo with "Save" when you are done with your entry* !!

9.2.2 Writing memos from the main menu

The explanations in section 9.2.1 are valid for all phases of the analysis. An important difference, however, is that from the main menu no entry (e.g. the file name) is already given in any of the input windows of a memo, because in this case AQUAD does not know to which file or which data segment the memo is to be assigned.

You can select the missing entries from lists that appear when you click in the corresponding input column (e.g. "Code"). However, you must enter directly in which line the text segment begins to which the new memo is assigned.

9.3 How do I find my memos again?

This is a question we have dealt with thoroughly in the programming of AQUAD. Because especially with the fleeting ideas that are advantageously recorded in memos, it often happens later that one remembers a tremendously important idea in this or that file, in connection with a certain segment, etc. - but simply no longer knows exactly what the significance of the thought was.

If one would remember details, one would have found the memo again immediately. Now, of course, you can read through all the memos one by one - this is also possible in AQUAD. However, support for a defined query is very helpful in this situation. If you run a query with the goal of retrieving your memos, all visible entries in your memo window can serve as search criteria.

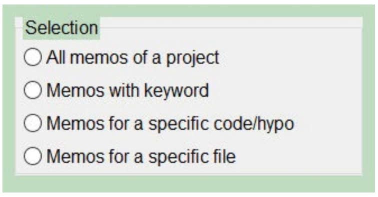
9.3.1 Searching for memos while coding

While you are busy encoding a particular file, a considerable part of your work will consist of "constant comparison" (Glaser & Strauss, 1967). As you ascribe a particular meaning to a file segment and code it, you will often recall another segment that seems very similar or very different. How did you code that segment - and why just that way? So you will go back to that segment and take another close look at the coding(s) associated with it. Or you may recall that you have used a particular code before in a similar situation. Memos with your thoughts when coding that segment or using a specific code would be very helpful now! Find the data segment or location where you used the critical code, click on the appropriate cell of the coding table cell in the "M"(emo) column, and read the memo related to that data segment and/or code.

If you right-click anywhere in the memo's text field, the contents will be transferred to the editor, from where they can be saved externally, printed, copied, and pasted elsewhere.

Scrolling through memos

The "Browse" option needs a bit more explanation. If you are searching memos specifically for certain criteria while coding, then use this button! On the second click, the list of possible search criteria appears, but in three cases here, this criterion must be entered into the memo form before you press the "Browse" button.



If you want to see all memos of the project, just click the first option - but then you will get a listing of all memos, sorted by files and line numbers.

For the second option, before scrolling to the line under the yellow heading "Enter text and search in memos", you need to enter a keyword or phrase that will serve as a criterion for displaying those memos (from the entire project) that contain the criterion. Even when writing the memos, you can use specific headings (e.g. "Code definition") to ensure that you find specific entries all together.

For the third option, before scrolling, select (double-click on the code line) the desired code for which you have written memos that you now want to retrieve.

For the fourth option, before browsing, select (double-click on the file line) the name of the file for which you have written memos that you now want to retrieve.

The search results of browsing will be displayed in the editor. If you want to save or print the found memos externally, you can find the corresponding commands in the toolbar at the top of the editor's output window under "File". You can paste selected parts or all selected memos into other memos or open texts in the editor under "Edit" (there "Copy" and "Paste").

9.3.2 Searching for memos from the main menu

If you activate the "Memos" option from the main menu, you will have immediate access to all memos associated with your current project. The same rules apply as when searching for memos while coding (see above).

Chapter 10: Implicant Analysis (Qualitative Comparative Analysis)

In addition to the possibility of testing hypotheses about relationships between units of meaning, the procedures for logical minimization of conditional configurations represent another special feature of AQUAD. The basic ideas of the qualitative meta-analyses or single case comparisons systematized with it are based on the work of Charles Ragin (Ragin, 1987).

To summarize, it should only be repeated here that with the component "QCA/implicants" the case-specific meaning configurations are transformed into truth values of a binary logic (feature applies/does not apply) and linked with each other according to the rules of Boolean algebra. One of these features serves as a criterion for the comparison. The comparison of all cases in which this criterion is "true" (or "false") leads to the reduction of configurations/to the main implicants of this criterion. Since there is often more than one such implication pattern in an investigation, the logical redundancy can be further reduced by analyzing the essential implicants and by delineating secondary implicants, which is sometimes necessary in complex conditional configurations. We recommend to work with the initial implicants, i.e. not to reduce the configurations further, because they all are represented by cases in your data – at least, if you want to develop an understanding or even a strategy of intervention. The result generally represents significant configurations of conditions. In the following the steps and possibilities are described when using the module "QCA/Implicants".

10.1 Writing a data table

AQUAD allows logical minimizations of the condition configurations of multiple studies or cases to be performed directly, using quantitative and/or qualitative data. One can combine these data into a table, i.e., write, edit existing tables, and print data tables. Qualitative data, e.g., "Significance A strongly expressed" (i.e., "true"), can be entered directly into the truth value table. However, it might be advantageous to represent qualitative data in the form of natural numbers and then enter, for example, the numerical value "9" for "true".

Correspondingly, one would write the interpretation "meaning A weak / not pronounced at all" as "1" or as "false" in the data table. AQUAD provides for both possibilities.

Here we demonstrate how to enter both quantitative data and qualitative information expressed in numerical values. To illustrate the steps, we introduce a fictitious example of meta-analysis of the association of school performance with class size. We assume 12 relevant studies were found in the literature. Let us further assume that four conditions had been recorded in all 12 studies: A = class size, B = aptitude, C = duration of the experiment, and D = school performance. Our data should include both qualitative values (A: class judged to be large or small) and quantitative measures (B: class aptitude level, C: experimental duration, D: achievement). The data from each case will later result in seven *different* configurations of conditions. We do not need to worry about distinguishing these configuration patterns. We only enter the results of all single cases without sorting them ourselves and leave the reduction to different condition configurations to the computer. Here are the different possible conditions (data classes) in this study:

A: size of the class - "small" or "large"- according to the information in the reports about the study.
9 = small class ("true" in the sense of the studies' hypothesis) / 1 = large class ("false").
B: results (standard values) of an ability test (arith. mean of individual values in a class)
C: time (duration in weeks)
D: Results (standard values) of an achievement test (arithmetic mean)

The first steps need no explanation: You select the menu option "Implicants" and there the submenu "Write data table".

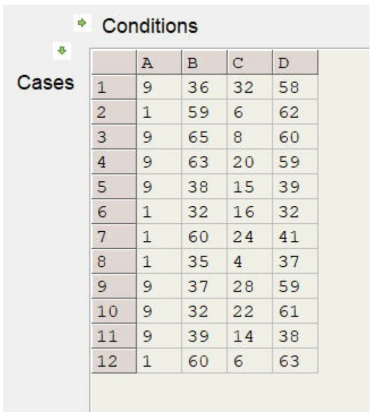
The two arrows in the upper left corner of the figure draw your attention directly to the two parameters of your data table that you need to set first:

- Number of conditions: In the current version, the program can handle between two and twelve conditions (including the criteria condition). The parameter "Number of conditions" defines the number of columns in the table. The program could accommodate more conditions, but expanding it makes little sense given the complexity of the expected results and the difficulty of interpreting them. In our case, performance (condition D) serves as the criterion for the comparisons - the configurations of the remaining characteristics A (size of the class), B (average ability), and C (duration of observation) are tested as conditions of criterion D.

  In our case, we write the number "4" in the first input field or use the upper arrow to find "4".

- Number of cases: At this point you need to define how many study results you want to enter in the table (in this example: the number of studies; usually we enter the number of text files or interview partners). In other words, you define the number of table rows. The only limitation here is that you need at least three cases for a meaningful comparison.

  In this example, we enter the number 12 because we only want to compare 12 studies.



	A	B	C	D
1	9	36	32	58
2	1	59	6	62
3	9	65	8	60
4	9	63	20	59
5	9	38	15	39
6	1	32	16	32
7	1	60	24	41
8	1	35	4	37
9	9	37	28	59
10	9	32	22	61
11	9	39	14	38
12	1	60	6	63

The table automatically expands according to the number of columns and rows you define, so you can immediately start filling the cells with your data. Remember, in our fictitious example we use both qualitative data (condition A) and quantitative data (conditions B, C and D). Click on the "OK" button to finish your entries.

Next, an additional window opens where you enter the name under which the table should be saved, for example "Class-size". When you are done, press "OK" in the small window and the table is saved (see confirmation).

If you work with these example data, you can find this file after saving in two formats in the subdirectory "..\cod" as "Class-size_DT.txt" in text format and as in the subdirectory "..\res" as "Class-size_DT.csv" in CSV format ("_DT" stands for Data Table). Both have already been copied during installation to the directory you have specified for the encoding files. If you want to experiment further with "QCA/Implicants", you will find these files in the corresponding selection box on the screen (see below).

A brief comment on the meaning of the digits in the table: We assume that we have no more precise information about the participating classes and only know whether they are large or small (condition A). For the transformation into truth values, we enter large numbers for features that are considered "true" and small numbers for features that are considered "false". Since we suspect a relationship between small classes and high performance, we assign the number 9 to small classes (arbitrary decision) and the number 1 to large classes.

Metric values are available for other conditions (B: ability values; C: duration of the experiment; D: performance values). These values are entered as they are. At a later point, the program transforms them into truth values.

It is recommended to transform all conditions into numbers, even if they are exclusively qualitative conditions originally expressed by the means of language (such as "much", "little", "often", etc.). Then enter these numbers into the table (as described above). This procedure has proven to be more convenient than reducing the results to the truth values "true" or "false" in the form of upper and lower case letters and typing them in (see below).

In the example above, condition A is handled in this way. In other cases we could, for example, write the number 9 in the corresponding column if we had to judge a statement like "Condition ... is given" or "Person ... can stand his ground". In the opposite case, i.e. if the "condition ... is not given" or if the "person ... cannot assert", we would enter the number 1. You can choose almost any value you want. It's just important that when the distribution is converted, default values are created that are above or below the cut-off, which are then replaced with uppercase ("true") and lowercase ("false") letters as appropriate.

By the way, there is a shortcut on how to get from the frequency table to the logical minimization table. You can invoke the "Count codes" option in the "Retrieval" submenu to produce a frequency list of those codes that represent relevant conditions for comparing the cases in your study. You can save this list in the editor. Additionally, the list will be converted into a data table and saved in CSV format. The result of the frequency count is saved under a file name (..._DT.csv) that you enter for it. Just select this file when you activate the option "Convert data table to truth table" in the submenu of "QCA/Implicants".

10.2 Converting data into truth values

On the way from a data table to the table of truth values, with which logical minimization is then performed, the conversion of the initial data into the reduced information "true" or "false" is the crucial step. To make the various results of the minimization process easier to interpret, AQUAD does not work with binary numbers and the corresponding algorithms. In an expression like "1001" or in the result "01*", the meaning of the individual truth values (or eliminated conditions) would have to be determined again and again on the basis of the position in the expression. AQUAD uses the letter abbreviations of the conditions instead, with upper case letters representing "true" and lower case letters representing "false". In our example, a case with the condition configuration would be

- A: small class ("true")
- B: with relatively high aptitude level ("true"),
- c: in which, after only a few weeks of experimentation ("false")
- d: low performance ("false") was observed,

not represented by the expression "1100" in the truth value table, but more easily understood by "ABcd".

Now, how do we get from the data table to truth value expressions like in our example? We either select the option "Convert data table to truth table" in the submenu of "QCA/Implicants" or we transform the original data into truth values ourselves and select "Write truth value table".

In the latter case, the procedure is the same as writing data tables, but instead of digits, you have to enter the expressions of the respective condition (letter in the column header) as uppercase ("true") or lowercase ("false") letters in the table cells.

When you transform data tables, AQUAD performs the following transformation strategy: For each condition, values in the table are first standardized (across all cases), i.e., transformed to default values with M=100, SD=10 . Then the data in this internal intermediate table are transformed into an upper or lower case letter according to a certain criterion for cut-off. The value for the cut-off is set to

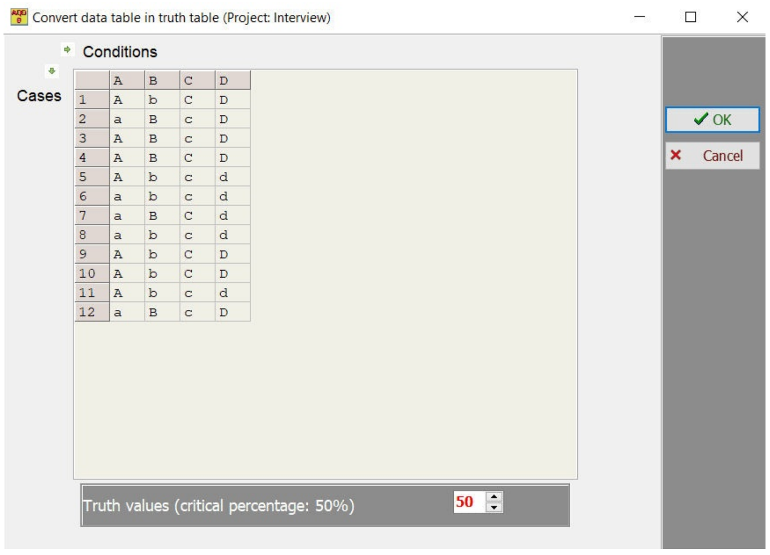
cut-off = 50

which means that the values of a condition that are below 50 – i.e. the "lower half", are transformed into the truth value "false" and symbolized by lower case letters. Correspondingly, the values above 50 are reduced to the truth value "true" and symbolized by upper case letters.

Of course, AQUAD makes it possible to change the cut-off if the research question should require it. At the bottom of the transformation window, you will see a small box that shows (in red characters) the cut-off value that has been set. You can change this value step by step (in steps of 5) by clicking on it.

Remember: the higher the cut-off, the more of your data will be assigned the truth value "false" (for example, if the box shows "70", data with values below 70 will be considered "false" and only the values above will be accepted as "true". Changes to the cut-off value apply only to the table for which you made the change. When you enter data into a new table, the default value of "50" is used – until you adjust it to meet the needs of your study.

Now we are ready to start the transformation process. You select from a window the data table that should be transformed into a table with truth values. You set the cut-off or accept the default one and then click "OK". The following figure shows for the data table "Class-size_DT.csv" the result table with the truth values "Class-size_DT_TT.csv" (_TT stands for "Truth Table" and is automatically appended to the name of the data table):



	A	B	C	D
1	A	b	C	D
2	a	B	c	D
3	A	B	c	D
4	A	B	C	D
5	A	b	c	d
6	a	b	c	d
7	a	B	C	d
8	a	b	c	d
9	A	b	C	D
10	A	b	C	D
11	A	b	c	d
12	a	B	c	D

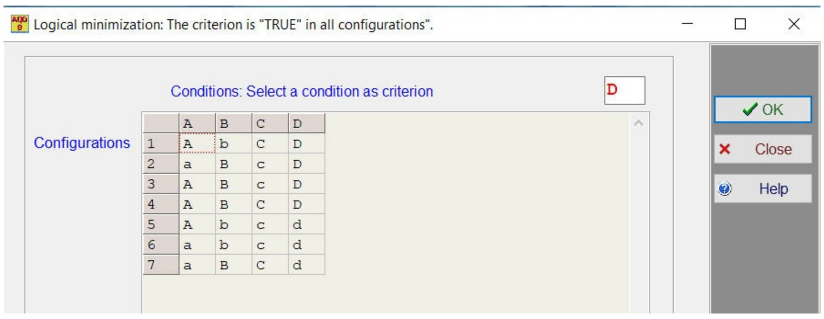
Truth values (critical percentage: 50%) 50

10.3 Finding implicants of "positive" and "negative" criteria

With the truth values we can now finally determine implicants. The submenu confronts us with the decision whether we want to pick out all cases in which the condition to be selected as criterion is "positive", i.e. "true" – or whether we want to determine the implicants for those cases in which the criterion is "negative", i.e. "false". As we will see, both forms of minimizing conditional configurations complement each other very usefully. Let us first stay with the option "Criterion: TRUE".

After we have selected the name of the truth value table with which we want to perform the minimization, the table is loaded on the screen. Now we have to select one of the conditions (columns!) as criterion by clicking on the column header. Then AQUAD will pick out from all the truth values in our table the combinations for which the critical condition (our criterion, that is) is "true".

You may be surprised to see that there are now only seven combinations in the table, rather than 12 cases. AQUAD removes all redundancies from the table. That is, for cases with identical truth values, only one combination remains in the final logical minimization table as a proxy for the others.

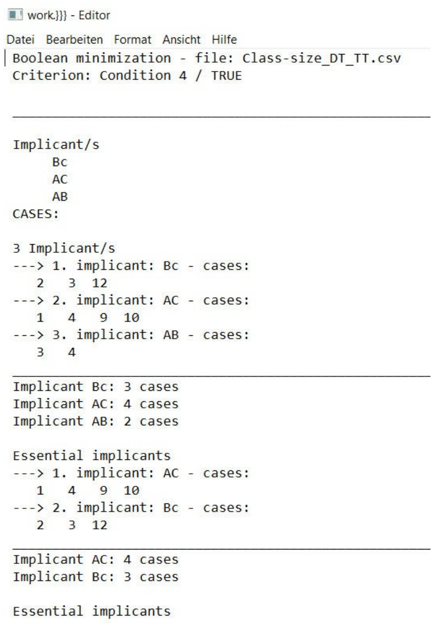

Logical minimization: The criterion is "TRUE" in all configurations".
Conditions: Select a condition as criterion
D
Configurations
OK
Close
Help

	A	B	C	D
1	A	b	C	D
2	a	B	c	D
3	A	B	c	D
4	A	B	C	D
5	A	b	c	d
6	a	b	c	d
7	a	B	C	d

In our case we chose the condition "D" as criterion. That is, we look for those configurations of conditions that may play a role in causing above-average school performance (criterion "D") along with class size. Or to put it another way: In our example, we chose condition "D" as a criterion because we wanted to find, among the various criteria, those that (together with class size) played a significant role in the cases in which we observed high school performance.

The fact that the minimization criterion has to be determined in each case points to another special feature of AQUAD: when constructing the data table, one does not have to decide a priori which characteristic should be the criterion later and therefore place it, for example, in the last column. This would only make sense in the context of research plans that specify independent and dependent variables.

For many qualitative research questions, on the other hand, such prior assumptions about causal links between conditions are the exception. AQUAD makes it free to pick any category (code, feature, meaning) from the total categories used and look for the typical configurations of the other features that occur along with the expression "true" (or "false") of the category selected as a criterion. Boolean minimization thus serves primarily heuristic purposes in AQUAD. Let us now look at the result in our example:



```
Boolean minimization - file: Class-size_DT_TT.csv
Criterion: Condition 4 / TRUE

________________________________________________

Implicant/s
     Bc
     AC
     AB
CASES:

3 Implicant/s
---> 1. implicant: Bc - cases:
   2   3  12
---> 2. implicant: AC - cases:
   1   4   9  10
---> 3. implicant: AB - cases:
   3   4
________________________________________________
Implicant Bc: 3 cases
Implicant AC: 4 cases
Implicant AB: 2 cases

Essential implicants
---> 1. implicant: AC - cases:
   1   4   9  10
---> 2. implicant: Bc - cases:
   2   3  12
________________________________________________
Implicant AC: 4 cases
Implicant Bc: 3 cases

Essential implicants
```

In our example, minimization for criterion "D" yields Bc, AC, and AB as the main implicants. That is, high performance is observed in school classes under the following conditions:

- At a high ability level (B) and a relatively short observation time (c) (Bc stands for the logical relation B and c); or
- with a low number of students (A) and a long observation time (C); or
- when the number of students is low (A) and the ability level is high (B).

If we take into account the overlap of the case mappings, we can summarize that in this study high performance is observed in all the cases where the classes were small and the observation was for a long period of time (AC), or in those where a high ability level was present with only a short observation period. In the first case, ability does not play an important role, and in the second case, class size does.

In complex condition configurations, the search for essential implicants may not lead to a complete result. This means that the condition configurations are not completely covered by the reduced configurations.

We not only get information about condition configurations, here for criterion "D", but also information about groups or clusters of comparable cases. As the example also shows, the main implicants are often redundant, i.e., they form overlapping groups of cases. There are cases that belong to two different configurations, namely cases 3, 4, and 10.

What is the "negative criterion" option all about? As you correctly guess, the only difference is the fact that in this case AQUAD chooses a combination of conditions for minimization for which it is true that the chosen criterion is "false" in each case. As a result, we get the "negative conditions" of the criterion, i.e., we get those implicants that are related to the logically "false" expression of the criterion condition.

In our example, we see that low school success (d) is observed in large classes with moderate aptitude levels (ab), or in classes with moderate aptitude and when the experimental period is long (bC), or in large classes and when the experimental period is long (aC). If you perform this minimization with your example table, you will see that main implicants do not necessarily have to be redundant. We get a result with overlapping groups in this case; there is no difference here between main implicants and essential implicants.

10.4 What else implicants can be used for...

From the previous examples it has become clear that the minimization logic in AQUAD is used to compare qualitative analyses, hence the name "Qualitative Comparative Analysis" for the procedure. In particular, it can be used to compare contexts of meaning or configurations of categories

- if you are examining a large number of individual cases, or
- if you want to perform a meta-analysis of qualitative studies.

This makes commonalities and differences between individual cases or studies clearly visible. During the final steps, when we want to summarize or group our results to differentiate between text types or speakers, the process of logical minimization seems inevitable.

In the effort to uncover causal relationships beyond the limits of case-specific conditions, we need to identify a category across all cases that is the effect, so to speak, in whose possible causes we are interested. For example, Marcelo's (1991) research was interested, among other things, in finding causes of young teachers' disciplinary difficulties in their classes. In the logical formulation "if ... then ..." of empirical causality, the effect category defines the "then" part. Very specifically, the analysis dealt with the problem of "When something yet unknown occurs, beginning teachers have discipline problems." What we are interested in is the content of the "if" part, i.e., the groups or configurations of categories that together "cause" the critical effect. Since such "if-then" links are known as the logical relation of "implication," we also say that the propositions within the "if" part imply the "then" proposition. We therefore call these causal propositions the "implicants".

To illustrate the procedure, we again take examples from Marcelo's (1991) study of novice teachers. As noted earlier, the author found that many of these young teachers spoke very frequently about discipline problems in their classes, but by no means all mentioned this problem. In looking for critical differences that could be used to explain why some of the young teachers experience discipline difficulties, the analysis focused on six categories, substantive foci of the interviews, namely teachers' statements about:

A reflection about themselves,
B teacher-student relationships,
C teaching methods,
D discipline problems,
E student motivation, and
F classroom climate.

Analysis of the configurations for condition D (discipline problems) as a criterion revealed three groups of implicants:

D = ABC + ACEF + abcef.

From this reduction, we can learn that we need to distinguish three groups of beginners with discipline problems (D). The interpretation of these groupings seems extremely relevant for the organization of teacher training:

• Configuration ABC: A first group can be characterized by configuration ABC, i.e., these teachers reflect on themselves, on teacher-student relationships and teaching methods - but not on student motivation and classroom climate.
• Configuration ACEF: A second group, which can be characterized by configuration ACEF, talks about themselves, about teaching methods, about student motivation and social climate - but does not seem to reflect on teacher-student relationships.
• Configuration abcef: The third group, characterized by the configuration abcef frequently mentions discipline problems in the interview, but almost none of the other central categories!

This form of implicant analysis yields clusters of cases. For theoretical and methodological reasons, but also for practical reasons, we might want to shift our gaze away from general results of different conditional configurations of a critical category and instead focus on individual cases. In other words: We should go back to the interviews - and especially to the interview transcripts of teachers who belong to one of the subgroups with discipline problems - and focus on very specific codings. When you study the list of cases you get as a result, it very much encourages you to compare permanently even at this level of analysis and opens the way from a high level of abstraction back to the concrete case-specific formulations.

Besides helping to summarize the results, logical minimization provides important heuristic functions in early stages of the analysis. In analyzing the implicants of conditional configurations, we can get valuable clues as to the direction our interpretation can take. Suppose our study included 50-60 interviews. Also, as we went through the first 10 interviews, we developed five important categories for interpretation. We simply call these categories here A, B, C, D, and E. These categories set in motion interesting, but unfortunately contradictory, conjectures about important meaning relationships in our data. You probably would not want to continue in your interpretive efforts, coding text after text, if you began to doubt your approach at this stage. After working for a long time, who might want to end up realizing that something crucial was overlooked from the beginning?

Instead, you can take the coding from the first ten or twelve interviews and identify a particularly important category as the "positive" criterion, then subsequently as the "negative" criterion of logical minimization, and have AQUAD find the implicants of that criterion. Assuming condition A is the critical criterion, we first find the implicants for all those cases where category A was considered important or "true" for the interviewees. Thus, we learn the condition configurations of B, C, D, and E that belong together (perhaps even are the cause) of the occurrence of condition A. The resulting configurations might be as follows:

A = BD + BC + bcd

Then we activate the option "Negative criterion" to find condition configurations of B, C, D and E in the interviews in which the criterion A was never mentioned or was considered unimportant, i.e. "false". Suppose we now find the following configurations:

A = BD + cde

Obviously, *there is a contradiction here*. The configuration BD is found both as a conditional configuration for statements under the critical condition A = true and for statements under the condition a = false. After only a short time, i.e., after 10 out of 60 interviews have been interpreted, we thus get an important heuristic hint on how to differentiate our coding system. Probably we forgot to include evaluative aspects when using categories B and C. Suppose our interviews were about student teachers' experiences during a classroom practicum. For example, we coded statements about their observations regarding teacher-student interactions, but we omitted from our codes whether a student teacher experienced a particular interaction as successful/positive or unsuccessful/negative.

Thus, codes referring to category B or D are found in most interview texts, but without reference to the qualitative expression of A. However, if we now take into account whether the observation of an interaction was judged by the interviewee to be positive and therefore likely to fit condition A, or whether an interaction was judged negatively and therefore did not fit condition A at all (but did fit a), then we can resolve the contradiction in a very short time. As you can see, as a heuristic tool, configural analysis can facilitate the work of discovering adequate categories even when only a few texts have been analyzed.

Finally, we should also think of the meta-analytic possibilities that logical minimization opens up for us. Meta-analyses need not be quantitative, although Glass, McGaw, and Smith (1981) explained this as "undeniable." Depending on the type of data, qualitative or quantitative meta-analyses are possible, and the two approaches may not differ in the rigor and systematic nature of the comparison of study results. A tool such as AQUAD can be used to meet this requirement for qualitative meta-analyses.

The sources of error in comparing studies, listed in a sobering compilation by Jackson (1980), cannot be completely eliminated even with computer support for analysis. As emphasized above, the researcher performs the analysis; the computer serves only as a tool. If, as Jackson (1980) points out, only a negligible percentage of those authors who use summaries of previous studies do not also discuss them critically, the tool is of no help. On the other hand, with computer support, the authors themselves should notice if they have only picked out specific constellations of conditions from findings of other studies or case analyses, or if they have overlooked contradictory configurations. Jackson's (1980, p. 459) charge that most comparisons are "conducted with far less rigor than is presently possible" takes on special poignancy with the availability of software such as AQUAD.

10.5 Function of implicants in the process of theory building.

How should researchers conceive of implicants as results of configural analyses? Do implicants serve as evidence that helps prove or reject theories behind them, as blueprints to reconstruct other people's worldviews, or to formulate theories that surface in the process of qualitative analysis? The answer, once again, is "neither ... nor." Since this is not the place to elaborate methodological considerations, Ragin's (1987) explanations of the dialogue between evidence and ideas in Boolean configurational analysis might be of interest as further reading:

The Boolean path to qualitative comparison ... is a middle way between two extremes, the variable-oriented and the case-oriented - it is the middle way between generalization and complexity. It allows researchers to do both, to include many cases and to estimate causal complexity (Ragin, 1987, p. 168).

Chapter 11: How to combine qualitative and quantitative analyses

11.1 On the debate about qualitative versus quantitative research methods

In the 1980s, one found arguments about "quality" and "quantity" as a recurring theme in the "Educational Researcher", a journal of the American Educational Research Association devoted to topics of a fundamental nature. The largely interrelated articles often read like a serial novel, even if the arguments were soon exhausted. Four types of texts could be distinguished:

(1) Texts that attempted to substantiate an irreconcilable opposition of quality and quantity in educational research (Smith, 1983; Smith & Heshusius, 1986);

(2) texts that articulated consternation about polarization and illuminated underlying misunderstandings (Shulman, 1981; Phillips, 1983; Tuthill & Ashton, 1983; Howe, 1985, 1988; Firestone, 1987) or developed alternative interpretations (Eisner, 1981, 1983);

(3) texts that responded to the incompatibility thesis with research strategies that incorporated quality and quantity in a complementary manner (Miles & Huberman, 1984, 2nd ed. 1994; Huberman, 1987); and

(4) texts that focused on qualitative approaches and sought to contribute to their systematization (Fetterman, 1982, 1988; Jacob, 1988).

However, a discussion of the goals and conditions of qualitative research, especially in comparison to quantitative approaches, did not seem to get off to a good start, at least in the field of psychologically oriented research. One got the impression that not so much complementary applications were worked out in factual reflection on the merits and problems of qualitative and quantitative approaches, but that a polarization process was underway. Even in thematically specialized research groups, one could observe the formation of opposing "methodological camps." In this process, the demarcation in terms of the type of data recognized also seemed to affect the research goals. Occasionally this led so far that questions and results of the "others" were no longer taken note of and certainly not discussed; one was content with the exchange of labels – for example, "soft" research here, "hard" research there.

Undoubtedly, the different methodological orientations do not define incompatible contradictions. There are enough convincing examples from research practice in which it has now been possible to simultaneously reconstruct the subjective world of the researched persons and to objectively reduce this information. Each research approach constructs a specific worldview, only some methods give the impression more strongly that one can simply find reality with them. Nevertheless, the strategy of delimitation does not seem to be overcome even today. Clearly and distinctly, Smith & Heshusius, 1986, p. 11) expressed this position at that time:

> "... if a quantitative researcher and a qualitative researcher disagree, is it even possible for them to talk to each other? The answer, at least for the moment, is a resounding No. The injunction that one must accept a particular result because it is based on facts will make little impression on someone who believes there can be no uninterpreted facts; on the other hand, the idea that facts are value-laden, that there is no appeal beyond dialogue and persuasion, will seem unscientific and insufficient to a quantitative researcher, to say the least."

The dichotomy and implicit mutual devaluation seem untenable. No matter what methods are used, research must be conducted systematically and produce reliable and valid results. And regardless of whether behavioral frequencies are counted or verbal responses are interpreted, the structures of meaning that the researcher constructs to guarantee the aforementioned criteria must be made visible. Now, there are undoubtedly precise rules for this in the field of quantitative methodology, while for some qualitatively oriented researchers it seems to be a question of whether one has to disclose the system of one's own construction processes in such a way that one oneself or others can control and, if necessary, repeat these processes. In our opinion, this is an indispensable requirement.

Unfortunately, textbooks of qualitative methodology, while describing in detail a wide variety of methods for obtaining data, treat the problems and procedures of data analysis rather stepmotherly. Sieber (after Miles, 1983) noted over 40 years ago that leading textbooks devoted only 5 to 10% of their coverage to data analysis; today, the ratio seems more favorable, but still very unbalanced. It is here that the development of qualitative data software can be a major advance. Computers as instruments of data analysis can support the processes of reduction and analysis of qualitative data in such a way that one can accurately control, reconstruct and, above all, communicate the course and results of the investigation. This could initiate a dialogue between members of the two camps, along the lines of how Miles & Huberman (1984, p. 23) described their own position. What mattered to them was that researchers proceed systematically and reach consensus on both content and methodology. They themselves felt that they were "right wing" qualitative researchers or "effeminate positivists"; for they recommended both paying attention to reality in interpretive approaches and, conversely, being clear about what one actually considers to be "data" and what role one's perspectives play in the process of attributing meaning to data.

In this context, the role of computer-assisted data analysis needs to be mentioned again, formulated in a slightly different way: Computers can be helpful in the process of qualitative research because they support communication, reconstruction, and control of the research processes. In this way, computers can help structure naturalistic approaches and systematize rigorous interpretations. At the same time, the use of computers supports the very pragmatic, but no less significant, goal of simplifying elaborate work processes.

Eisner (1981, p. 6) saw such an understanding of methods as limiting the possibilities and tasks of qualitative methods too much. He wanted to give more weight to "artistic"-intuitive processes: "Manifest behavior is treated primarily as a hint, as a stepping stone to get to another place. The other possibility is 'indwelling,' empathy; i.e., imaginatively participating in the experience of others. ... The difference is subtle but significant. In the former choice of object, the observable is treated in a kind of statistical way; one intuitively (or statistically) estimates the probability that this behavior has some special meaning or other. ... The latter choice is based on the ability ... to intuitively project oneself into the life of another in order to experience what that person is experiencing. ... A focal point, therefore, in artistic approaches to research is the meanings and experiences of the people who operate in the cultural network under study."

Remarks such as Eisner's (1981) are easily (and readily) thoroughly misunderstood if one implies that they aim, in reverse dogmatism, to replace fact with fiction. On the contrary, the mutual complementarity of the perspectives of qualitative and quantitative research, as also determined by the choice of subject matter, is very vividly illustrated. To give an example, which for quantitatively sworn readers perhaps already exceeds the limit of what is scientifically permissible: What knowledge do quantitative studies on the situation of mentally handicapped children convey - and what does one learn in comparison, for example, by reading Härtling's "Hirbel"? But what is the value of empathy for this fate if one has no secure basis for drawing conclusions in other situations? Quantity by itself is meaningless, quality by itself remains inconsequential. Especially research in application-oriented fields such as educational science or educational psychology depends on intelligent complementation of qualitative and quantitative approaches to their subject areas.

Thus, one can at first only shake one's head in bewilderment when reading the statement by Smith and Heshusius (1986, p. 11) quoted above as a "conclusion to the debate." However, one should not stop at shaking one's head. The implications of this dichotomization are too serious for the social sciences not to try to bridge the chasm that has been torn open here. One possibility seems to be the demand to open up and disclose the dimensions of meaning of the data qualitatively when using quantitative methods, to systematize and document the processes of interpretation when using qualitative methods, and furthermore to order the findings in a quantifying way. If this requirement is met, then one can and must draw on findings and procedures of the other methodological orientation under each of the two seemingly incompatible orientations, and do so for the benefit of one's own research. The prerequisites for this are transparency and regularity, i.e. real methodology in the practical implementation of methodological approaches in research.

In this respect, there is no such thing as "good" or "bad", no such thing as "hard" or "soft" research, but only methodological rigor or sloppiness - in both approaches, regardless of the quality of the theoretical anchoring of an investigation. For the opening as well as the acceptance of qualitative approaches, the development of systematic procedures for data analysis seems to be promising.

Fortunately, in the discussion about the possibilities of qualitative content analysis and in the development of practicable procedures, many suggestions can be found on how the complementary relationship between quality and quantity can be systematically taken into account. For example, as early as 1988 Mayring proposed an overarching phase model for optimizing the relationship between qualitative and quantitative analysis. According to this model, content analysis starts from qualitative determinants of the research question, the critical terms or categories, and the instruments. Depending on the goal and/or subject of the analysis, the analytical instruments can be used with recourse to quantitative methods (e.g., determination of word frequencies, cluster analysis of critical terms). The results of this step are related back to the research question in the final phase, conclusions are drawn; this again requires qualitative-interpretative procedures. Villar and Marcelo (1992) demonstrate and discuss, using their own research as an example, different ways of combining qualitative and quantitative methods in the design of investigations, in data collection, and in data analysis.

This scheme seems to us quite exquisitely to set the stage for fulfilling Lisch and Kriz's (1978, p. 46) demand, also made long ago, not to eliminate but to make explicit the subjective parts in content analysis: "By making the content analyst as aware as possible of his decisions in reconstructing social reality and by communicating them to his scientific community . . the interpretive framework becomes intersubjectively comprehensible and verifiable, and thus the question of the classification of the content-analytical results in an action-relevant theory becomes decidable. ... The specific experience of the content analyst with a text is made communicable, reconstructable, and thus as far as possible re-experientializable."

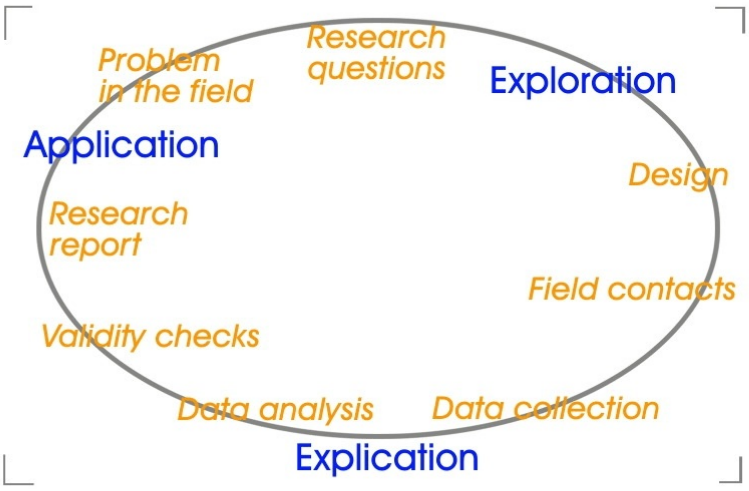
Both the methodological developments of recent years and the research practice in the social sciences – at least outside Germany – now fortunately show that research is increasingly oriented towards the challenges of the respective research questions instead of the framework of specific methods. The optional use of qualitative and quantitative methods according to a strategy of "mixed methodology" (Mathison, 1988; Tashakkori & Teddlie, 1998) promises to succeed in dealing with those aspects of a problem, of a complex research question, which could not have been investigated without this orientation due to the repertoire of methods available in each case – or which would not have been perceived at all due to methodological blinkers. The fact that a handbook on approaches and applications of "mixed methods" (Tashakkori & Teddlie, 2003) has also been published suggests a fruitful unfolding of this approach in the practice of social and behavioral science research. Recent reviews can be found in Mayring, Huber, Gürtler, and Kiegelmann (2007) and Gläser-Zikuda, Seidel, Rohlf, Gröschner, and Ziegelbauer (2012).

11.2 "Mixed methods" - buzzword or research strategy?

Qualitative and quantitative methods are thus not mutually exclusive, but rather have a complementary relationship. First, it is of course true that the choice of method depends on the question a study is intended to help answer. If you want to estimate how many people will vote for a certain party in an election or want to buy a certain product in the course of the next six months, you are interested in the most reliable numerical values possible as a result. If, on the other hand, you want to find out why people want to buy a certain product or why they prefer a competitor's product, you will prefer differentiated evaluations from the customers' subjective point of view.

If one takes a closer look at the two studies outlined, where in the first case there is a percentage value at the end with an indication of the confidence limits, i.e. a quantity indication, and in the second case a compilation of attributed qualities, then one notices that qualitative and quantitative activities occur in both studies. However, in the first case the role of qualitative procedures is usually not discussed, in the second case the role of quantitative procedures.

The output of research is a problem, in the social science field usually a problem in an area of life or social action. The more or less fuzzy perception of this problem gets contours in the formulation of research questions. In these phases, hypotheses are not tested quantitatively, but the first step is to specify questions. In the further exploration, hypotheses must now be formulated in quantitative studies and a research design must be drafted as well as methods selected with which the (quantitative) testing of the hypotheses is possible. In the case of qualitative studies, the design focuses on methodological possibilities for obtaining further information about the problem area in such a way that scientific hypotheses can be formulated about it. Whether one works with tests, rating scales, questionnaires, interviews, review of diaries, etc. as a data collection method – in any case, one needs access to the field in which the information can be collected. Establishing field contacts, building trust, and eliciting willingness to participate in the study are necessary research activities beyond the polarizing view of qualitative and quantitative methods.



We only want to mention, but not discuss further, the fact that of course the quantitative instruments used in quantitative studies in the following phase to explicate the problem were not themselves developed exclusively with quantitative methods. Even in the construction of a highly standardized test, someone at some point must have had the idea and decided that the test items could capture what the test was intended to measure. In general, it is important to remember that researchers themselves and the roles they ascribe to themselves in their undertakings from their epistemological orientations determine the range of research activities that are reflected as significant in a given research context. Maxwell (2002) emphasizes the influence of personal characteristics of the researcher on the one hand, e.g., prior experiences, beliefs, personal goals, and on the other hand, the personal relationships of the researcher to the research field, i.e., primarily to the researched. How researchers conceptualize and engage in a study depends on their identity and perspective as well as on their relationships to the subjects of the research. The reverse is also true for the researched.

Whatever methodological approach one has chosen, the empirical data must now be collected with the chosen instruments and then analyzed. The results must be checked for validity - here, too, qualitative and quantitative approaches complement each other, depending on the validity criterion. Finally, a research report must be written. The readers of the report and/or the researchers themselves will then consider whether and how the results can contribute to solving the problem being researched - and they will also not be guided by the dichotomy of qualitative vs. quantitative methods.

Even if one agrees in principle with such analyses and concedes the mutual complementarity of qualitative and quantitative approaches at the abstract level of the overall context of phases in the research process, the question remains whether qualitative and quantitative methodology can and may be systematically combined at the concrete level of individual sections, especially data collection and analysis. We will see below that such combinations are already implicit in the qualitative research process, explicitly provided for as an important feature of the design in planning, conducting, and validating entire studies.

11.2.1 Implicit combination of methods in qualitative research processes

Above we described the interplay of differentiating and generalizing strategies of coding with reference to Shelly and Sibert (1992) as a cyclical process of inductive and deductive inferences. Mayring (2001) notes the systematic generation of categories and assignment of data segments as a common goal of both processes and concludes, "When working with categories in such a systematic way, it suggests itself to conceive of these assignments as 'data' and to work on them quantitatively in a second step of analysis" (para. [16]).

The findings of a first, qualitative-interpretative reduction of the initial data are thus, for example, counted and ordered by frequency, expressed in percentages, arranged on ordinal scales (much - medium - little), compared between file segments and/or cases using statistical procedures, etc. The contribution of these new results to answering the research questions must then be interpreted qualitatively again in a third step (Mayring, 2001, [17]).

11.2.2 Explicit combination of methods in qualitative/quantitative research processes

According to Villar and Marcelo (1992), the progress of social science research is based precisely on the fact that one uses the variety of methods according to the research problem - and does not limit the problem and empirical approaches according to the possibilities of a methodological approach. They point to Greene, Caracelli, and Graham's (1989) suggestion of collecting data in a "mixed-method" design. As Villar and Marcelo (1992) go on to point out, this opens up the research process to the requirement of "triangulation," which at least increases the validity of research by incorporating different methods, data sources, and researchers (Mathison, 1988).

In Villar and Marcelo (1992), this idea is concretely unfolded for the central phases of empirical research, namely access to the field, methods of data collection, and data analysis. The authors show concrete procedures which they themselves have used in their studies in order to make the advantages of qualitative and quantitative methods complementary fruitful for their work:

Combination of methods in the phase of field access

When it is important in a study to work with a representative sample from a defined population (e.g., "the" elementary school teachers; "the" high school mathematics teachers; etc.), one often combines a quantitatively based selection of subjects with checks for representativeness based on qualitative characteristics of the sample thus drawn. As an example, Villar and Marcelo (1992) cite a study of the socialization of beginning teachers in which participants were selected from a list using a table of random numbers. They then examined the composition of the sample according to the gender of the subjects, their subject, the type of school, and the socio-geographic location of the school site-all qualitative characteristics.

But even in the case of single-case studies, which are more common in qualitative research, the "mix of methods" in this phase can increase the precision of the selection of individual cases and thus, on the one hand, help to make the best use of the scarce resources of a research project and increase the validity of the results. In "theoretical sampling," one selects cases not randomly but purposefully so that, depending on the design of the study, they are typical or critical to the research question(s), extreme or unique, or simply particularly informative because they may not have been previously accessible (Yin, 1989). Quantitative diagnostic tools can be advantageous in this regard: If, for a qualitative study of the subjective effects and individually stimulated learning processes of using particular teaching methods, we suspect that high-performing and low-performing students will express themselves differently, we can use the results of appropriate school achievement and/or intelligence tests to specify the selection of only a few subjects whom we can interview or observe for practical reasons.

These examples show, as should also be expressed in the process model in the diagram above, that decisions cross over and intertwine the various stages of the process. Thus, of course, decisions about empirical methods and instruments are made at the design stage, and then have practical implications at the data collection stage.

Combination of methods in the phase of data collection

Here, too, we must start with the formula: "It depends ..." - on the research question namely, if one wants to judge which combination is to be recommended. For the area of qualitatively oriented research on teaching/learning processes in initial and in-service teacher education, Villar and Marcelo (1992) cite participant observation, the in-depth interview, and the class diary as commonly used methods of data collection. Approaches to impact research in this area, on the other hand, usually make use of observation without participation and determine the frequency of certain phenomena according to predetermined category systems.

As an example of one of the possible combinations, the authors describe a quantitative study in which classroom practices at the University of Seville (Villar, 1981) were recorded by non-participant observation (analysis of tape recordings with predetermined analysis systems), participants were interviewed to obtain information about their subjective views, and then were asked to complete (quantitative) rating scales about their beliefs, problems, and the social climate of the seminar. With this combination the mutual supplementation of the different findings in the sense of a deepening of the information from the observation by subjective information in the interview is possible as well as a (methodical) triangulation by matching the data from interviews and rating scales (see below as well as Mayring, 2001).

Combination of methods in the phase of data analysis

Combinations of qualitative and quantitative analysis techniques are possible both in the analysis of data from a specific methodological approach, e.g., data from interviews, and in the analysis of data obtained with different methods, e.g., in the combination of interview and questionnaire.

An example of the first situation has already been described in detail above. A simple frequency analysis helped to identify the critical content of a large number of interviews. Later, correlation analyses still verified relationships between critical statements (cf. Villar & Marcelo, 1992).

The second type of situation is characteristic of triangulation analyses. On the one hand, under a qualitative perspective, the specific contributions of different methods can be used in the analysis (cf. Medina, Feliz, Domínguez & Pérez, 2002; see below); on the other hand, qualitative and quantitative analyses, especially dimension-analytical methods such as factor or cluster analyses and regression methods can serve well to identify or confirm central patterns of argumentation. With an emphasis on the quantitative approach, Schweizer (2004) describes in detail, using the first chapter of Cervantes' "Don Quixote" as an example, how relevant hermeneutic findings (e.g., here on the importance of reading for individual development as well as on possibilities of accessing motives and interests of the real author by looking behind his fictional world) can be elaborated via the (quantitative) analysis of word forms/vocabulary and assessment of their contribution to text comprehension.

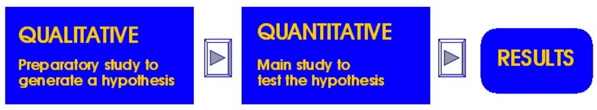
In a final phase of data analysis, which we refer to in the AQUAD context with Ragin (1987) as "implicant analysis" or "logical minimization" or "qualitative comparative analysis," the goal is to generalize the interpretive findings beyond the individual data set or case. If we ask what is general or typical about a particular case, we get in response details of relevant features it has in common with a class of similar cases. Conversely, if we ask what makes a case seem unique, the answer will point us to attributes that distinguish that case from the others. We are satisfied with the classification of cases into classes only when each of these classes contains only cases that are as similar as possible to each other in critical attributes, but at the same time are thereby as different as possible from the cases in other classes. With the process of classification, the goal is not only an order and representation of the momentarily given cases, but one tries in this order implicitly or explicitly to clarify connections, to make links between experiences, to build up expectations, to make predictions possible. Examples of this can be found in chapter 11 and in Huber (2001). Mayring (2001) sees in the "designation of the individual case as typical for a particular subject area ... a first, quantifying generalization step..." [18].

11.2.3 Combination of methods in research design

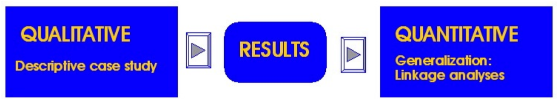
At the level of empirical study design, Mayring (2001) has proposed four models of combining qualitative and quantitative methods, which he calls the pre-study model, the generalization model, the in-depth model, and the triangulation model. We assign these combinations to two types and complement them with a third combination type:

(1) Macro sequences of method combination:

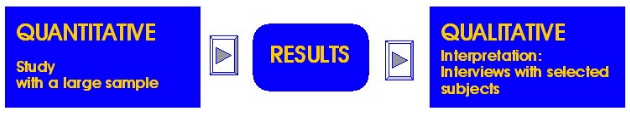
• The model of preparatory studies:
This model describes the approach according to which a kind of division of labor used to be postulated in "traditional empirical scientific understanding," "... whereby qualitative methods are used primarily at the beginning of a research project in the sense of exploratory studies ... are used. They are intended to clarify the research field, to differentiate problems and questions, to formulate preliminary hypotheses, and to test the practicability of research methods" (Krapp, Hofer & Prell, 1982, p. 130). This view of merely assisting "actual", quantitative research does not do justice to the complementary function of qualitative approaches in empirical research. Nevertheless, it becomes clear in this model that quantitative studies can hardly do without qualitative grounding. Schematically, the relationship in the preliminary study model looks like this (cf. Mayring, 2001 [21]):


QUALITATIVE
Preparatory study to generate a hypothesis
QUANTITATIVE
Main study to test the hypothesis
RESULTS

• The model of generalization:
According to this combination approach, a genuine, independent qualitative study is conducted, the empirical results of which are already ascribed significance in their own right. However, one does not want to or cannot – mostly in the context of application (phase of application; see above) – draw any significant conclusions for practice-changing measures from the relatively narrow data basis of qualitative studies before one has quantitatively verified the generalizability of the findings and statistically validated their probability or the strength of the effects of new measures. Mayring (2001)gives the example of a case analysis, the results of which are verified in a representative follow-up study:



• The model of profoundization:
The stages of the design are reversed in this model, i.e., a representative quantitative survey and data analysis is followed by a smaller, case-centered qualitative study, the results of which should help to better understand the significance of the quantitative findings (cf. Huber, A. A., 2007).


QUANTITATIVE
Study with a large sample
RESULTS
QUALITATIVE
Interpretation: Interviews with selected subjects

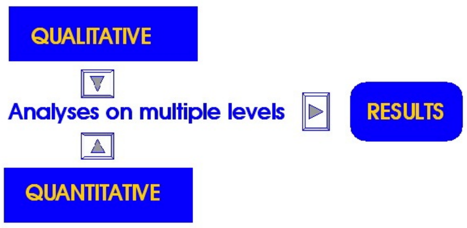
Many studies within the youth research group of J. Held at the Department of Educational Psychology at the University of Tübingen are designed according to this model (cf. Held, 1994; Kiegelmann, Huber, Held & Ertel, 2000). In a project on "Political Orientations of Adolescent Workers" Held, Horn & Marvakis (1996) investigated the reasons of adolescents for their orientations. In the first step, the authors presented a questionnaire to the young people; in the second step, the young people helped to interpret its results in follow-up interviews, thus deepening and concretizing the quantitative findings.

In this study as well as in an earlier one by Held and Marvakis (1992), another effect of the profoundization model becomes visible, its linking of explication and application, of knowledge and application in the research process: Characteristic of the methodological procedure was that the contact to the interviewed young people was not yet finished with the execution of a questionnaire survey. This was followed by feedback and discussion of the results in the schools and companies. The results served as a stimulus for group discussions and in-depth interviews, and the quantitative instrument became a medium for potential change. The research process thus becomes a practical educational process at the same time.

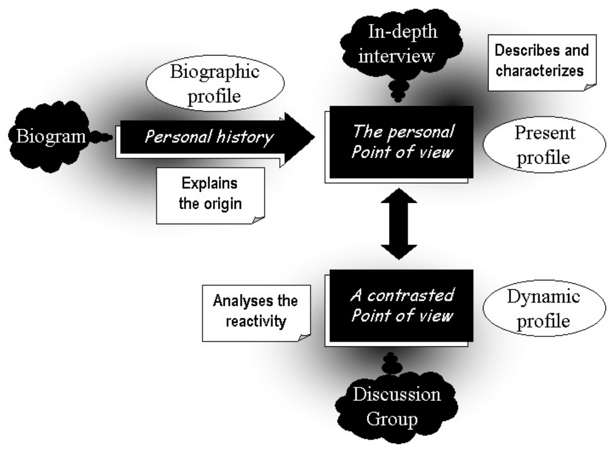
A. A. Huber (2007) gives a detailed example of how in a study on cooperative learning the extensive quantitative findings can be better interpreted by this model and partly unexpected results can be explained.

(2) Simultaneous application of qualitative and quantitative methods

• The triangulation model:
In this approach, different methodological approaches to the research field stand simultaneously and equally side by side. The point is to compare the more or less varying answers that result from the perspective of different methods to the research question and to determine the intersection of the individual results as the final finding.



At the level of methodological triangulation (see above), all conceivable combinations can occur, i.e., in the field of qualitative research, the combination of qualitative and quantitative methods or the combination of different qualitative methods. They are not used in a sequence necessitated by the design, but simultaneously or in parallel. A well-argued example of the latter triangulation is given by Medina, Feliz, Domínguez, and Pérez, (2002, p. 178):



The biogram helps to open up the personal life or professional history, the in-depth interview contributes to a differentiated understanding of individual relevant aspects, and finally the group discussion makes it possible to relate and compare the subjective perspectives of the individual participants in the study. In the intersection of the temporal-biographical perspective, the personal-individual perspective, and the dynamic social-interactive perspective, we finally find balanced answers that can be assessed in their scope through the different perspectives.

Several articles in Tashakkori and Teddlie (2003), especially the article on integration rules by Erzberger and Kelle (2003), provide information about other variants of the triangulation model, e.g., combining the perspectives of several researchers or different theoretical orientations. On the relationship between "mixed methods" and triangulation, Gürtler and Huber (2012) provide an overview.

(3) Micro-sequences of method combination

In both implicit and explicit method combination, depending on the decision for a particular strategy, one is often faced with the question of how to analyze the available qualitative data using quantitative methods as well. Answers to this question can be summarized in the third type of combination, the generation of micro-sequences of method combination. In each case, the way to achieve this is through reduction and mapping of selected aspects of the data material on a quantitative scale, usually on a nominal scale. With regard to the concrete procedure, this means that selected elements of the data material are counted, the frequencies are compiled in lists, and finally these lists are converted into table form for statistical analysis. The results of the quantitative analysis often form the starting point for further qualitative interpretations, e.g. in the form of meta-coding and subsequent analysis of implicants.

We have already discussed the determination of word frequencies above, more information follows in the next section. In Section 11.4 we also describe the determination of code frequencies. Finally, in Section 11.5, we discuss micro-sequences of method combination using two specific examples.

We see, the advantages of combining qualitative and quantitative data/methods can be used on the primary level of the given data, here the individual words, but also on the secondary level of the results of our interpretation, here the codings. We look at the procedure in detail below.

11.3 Word frequencies

Berelson (1952) developed the basics of content analysis, which for him was a quantitative procedure based on determining the frequency of elements of manifest content. We have already touched on the fact that non-quantitative procedures are certainly significant in determining and selecting elements relevant to the analysis. Let us focus here on the "mechanics" of the process.

Access to the counting functions in texts can be found in the main menu of AQUAD in the function group "Methods of Analysis" -> "QUANtitative Content Analysis". The following figure provides an overview of how lists of critical words can be created. The "Count words" can be selected in the preceding submenu of the subfunctions. The details can be found in chapter 8.

11.4 Code frequencies

The combination of qualitative and quantitative analyses is especially interesting with results on the secondary level of analyses. The approach is to count the frequency of codes. All types of codes can be counted. Especially counting of sequence codes, which can be additionally inserted into the code files with the various forms of analysis of "links" as a result, should be interesting in advanced stages of the analysis.

For counting, the program needs a listing of the codes it should look for in the code files. This list can be compiled anew and up-to-date at each counting by selection from the code register. The alternative would be to generate a list of relevant codes once by selection and save it. For counting, you then simply load this list. Here we now take a closer look at this second possibility. So the sequence of steps starts with creating a list of codes that we consider important for a particular question:

① Main menu: "Retrieval" -> "Code Catalog" -> "Create Code Catalog".

Then we start the function for counting codes, load the list we just created, start the counting process and finally save the results in the form of a frequency list of the selected codes:

② Main menu: "Retrieval" -> "Count codes".

For the export and further use of these simply listed frequencies in statistical programs, we need a table representation of the results, and preferably in the form "CSV", i.e. as "comma separated values". This table is automatically created and saved from the frequency list of codes. All data appear here row by row and separated by commas. The first column contains only letters as labels for the variables for export to statistical programs, e.g. SPSS, R or the AQUAD module for exploratory data analysis.

11.5 Combination of methods: Example

In the following section the combination of qualitative and quantitative methods will be demonstrated using the example of a study on humor by Leo Gürtler. Mixed methods seem interesting to us especially when it comes to the analysis of large data sets (e.g. questionnaires with open answers), where certain partial questions are of a quantitative nature and both approaches can complement each other in an integrating way. In each case, the focus is on the logical connection of the process of theory modeling and testing.

In a larger questionnaire study (N = 363, cf. example above on sequence coding) with students from Realschule and Gymnasium, subjective theories on humor in the classroom were surveyed. The central research questions were:

- How do students experience humor in their classroom?
- Can different views and courses of action be identified in this context and how are they connected?

The theoretically derived questionnaire construction developed nine thematic areas, each of which resulted in an open question. These nine themes were given speaker codes in the later coding with AQUAD to allow selective evaluation according to these themes. Example questions included:

Question 1:	What is humor to you ?
Question 6:	Maybe you also know negative experiences with humor (in class). if you know them, what was going on ?

The answers were written into individual text files per person and an AQUAD project was created from them. Then the answers were coded according to content and structure. Content here denotes, for example, codes such as "outward humor" or "social atmosphere/climate", whereas structure was introduced as a validity aspect. This made it possible to check whether the questions were answered according to their meaning. Example categories here are "definition", a structural category, which is to be expected for question 1 – the question about the subjective definition of humor - and not, for example, for the question about action sequences of humorous teaching sequences. The question about sequences of actions resulted in categories such as "positive effect(s)" or "negative cause(s)."

This first coding process was followed by a second one, in which the content codes were combined into metacodes with the help of a qualitative content analysis. This reduced the number of codes to a manageable level (N = 105 content metacodes, N = 22 structural codes). The structural codings were fewer in number and qualitative content analysis was not necessary for these.

11.5.1 Variance analysis of code frequencies

From the memos written down during coding, the resulting metacodes, and through the subjective impression of the researchers, several hypothesis areas emerged that could be broken down into the nine themes. Additionally, it was noticeable that apparently girls produced far more text than boys and students from the Gymnasium more than those from the Realschule (although the latter could also be due to an age difference in the sample). This provided an initial reason for the use of quantitative methodology. A two-factor univariate analysis of variance was conducted with the factors gender versus school type and the following a priori contrast coding (Helmert contrasts after Fox 2002, p. 142 ff.):

	[1]	[2]	[3]
Male high school (mG)	- 0.5	+ 0.5	- 0.5
Male Realschule (mR)	- 0.5	- 0.5	+ 0.5
Female Gymnasium (wG)	+ 0.5	+ 0.5	+ 0.5
Female Realschule (wR)	+ 0.5	- 0.5	- 0.5

Thus, the following comparisons were made:

[1]: mG & mR versus wG & wR -> test for gender.
[2]: mG & wG versus mR & wR -> test for school type
[3]: mG & wR versus mR & wG -> test for interaction gender x school type.

The resulting linear model was highly significant, as were the gender and school type factors. The gender x school type interaction showed no significance, although a slight tendency to interact, as suggested by a graphical plot. Thus, it seemed justified and also necessary to conduct further qualitative analysis separately for the mG, mR, wR, and wG groups.

In order to adequately reconstruct subjective student understanding, linkage hypotheses were produced in large numbers. These were derived from theoretical considerations that formed through review of the data material, the resulting coding categories, and the memos produced. In general, these linking hypotheses were limited to the themes and were located within each question. Some exceptions, on the other hand, were also formulated across questions and thus yielded speaker comparisons, since the questions were assigned speaker codes as mentioned above. Now a few examples to illustrate this procedure:

```
Sequence code name
FR_HYPO_2-0-26:     AND    /$M2 Limits of humor
                    AND    MX_negative cause (-)
                    AND    MX_justification/ushg.
```

Here we checked whether negative causes were mentioned in question 2, "Limits of humor", and then justified, i.e., not only voiced but also justified by the students in terms of arguments. Note that we have included question 2 (it is, after all, available as a speaker code) in the sequence coding. By this little trick we always know later that a positive result of this sequence hypothesis is to be located exclusively at question 2. Formulating linkage hypotheses in this way can very easily open up new perspectives. Therefore, we recommend to include speaker codes in the linking hypotheses for hypotheses that refer to limited areas such as questions or even speech acts. To find all those text passages where the same sequence "negative cause" and "justification" occurs, but not related to question 2, you only have to put the first code to the back and link it to "NOT". Remember that "NOT"-links are not allowed at the beginning of a linkage hypothesis. The sequence code then looks like this:

```
Sequence code
FR_HYPO_X-X-XX:     AND        MX_negative cause (-)
                    AND        MX_reason/relation
                    NOT        /$M2 limits of humor
```

According to this basic pattern quite a number of linkages could be formulated and tested. The quantitative step that followed tried to encounter typical sequences in the answers across students in all groups (mR, mG, wR, wG) or across groups. That is, based on principles of cluster analysis we looked for prototypes in data structures. Contrary to an ideal type in Max Weber's (1988) sense, which never occurs in reality, a prototype is an empirical fact – we only have to find it. A prototype is defined as *the datum or data, which has/have minimal distance to all other data*.

The procedure follows the principles of cluster analysis in explorative data analysis. Normally it has two phases: (1) The distances between each datum and all other data are computed; (2) according to various criteria these distance scores are used to determine clusters. The first step leads directly to the identification of prototypes: If we compute the sum of distances separately for each data set, then – according to the definition – the smallest sum indicates the prototype. Or in other words: The prototype is the center of the data given.

The next step of analyses brings us back into the qualitative domain, that is, we are mixing methods again. Based on our knowledge of prototypes and contrasting data sets (that is, those with maximal distance to all other data) we continue with focused interpretations. In the following we talk about contrasting types as outliers or "runaways." Now, above all the following questions need to be answered:

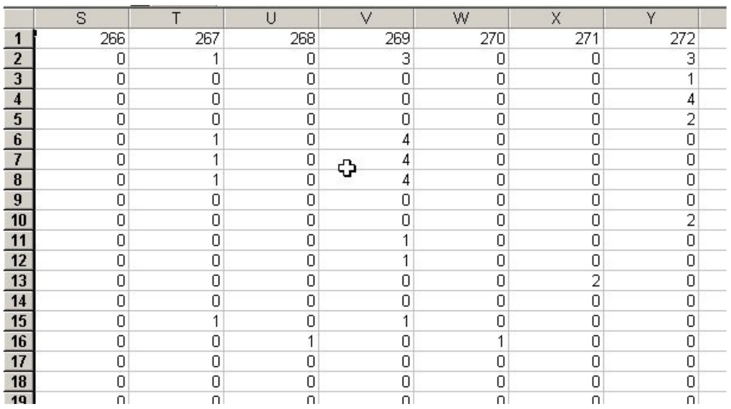
1. What is characteristic in a/the prototype(s) as compared to runaways?
2. Which codes are involved ?
3. Do we learn about anything special or systematic as a result of comparing prototypes and runaways?

As regards our example we expect answers telling us, which of our theoretically grounded linkage hypotheses are represented best or worst by our data. The best fitting data are prototypes, the worst fitting ones are runaways. Remember: *Do not concentrate your interpretations exclusively on prototypes, but pay attention also to runaways as components of equal importance. The most relevant answers to your research questions will be elaborated within the field of tension between the poles of typical and a-typical answers.*

In the following we want to give concrete advice how to put the results of data analyses with AQUAD into distance matrices and continue with quantitative analyses.

Data analysis: How to create distance matrices based on linkage analyses

As already explained above, we have to transform our codings into distance matrices which show how close or how distant linked data segments represented by a sequence code occur in our data. Basis of calculations are the frequencies of all sequence codes in the files of all persons. We have AQUAD count the code frequencies (Which sequence occurs how many times in the code files of which person?) and convert the resulting lists into tables. These tables then are converted int CSV format, because this format can be read by spreadsheet programs or – in our example – by the components of the language "R" (2004) for statistical programming. The final table should look like this example (part of the original table):



	S	T	U	V	W	X	Y
1	266	267	268	269	270	271	272
2	0	1	0	3	0	0	3
3	0	0	0	0	0	0	1
4	0	0	0	0	0	0	4
5	0	0	0	0	0	0	2
6	0	1	0	4	0	0	0
7	0	1	0	4	0	0	0
8	0	1	0	4	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	2
11	0	0	0	1	0	0	0
12	0	0	0	1	0	0	0
13	0	0	0	0	0	2	0
14	0	0	0	0	0	0	0
15	0	1	0	1	0	0	0
16	0	0	1	0	1	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0

Headline (Row 1): Name or consecutive number of linkage hypotheses = variables (here: 266, 267, 268 etc.)
Columns: Persons
Rows: Linkage hypotheses
Cells: Frequencies of linkage hypotheses

In case rows and columns should be reversed in your table, "R" offers an efficient option to transpose matrices.

Statistical analysis with "R"

"R" (2004) is a programming language dedicated to statistical analysis. You get it free on the Internet (under http://www.r-project.org). On this website you will find also information, introductions, and references to literature on how to work with "R." As regards distance matrices, a particularly helpful manual was written by Handl (2002). Nevertheless, just to give you an impression, we will describe a few commands:

```
table <- read.csv("file_name_of_our_frequency_table.csv", header=TRUE)
```

That's all you type to load a frequency table in format CSV and put it into *one* variable "*table*" in "R." Our table had column headers (consecutive numbers of linkage hypotheses), therefore the characteristic "header" is set to "TRUE". If your table has no labels in the first row, you have to set header=FALSE, otherwise the first row is neglected in later calculations. If you want to check what you got, the command `table` puts your matrix on the screen.

All you should do now is follow the instructions by Handl (2002, p. 83 ff). Please observe to standardize the matrix (your table) *before* you calculate distances. Thus you balance differences of standard deviations due to interpersonal differences (Handl, 2002, p. 97). Since we want to identify prototypes, we have to calculate the complete matrix of distances(Handl, 2002, p. 96 ff.; on p. 437 you find the program that performs this task). Principally should be found in the rows those variables, which are the objects of distance calculations. In our example we defined the rows by linkage hypotheses; the above screen shot shows hypothesis 1 - 19 as row headers. The columns are reserved for the "data carriers", that is, the persons in our example. Consequently, the cells are filled with frequency scores of linkage hypotheses for each student. Reversed matrices like our initial CSV-table are transposed most simply:

```
table <- t(table)     # reversal of rows and columns in a matrix named "table"
```

You should use "euclidic" distances as unit of distance calculations, except you have theoretically justified reasons to apply an alternative distance unit. The prototype is identified by comparing the sum of distances per row (that is, per linkage hypotheses). The diagonal cells contain the score ZERO, that is the maximal score of similarity or identity, because in these cells we find the distance of a variable to itself. Correspondingly, a score of 1 would express the maximum of dissimilarity.

Well, that is how we construct the vector of prototypes:

```
prototype <- apply(distance_matrix, 1, sum)
# adding the scores of a distance table or matrix labeled "distance_matrix" by rows and creating a  prototype
vector labeled "prototype"
```

This vector is saved together with the distance matrix in format CSV:

```
write.table(cbind(prototype,distance_matrix), file="prototype_table.csv",sep=",")
# saving the prototype vector together the distance matrix under the name"prototype_table.csv"
```

The new table can be loaded and opened in any spreadsheet program. The first (left) column contains the prototype vector. We sort the table according to this vector in ascending order. The resulting order of linkage hypotheses will indicate on top, which linkage hypotheses are most prototypical (highest scores) and at the bottom those linkages, which are least prototypical, that is "runaways." Now we should find out, which meanings are carried by these linkage hypotheses. In our study we found the following order of linkages (across students, independent of school types) as regards question 1 (*What do you understand by "humor?"*); the figure behind "position" indicates the amount of prototypical characteristics: "1" stands for a maximum of prototype qualities, "2" is somewhat less prototypical, etc.

Prototype position 1:	AND	M_externally_dir_humor	AND	M_laughing, joyful, fun
Prototype position 1:	AND	M_laughing, joyful, fun	AND	M_externally_dir_humor
Prototype position 2:	AND	M_classical humor	AND	M_laughing, joyful, fun
Prototype position 2:	AND	M_classical humor	AND	M_borderline / distance to (-)

What stands out is that the linkages placed on position 1 differ only in the sequence of their components. Since both received the same prototype score, we can treat them as equivalent. Thus we can state that our sample is characterized that our students conceive of humor as something that has to do with laughing, joy, and fun and has an external object (or is directed externally). This may indicate that humor is a social, interactive matter – not something happening alone at home in front of TV. Additionally, this finding backs the validity of our students' answers: They really had the classroom context in mind, when they filled in the questionnaires.

The linkages on position 2 reveal an understanding of humor as something that has to do with "jokes" (classical humor), which cause laughing or fun (second place within the linkage sequence) – but also negative experiences (see second linkage on prototype position 2).

Next we should gather information about the distribution of these linkages among our sample and then return to AQUAD and retrieve and compare the corresponding original answers of students.

Additionally, we would like to draw your attention to a publication by Oldenbürger (1981), who described how to dichotomize empirically a data base (p. 153ff.) by establishing a proximity matrix (our distance matrix is one of this kind) and distinguishing between concepts related to each other and concepts, which do not occur in linkages. Thus, it is possible not only to identify prototypes, but also clusters that – unlike the findings of "classical" hierarchical cluster analysis – must not be strictly separated from each other. This sort of splitting up a data base is directly built on empirical data and of course compatible to the procedures of "grounded theory" (Glasser & Strauss, 1965). The output is a 0-1-matrix, in which we can read directly whether a given code is related to any other codes (matrix score = 1) or not (matrix score = 0). Again, adding the scores across rows amounts to the already described prototype vector. However, here the maximum indicates the prototype, because "1" represents an existing link or minimal distance, while a matrix score "0" represents a lack of relations. There is a program called "OptSchnPr" available in in the "R"-module "Proxim" (Oldenbürger, 2004; see web link in the references), which was constructed to split up distance matrices. A further "R"-program to manage input and output of CSV-formated tables is available from the author of AQUAD.

11.5.2 How to identify types by quantitatively based analysis of implicants

Usually hypotheses and decisions about promising components of implicants are formulated qualitatively, based on impressions noted in memos, comparisons within and across original data, linkage hypotheses, etc. Sometimes there are only minimal differences or we are confronted with highly complex data, for instance, if our topic is really new and we cannot rely on any available empirical findings. In such a situation alternative possibilities to generate hypotheses would be helpful. One of these alternatives will be presented subsequently.

The necessary statistical tools are really simple. We begin by counting codes, that is, we count those codes or meta-codes, which signal some relation as regards content area or topic. Then we standardize (z-transformation) the frequencies of each code across subjects or cases and correlate afterwards the resulting z-scores. Beginning with extremely scoring codes (for instance, z-scores beyond a range of ± 2 units of standard deviation, which is above or below scores of ± 2 in a z-distribution) we select codes correlating significantly ($p <= 0.05$) with these "extreme" codes as components of code configurations (implicants) and decide about a criterion code (a possible "cause" or "predictor" of "effects" represented in implicants). However, as already described, we are not limited by a research design that postulates independent and dependent variables, thus enforcing determined configurations. Instead we state plausible, reasonable, but anyway for the time being undetermined components as well as criteria (= "predictors") of implicants.

In a following step we have to evaluate qualitatively, whether these configurations of codes are relevant and contribute to understand our research problem better. Finally we have the remaining selection of configurations tested in AQUAD – both applying the criterion in a positive sense (criterion = TRUE) and in a negative sense (criterion = FALSE). This strategy resembles a sort of double-entry accounting and may sensitize researchers not to focus their attention on positive findings only, but to derive knowledge also from "negative" configurations.

We repeat these steps until a well-founded classification or typology does not change relevantly, that is until we reach a state of analysis that is called "saturation" by proponents of Glaser and Strauss' (1965) approach of grounded theory. In other words: At this point we do not expect any longer to come upon substantially new discoveries that would necessarily modify the answers to our research questions.

Of course, we must neither overdo in testing numbers of various configurations and smuggle the notorious "shotgun strategy" from quantitative testing into the field of qualitative analysis nor should we build complex arguments referring to several people upon a single criterion of implicants. The first strategy would rise the probability to detect reasonable combinations "at random" and – even worse – accept them blindly. The second strategy is problematic, because interpretations based on very few implicants demand a degree of generality, which is not permitted due to the preceding data reduction.

What we recommend is to identify at least one configuration per case (person, interview, video, etc.) to raise the probability that relations between configurations across units of analysis can be found. Especially tensions or even contradictions between configurations ("Why do we find this here, but not in other cases?") stimulate further, empirically based conclusions by reverting to the original data. Interpretations then do not depend on a single analysis of implicants, but are derived from explicit efforts to establish a typology. And, not to forget, because Weber's ideal types are constructs not to be found in reality, we are not so much interested in types, but in relations of various types and differences between them.

An example will demonstrate subsequently how to put these considerations into practice. Gürtler (2004) got a hunch during the analysis of teachers' implicit theories on humor in private and professional contexts that one of his subjects had a highly idealized, spiritually slanted concept of humor. As described above, all (meta-)code frequencies were z-transformated and correlated to identify a metacode representing an ideal attitude towards humor. This metacode was used as criterion in logical minimizations. All metacodes that correlated significantly with the criterion were put into the model of implicants and tested, one after the other. These are the results:

```
A = M_awareness
B = M_ideal_attitude_tow._humor
C = M_communication_soc._IA
D = M_self-knowledge_insight
E = M_self-concept_int_ext
F = M_jokes_active_evaluation
```

Criterion:	Condition B / FALSE
Implicant/s:	ACeF [6 9], acde [4 8 10]
Criterion:	Condition B / TRUE
Implicant/s:	ACDEf [1]

Further analyses showed it was necessary to test similar configurations:

```
A = M_ideal_attitude_tow._humor
B = M_cog.activities_without_soc._IA
C = M_communication_soc._IA
D = M_people_unspecific
E = M_self-knowledge_insight
F = M_reflection
G = M_negation_of_concept
```

Criterion:	Condition A / TRUE
Implicant/s:	BCDEFG [1]

(There were no cases matching the criterion condition 1 = FALSE)

Both results led to interesting assumptions. One of them, for instance, corresponds to the researcher's impression during the interview that in this case humorous activities involving jokes are meaningless (ACDEf), while self-knowledge, social interaction/communication, and this person's self concept played an important role.

Taking into account also the negative criterion (Condition B = FALSE) showed that in one of the other configurations jokes were an important component, which made the criterion "ideal attitude towards humor" meaningless, that is FALSE. In this configuration (ACeF) the role of self-concept (from an intra-personal and an external point of view) is no longer important.

A second analysis with a slightly modified model did no longer include the code "awareness", but integrated "cognitive activities without social interaction" and "reflection" as predictors. More detailed analyses should clarify the meaning of this configuration.

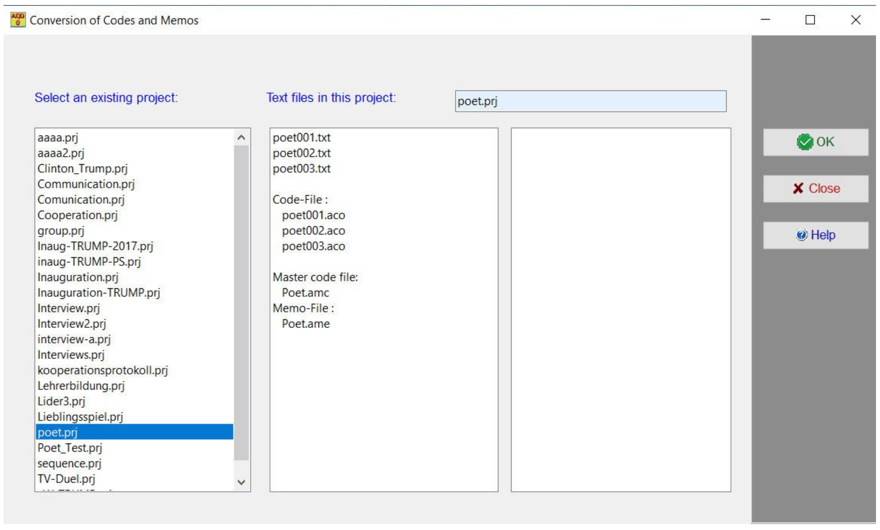
Chapter 12: Converting codes from AQUAD 7 to the AQUAD 8 format

AQUAD 8 cannot work with the codes from the previous version 7 at first, because earlier the codes contained not only 60 characters, but additional information. With this conversion function you achieve that the old files are adapted to the conditions of version 8.

However, the conversion will only run without problems if you have kept to the conventions valid there when working with the old version. In detail, the conversion routine expects that

• the text, sound, video or image files of your project have been copied to the root directory (e.g. C:\AQUAD_8), the corresponding project, code and memo files to the ..\cod subdirectory of AQUAD 8 (e.g. C:\AQUAD_8\cod);
• the names of the text files end in ".txt".

When you call the conversion routine, the window shown below appears. On the left, the names of all available project definitions appear . In the window on the right, select the name of the Version 7 project you want to convert, in this case "poet.prj". In the middle box you will find after the selection the names of the (text) files, code files, the code register and the memo file - if available. If you then click on the "OK" button in the right margin, the rest will run automatically:



Conversion of Codes and Memos

Select an existing project:

aaaa.prj
aaaa2.prj
Clinton_Trump.prj
Communication.prj
Comunication.prj
Cooperation.prj
group.prj
Inaug-TRUMP-2017.prj
inaug-TRUMP-PS.prj
Inauguration.prj
Inauguration-TRUMP.prj
Interview.prj
Interview2.prj
interview-a.prj
Interviews.prj
kooperationsprotokoll.prj
Lehrerbildung.prj
Lider3.prj
Lieblingsspiel.prj
poet.prj
Poet_Test.prj
sequence.prj
TV-Duel.prj

Text files in this project:

poet001.txt
poet002.txt
poet003.txt

Code-File :
poet001.aco
poet002.aco
poet003.aco

Master code file:
Poet.amc
Memo-File :
Poet.ame

poet.prj

Chapter 13: Explorative-Statistical Analysis

Tukey's (1977) module on descriptive-exploratory statistics shows that one does not need to calculate significances to discover significant relationships and differences in data, but that creativity and case adequacy as well as data transformations are sufficient to derive substantive hypotheses for further investigation in combination with various sources of information. The guiding principle is always the idea that instead of testing confirmatorily, it is better to examine data for their actual content in order to identify structures and patterns. We propose to do without inferential statistics in qualitative data analysis and still draw reasonable and substantive conclusions.

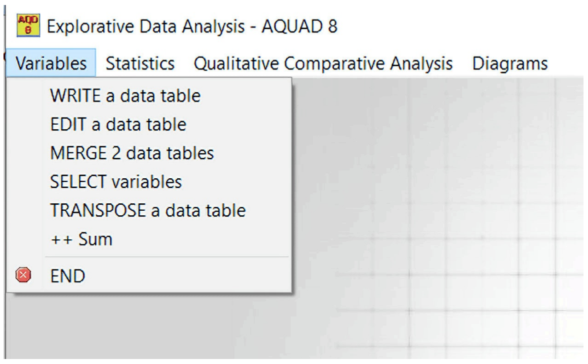
In the "..\prg\" subdirectory of all Aquad 8 modules, you will find the separate module "aquad_eda.exe" for this purpose. Its menu provides access to the following functions:
- Creation and editing of CSV data tables
- Exploratory-descriptive statistics of frequency data (distribution parameters, correlations and data plots)
- Exploratory-classificatory statistics of frequency data (cluster analysis, multidimensional scaling and Calculation of prototypes)
- Qualitative-comparative analysis (Boolean minimization) according to Ragin (1987) and
- the presentation of data from CSV tables in various charts based on a free program for table analysis.

To use these possibilities, as listed at the beginning of the introduction, the free software packages "R" (for the explorative-statistical analyses) and "LibreOffice" for additional display of diagrams must be installed. When using the software for the first time, R must reload statistical packages not present in the standard installation. You will be prompted to select one of the many R servers from a list.

In explorative-statistical analysis, graphical methods of data visualization and (non-)linear data transformations are used in particular to clearly highlight differences, correlations, trends, and the like. Interestingly, the trend in testing complex models (e.g. regression / anova, HLMs / MLMs, ...) also increasingly points to the use of graphical methods instead of basing arguments on coefficients only. Behind the explorative-statistical analysis are mostly data transformations and subgroupings as well as manifold visualizations to be able to order, structure or simply display data from different points of view and according to criteria of interest. For the additional representation of the data in CSV tables, the menu item "Diagrams" is available in "aquad_eda.exe", as mentioned several times.

13.1 Modification of data tables

When determining frequencies, e.g. code or word frequencies and frequency analysis of tables, Aquad 8 saves the results in CSV tables for further use in the built-in analysis modules or external programs. For many analyses, it is useful or necessary to first edit these tables, for example, to delete individual columns (such as the column with the total sum of frequencies in each row of the table), to merge columns, to transform the entire table (swap columns and rows), and so on.

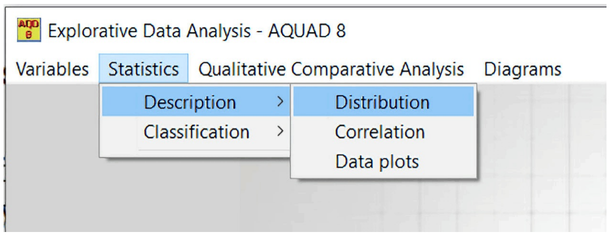


The full range of possibilities of a spreadsheet program is available if you select the "Diagrams" menu item, then open and edit the CSV table you want to work with. For this purpose, you should be somewhat familiar with working with spreadsheet programs (e.g. Microsoft's "Excel", in AQUAD 8 the built-in module "scalc" from the office program "LibreOffice"). To simplify frequent operations, some functions are available separately under "Variables", among others also the possibilities to write a data table yourself or to edit an available one.

13.2 Explorative-descriptive statistics

This module allows the analysis of frequencies of codes in a qualitative text, image, audio or video analysis, of word frequencies in a quantitative text analysis, description of the distribution of frequencies, calculation of correlations and two- or three-dimensional plots (i.e. graphical representations of curves and data) for exploratory graphical data analysis.

13.2.1 Description



(1) Distribution

This R script provides descriptive parameters of the data distribution in a CSV table, namely:
- Number of cases (all)
- Number of missing values (NA)
- Number of observations (without missing values),
- Sum
- Arithmetic mean (MW)
- Geometric mean
- Harmonic Mean
- Variance (VAR)
- Standard deviation (SD)
- Coefficient of Variation (VC)
- Median
- Median absolute deviation (MAD)
- minimum
- maximum
- range
- mean deviation
- Skewness
- Kurtosis (Excess)
- 1st quantile
- 3rd quantile
- Interquartile Range (IQR)
- Standard error of the mean (S.E. MW)

- lower confidence interval mean (LCL)
- upper confidence interval mean (UCL)

The results are displayed within the R environment, additionally as a table and as a text file in the "..\res" subdirectory of Aquad 8. Thus, for example, if the distribution of frequencies of selected metacodes M1 to M8 from the study of Gürtler described in Section 11.5 is to be described, the results can be found in the following files in the results subdirectory: The file to be analyzed is named "M1-M8_DT.csv",
the distribution descriptors appear as a table
- "M1-M8_DT_Out_descriptive-stats.csv" and as text file
- "M1-M8_DT_Out_descriptive-stats.txt".

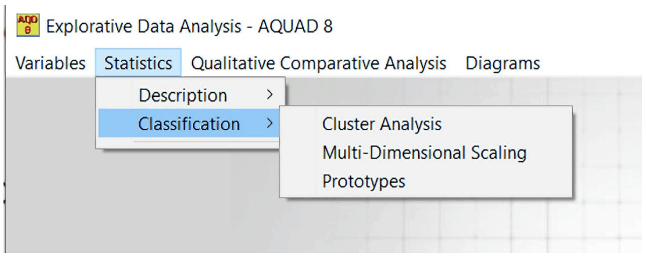
(2) Correlation

The function prompts to open the CSV table to be analyzed (e.g. "M1-M8_DT.csv") and outputs the results, among others, in the file "M1-M8_DT_Out_corrcoeffpvalues.txt" in the form of two tables with the (two-digit) correlation coefficients and their p-values.

(3) Data plots

The distribution of each variable (column) in the output table is represented as boxplots, e.g. for the first variable M1 in the table "M1-M8_DT.csv" in the boxplot file "M1-M8_DT_Out_Boxplot_M1.png". Other graphical representations can be selected under the item "Diagrams" of the main menu.

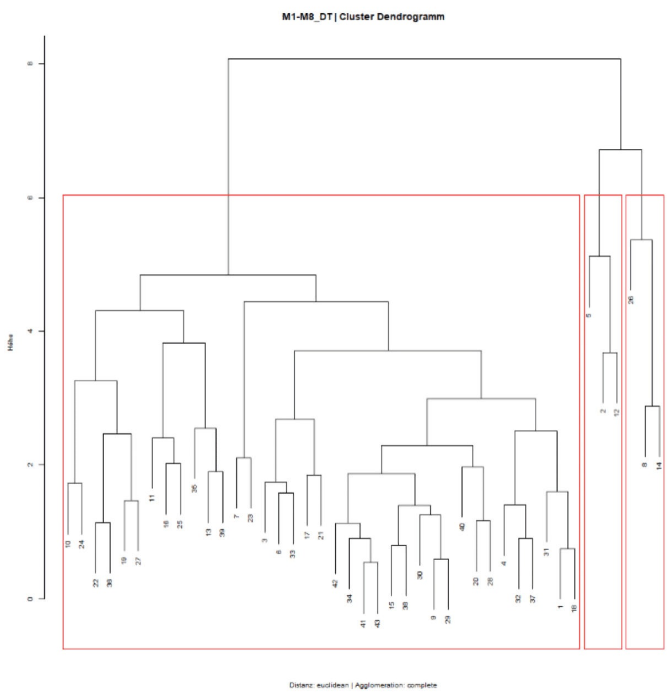
13.2.2 Classification


Explorative Data Analysis - AQUAD 8
Variables
Statistics
Qualitative Comparative Analysis
Diagrams
Description
Classification
Cluster Analysis
Multi-Dimensional Scaling
Prototypes

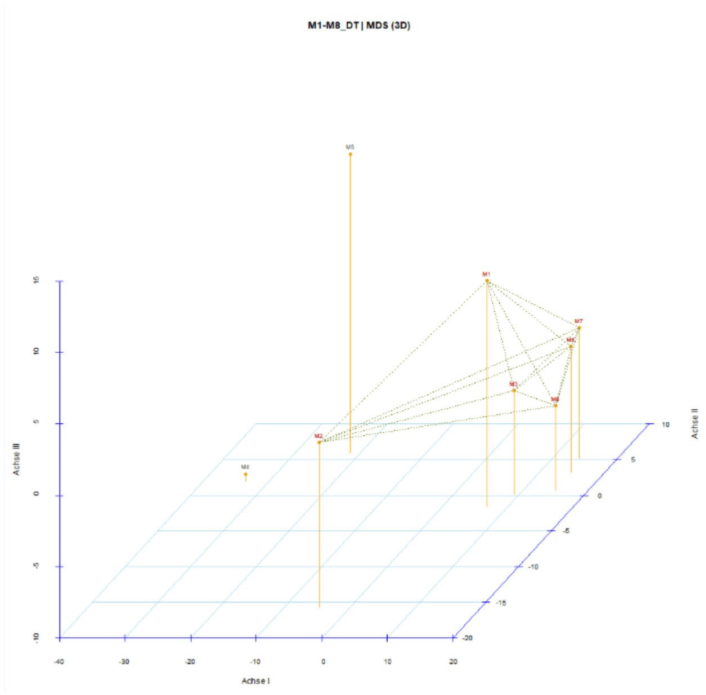
If the corresponding evaluation makes sense, the frequency tables can be analyzed in more detail with the help of R packages for cluster analysis, multidimensional scaling and the determination of prototypes.

To use these scripts, you should learn the basics of statistical procedures, because the software outputs a large number of specific graphical results and text lists.

If you calculate *cluster analyses* of the frequencies in "M1-M8_DT.csv", the dendrogram for the option "euclidian.complete" "M1-M8_DT_Out_hcluster_dendrogram_euclidean.complete.png" appears among others:



A *three-dimensional scaling* of the data of the same CSV table provides, among others, the following graph:



Undoubtedly, multidimensional scaling visualizes the relationships between variables, in this case the frequencies of selected metacodes of a study by Gürtler, and their groupings impressively and can be very informative in terms of the specific research question. But decisively whether this analysis makes sense depends on the research question.

The same is true for the analysis of *prototypes*. In addition to various tables, this R script provides a very clear visualization of the optimal cut through the proximity matrix of the output table. Without understanding the basic concepts and their meaning, you should not use these classification algorithms on a "let's see what comes up" basis. In classical inferential statistics, the term "shotgun approach" would be used for this procedure ("something will be significant").

13.3 Qualitative Comparative Analysis / Boolean Minimization

After the detailed description of the method above in chapter 10, here only a short remark on the sense and purpose of Boolean minimization in the analysis of qualitative data. Especially in the study of social experience and behavior, one should not expect to find a single definite cause for a given phenomenon. Rather, it is important to uncover those constellations of conditions under which a critical phenomenon is typically to be expected. To do this, one must compare many cases in which this phenomenon occurs. Necessarily, this is accompanied by a certain fuzziness, which, however, is not "harmful" or "obstructive", but just corresponds to reality. Ragin (1987) has opened up an explicitly comparative procedure by applying Boolean algebra to qualitative data. The term qualitative comparative analysis (QCA) is often found in the literature. Since the goal of the analysis is the reduction of the initial conditions to minimal condition configurations, so-called implicants, we use the name implicant analysis, which is to be seen as synonymous to QCA.

Thus, the aim is to explain a certain defined set of outcomes with a minimal set of conditional factors. Applied to social phenomena, the approach is used to find the minimum set of conditions for a criterion to be logically TRUE or FALSE across all individual cases examined (these can be persons, texts, but also entire studies). Due to its flexibility, the algorithm is also suitable for meta-analysis.

In addition to its own algorithm for Boolean minimization, which is described in detail above in Chapter 10 and which has been available in AQUAD since version 4, AQUAD makes use of the R package QCA (Qualitative Comparative Analysis) to determine implicants. However, since this package seems to cause problems in newer R versions, we have kept the script in the list of components installed by AQUAD 8, but have not activated it in "aquad_eda.exe", but have put a link to the proprietary algorithms in Aquad. For details and examples of how to use the QCA module, please refer to Chapter 10 above.

13.4 Diagrams

As mentioned at the beginning, the complete range of possibilities of a spreadsheet program is available for the visualization of distributions if you select the menu item "Diagrams", then open and edit the CSV table to be processed. In detail, the "scalc" module of the free "LibreOffice" software called up by AQUAD offers the following display options, either two- or three-dimensional:

- Bar charts
- Bar charts (horizontal)
- Pie charts
- Area diagrams
- Line charts
- Scatter plots (X,Y)
- Bubble charts
- Network diagrams
- Course diagrams
- Columns and lines

Of course, you can select or delete data ranges, change the labels of the axes, change elements of the diagrams and the background of the diagrams in color - just to list some important options.

The prerequisite for using the diagram module is that you have installed the free software "LibreOffice" on your computer.

References

Berelson, B. (1952). *Content analysis in communication research*. Glencoe, Ill.: The Free Press.

Creswell, J. W. (2012). *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*. Thousand Oaks, CA: Sage Publications.

Eisner, E. W. (1981). On the differences between scientific and artistic approaches to qualitative research. *Educational Researcher, 10(4)*, 5-9.

Eisner, E. W. (1983). Anastasia might still be alive, but monarchy is dead. *Educational Researcher, 12(5)*, 13-14/23-24.

Erzberger, C., & Kelle, U. (2003). Making inferences in mixed methods: The rules of integration. In A. Tashakkori & C. Teddlie (Eds.), *Handbook of mixed methods in social and behavioral research* (pp. 457-488). Thousand Oaks: Sage.

Fetterman, D. M. (1982). Ethnography in educational research: The dynamics of diffusion. *Educational Researcher, 11(3)*, 17-22.

Fetterman, D. M. (1988). Qualitative approaches to evaluating education. *Educational Researcher, 17(8)*, 17-24.

Firestone, W. A. (1987). Meaning in method: The rethoric of quantitative and qualitative research. *Educational Researcher, 16(7)*, 16-21.

Fischer, P. M. (1994). Inhaltsanalytische Auswertung von Verbaldaten. In G. L. Huber & H. Mandl (Hrsg.), *Verbale Daten* (2. Aufl.) (S. 179-196). Weinheim: Beltz.

Fox, J. (2002). *An R and S-PLUS companion to applied regression*. Thousand Oaks: Sage .

Fühlau, I. (1978). Untersucht die Inhaltsanalyse eigentlich Inhalte? Inhaltsanalyse und Bedeutung. *Publizistik*, 7-18.

Fühlau, I. (1982). *Die Sprachlosigkeit der Inhaltsanalyse. Linguistische Bemerkungen zu einer sozialwissenschaftlichen Methode.* Tübingen: Gunter Narr.

Galtung, Johan (1990). Theory formation in social research: A plea for pluralism. In E. Øyen (Ed.), *Comparative methodology: Theory and practice in international social research* (S.96--112). London: Sage.

Gläser-Zikuda, M., Seidel, T., Rohlfs, C., Gröschner, A., & Ziegelbauer, S. (Hrsg.) (2012). *Mixed methods in der empirischen Bildungsforschung*. Münster: Waxmann.

Glaser, B. G. (1978). *Theoretical sensitivity*. Mill Valley, CA: Sociology Press.

Glaser, B. G. (1992). *Emergence vs. forcing. Basics of grounded theory analysis*. Mill Valley, CA: Sociology Press.

Glaser, B. G. & Strauss, A. L. (1965). T*he discovery of grounded theory. Strategies for qualitative research*. Chicago: Aldine Publishing Company.

Glaser, B. G., & Strauss, A. L. (1979). Die Entdeckung gegenstandsbezogener Theorie: Eine Grundstrategie qualitativer Sozialforschung. In C. Hopf & E. Weingarten (Hrsg.), *Qualitative Sozialforschung* (S. 91-111). Stuttgart: Klett-Cotta.

Glass, G. V., McGaw, B., & Smith, M. L. (1981). *Meta-analysis in social research*. Beverly Hills: Sage.

Greene, J. C., Caracelli, V. J., & Graham, W. F. (1989). Toward a conceptual framework for mixed-methdos evaluation designs. *Educational Evaluation and Policy Analysis, 11 (3)*, 255-274.

Günther, U. L. (1987). Sprachstil, Denkstil und Problemlöseverhalten. Inhaltsanalytische Untersuchungen über Dogmatismus und Abstraktheit. In P. Vorderer & N. Groeben (Hrsg.), *Textanalyse als Kognitionskritik?* (S. 22-45). Tübingen: Narr.

Gürtler, L. & Huber, G. L. (2012). Triangulation. Vergleiche und Schlussfolgerungen auf der Ebene der Datenanalyse. In M. Gläser-Zikuda, T. Seidel, C. Rohlf, A. Gröschner & S. Ziegelbauer (Hrsg.), *Mixed methods in der empirischen Bildungsforschung* (pp.37-50). Münster: Waxmann.

Gürtler,L., Studer, U. M., & Scholz, G. (2010): *Tiefensystemik, Bd. 1. Lebenspraxis und Theorie: Wege aus Süchtigkeit finden.* Münster: Verlag Monsenstein und Vannerdat, MV-Wissenschaft.

Härtling, P. (o.J.): *Das war der Hirbel.* Stuttgart: Klett-Cotta.

Handl, A. (2002). *Multivariate Analysemethoden. Theorie und Praxis multivariater Verfahren unter besonderer Berücksichtigung von S-PLUS.* Berlin: Springer.

Held, J. (1994). *Praxisorientierte Jugendforschung. Theoretische Grundlagen, methodische Ansätze, exemplarische Projekte.* Hamburg: Argument.

Held, J., Horn, H.-W. & Marvakis, A. (1996). *Gespaltene Jugend. Politische Orientierungen jugendlicher ArbeitnehmerInnen.* Opladen: Leske+Budrich.

Held, J., & Marvakis. A. (1992). Empirische Jugendforschung und ihr Verhältnis zur politischen Bildung. In K.-H. Braun & K. Wetzel (Hrsg.). *Lernwidersprüche und pädagogisches Handeln.* Bericht von der 6. internationalen Ferienuniversität Kritische Psychologie, 24. bis 29. Februar 1992 in Wien (S. 243-256). Marburg: Verlag Arbeit & Gesellschaft.

Hildenbrand, B. (2006): *Einführung in die Genogrammarbeit* [Introduction to the work with genograms]. Heidelberg: Carl Auer.

Howe, K. R. (1985). Two dogmas of educational research. *Educational Researcher, 14 (8)*, 10-18.

Howe, K. R. (1988). Against the quantitative-qualitative incompability thesis or dogmas die hard. *Educational Researcher, 17 (8)*, 10-16.

Huber, A. A. (2007). How to add qualitative profundity to quantitative findings in a study on cooperative learning. In Ph. Mayring, G.L. Huber, L. Gürtler, & M. Kiegelmann (Eds.) *Mixed methodolgoy in psychological research* (pp. 179-190). Rotterdam: Sense Publishers.

Huber, G. L. (Ed.) (1992). *Qualitative Analyse. Computereinsatz in der Sozialforschung.* München: Oldenbourg.

Huber, G. L. (1992). Qualitative Analyse mit Computerunterstützung. In G. L. Huber (Hrsg.), *Qualitative Analyse. Computereinsatz in der Sozialforschung* (S. 115-176). München: Oldenbourg.

Huber, G. L. (2001). *Typenbildung in der qualitativen Analyse am Beispiel der Lehrerforschung.* Beitrag zur Tagung der Fachgruppe Pädagogische Psychologie der Deutschen Gesellschaft für Psychologie, Landau 2001.

Huber, G. L. & Kenntner, S. (1988). *Struktur biographischer Selbst-Schemata.* Bericht an die DFG. Tübingen: Arbeitsbereich Pädagogische Psychologie am Institut für Erziehungswissenschaft I der Universität Tübingen.

Huber, G. L., & Marcelo García, C. (1992). Voices of beginning teachers: Computer-assisted listening to their common experiences. In M. Schratz (Ed.), *Qualitative voices in educational research.* (S. 139-156). London: Falmer.

Huber, G. L., & Roth, J. W. H. (2004). *Research and intervention by interviews and learning diaries in inservice teacher training.* Paper presented at the workshop on "Mixed Methods in Psychological Research" (organized by SIG #17, EARLI and the Center for Qualitative Psycholgy, University of Tübingen) at Freudenstadt, Germany, Oct. 21-24, 2004.

Huberman, M. (1987). How well does educational research really travel? *Educational Researcher, 16 (1)*, 5-13.

Jackson, G. B. (1980). Methods for integrative reviews. *Review of Educational Research, 50*, 438-460.

Jacob, E. (1988). Clarifying qualitative research: A focus on traditions. *Educational Researcher, 17 (1)*, 16-24.
Jordell, K. (1987). Structural and personal influence in the socialization of beginning teachers. *Teaching and Teacher Education, 3*, 165-177.

Kelle, U. (Ed.) (1995). *Computer-aided qualitative data analysis. Theory, methods, and practice.* London: Sage.

Kiegelmann, M., Held, J., Huber, G. L. & Ertel, I. (2000). Das Zentrum für Qualitative Psychologie an der Universität Tübingen [24 Absätze]. *Forum Qualitative Sozialforschung; Forum: Qualitative Social Research [On-line Journal], 1/(2).* Verfügbar über: http://www.qualitative-research.net/fqs-texte/2-00/2-00kiegelmannetal-d.htm

Kracauer, S. (1952). The challenge of qualitative content analysis. *Public Opinion Quarterly, 16*, 631- 642.
Lisch, R. & Kriz, J. (1978). *Grundlagen und Modelle der Inhaltsanalyse.* Reinbek bei Hamburg: Rowohlt.

Marcelo García, C. (1991). *El primer año de enseñanza.* Sevilla: Grupo de Investigación Didáctica de la Universidad de Sevilla.

Marcelo García, C. (1992). *Desarrollo profesional e iniciación a la enseñanza. Estudio de caso de un programa de formación para profesores principantes.* Resolución de 30 de sept. de 1992, de la Universidad de Sevilla.

Mathison, S. (1988). Why triangulate? *Educational Researcher, 17 (2)*, 13-17.

Mayring, Ph. (1988). *Qualitative Inhaltsanalyse.* Weinheim: Deutscher Studien Verlag.

Mayring, Ph. (2001, Februar). Kombination und Integration qualitativer und quantitativer Analyse [31 Absätze]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* [On-line Journal], *2*(1). Verfügbar über: http://qualitativeresearch.net/fqs/fqs.htm

Mayring, Ph. (2007). Mixing qualitative and quantitative methods. In Ph. Mayring, G.L. Huber, L. Gürtler, & M. Kiegelmann (Eds.) *Mixed methodolgoy in psychological research* (pp. 27-36). Rotterdam: Sense Publishers.

Mayring, Ph., Huber, G.L., Gürtler, L., & Kiegelmann, M. (Hrsg.) (2007). *Mixed methodolgoy in psychological research.* Rotterdam: Sense Publishers.

Maxwell, J. (2002). Realism and the roles of the researcher in qualitative psychology. In M. Kiegelmann (Ed.), *The role of the researcher in qualitative psychology* (pp. 11-30). Tübingen: Ingeborg Huber Verlag.

Medina, A., Feliz, T., Domínguez, M.-C., & Pérez, R. (2002). The methodological complementariness of biograms, in-depth interviews, and discussion groups. In M. Kiegelmann (Ed.), *The role of the researcher in qualitative psychology* (pp. 169-184). Tübingen: Ingeborg Huber Verlag.

Miles, M. B. (1985). Qualitative data as an attractive nuisance: The problem of analysis. In J. Van Maanen (Ed.), *Qualitative methodology*. (5. Aufl.). (S. 117-134). Beverly Hills: Sage.

Miles, M.B. & Huberman, A.M. (1984). *Qualitative data analysis: A sourcebook of new methods.* Beverly Hills, CA: Sage.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis. An expanded sourcebook.* (2nd edition). Thousand Oaks, CA: Sage.

Oldenbürger, H. (1981). *Methodenheuristische Überlegungen und Untersuchungen zur "Erhebung" und Repräsentation kognitiver Strukturen.* Göttingen: Dissertation.

Oldenbürger, H. (2004). *Proxim: Repräsentation von Proximitymatrizen - Beschreibung und Analyse.* http://www.liteline.de/~holdenb/fst/nwz/R-PHP/Proxim.R

Oevermann, U. (1996). *Konzeptualisierung von Anwendungsmöglichkeiten und praktischen Arbeitsfeldern der objektiven Hermeneutik.* Frankfurt am Main: Goethe-Universität.

Oevermann, U. (2002). *Klinische Soziologie auf der Basis der Methodologie der objektiven Hermeneutik.* Frankfurt am Main: Goethe-Universität. Abrufbar über: http://publikationen.ub.uni-frankfurt.de/frontdoor/index/index/docId/4958 [Zugriff am 22.01.2012].

Oevermann, U., Allert, T., Konau, E., & Krambeck, J. (1979). Die Methodologie einer "objektiven Hermeneutik" und ihre allgemeine forschungslogische Bedeutung in den Sozialwissenschaften (pp. 352-434). In H.-G. Soeffner (Ed.), *Interpretative Verfahren in den Sozial- und Textwisssenschaften.* Stuttgart: Metzler.

Phillips, D. (1983). After the wake: Post-positivistic educational thought. *Educational Researcher, 12(5)*, 4-12.

Popko, E. S. (1980). *Key-word-in-context bibliographic indexing. Release 4.0 users' manual.* Cambridge, Mass.: Harvard University, Laboratory for Computer Graphics and Spatial Analysis.

Popper, K. (1934/2005). *Logik der Forschung* (11. Aufl.).Tübingen: Mohr Siebeck.

R - Development Core Team (2004). *R: A language and environment for statistical computing.* Vienna: Austria. http://www.r-project.org

Ragin, C. C. (1987). *The comparative method. Moving beyond qualitative and quantitative strategies.* Berkeley: University of California Press.

Reichertz, J. (2000). Zur Gültigkeit von Qualitativer Sozialforschung [76 Absätze]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research [On-line Journal], 1*(2). Abrufbar über: http://qualitative-research.net/fqs/fqs-d/2-00inhalt-d.htm [Zugriff: 22.01.2012].

Saldaña, J. (2016). *The coding manual for qualitative researchers.* London: Sage Publications.

Schratz, M. (Ed.) (1993). *Qualitative voices in educational research.* London: Falmer Press.

Schweizer, H. (2004). Preparations for the Redemption of the World: Distribution of Words and Modalities in Chapter I of Don Quixote. In M. Kiegelmann & L. Gürtler (Eds.), *Research Questions and Matching Methods of Analysis* (pp.71-108). Tübingen: Ingeborg Huber Verlag.

Shelly, A. L. (l986). The stages of qualitative data analysis. In A. L. Shelly, R. A. Archambault, C. Sutton & P. Tinto (Eds.), *The stages of qualitative research: Researcher thinking facilitated by logic programming.* CASE Technical Report No. 8607, Syracuse, NY: Syracuse University, The Center for Computer Applications and Software Engineering.

Shelly, A. L., & Sibert, E. E. (1985). *The QUALOG user's manual.* Syracuse, NY: Syracuse Univer sity, School of Computer and Information Science.

Shelly, A. L., & Sibert, E. E. (1992). Qualitative Analyse: Ein computerunterstützter zyklischer Prozeß. In G. L. Huber (Hrsg.), *Qualitative Analyse. Computereinsatz in der Sozialforschung* (S. 71- 114). München: Oldenbourg.

Shulman, L. S. (1981). Disciplines of inquiry in education: An overview. *Educational Researcher, 10*, 5-12.
Smith, J. K. (1983). Quantitative versus qualitative research: An attempt to clarify the issue. *Educational Researcher, 12(3)*, 6-13.

Smith, J. K. & Heshusius, L. (1986). Closing down the conversation: The end of the quantitative-qualitative debate among educational researchers. *Educational Researcher, 15(1)*, 4-12.

Strauss, A., & Corbin, J. (1990). *Basics of qualitative research. Grounded theory procedures and techniques.* Newbury Park, CA: Sage.

Studer, U. M. (1988). *Verlangen, Süchtigkeit und Tiefensystemik.* Fallstudie des Suchttherapiezentrums für Drogen-abhängige
START AGAIN in Zürich zwischen 1992–1998. Bericht ans Bundesamt für Justiz.

Tashakkori, A., & Teddlie, C. (1998). *Mixed methodology: Combining qualitative and quantitative approaches* (Applied Social Research Methods, No. 46). Thousand Oaks: Sage.

Tashakkori, A., & Teddlie, C. (Eds.) (2003). *Handbook of mixed methods in social and behavioral research.* Thousand Oaks: Sage.

Tesch, R. (1990). *Qualitative research. Analysis types and software tools.* New York: Falmer.

Tesch, R. (1992). Verfahren der computerunterstützten qualitativen Analyse. In G. L. Huber (Hrsg.), *Qualitative Analyse. Computereinsatz in der Sozialforschung* (S. 43-70). München: Oldenbourg.

Thomas, W. I., & Znanecki, F. (1918). *The Polish peasant in Europe and America.* Boston: Badger.

Tuthill, D. & Ashton, P. (1983). Improving educational research through the development of educational paradigms. *Educational Researcher, 12(10)*, 6-14.

Van der Linden, J., Erkens, G., & Nieuwenhuysen, T. (1995). Gemeinsames Problemlösen in Gruppen. *Unterrichts-wissenschaft*, 23, 301-315.

Villar, L. M. (Ed.) (1981). *Las prácticas de enseñanza.* Sevilla: ICE de la Universidad.

Villar, L. M., & Marcelo, C. (1992). Kombination qualitativer und quantitativer Methoden. In G. L. Huber (Hrsg.), *Qualitative Analyse. Computereinsatz in der Sozialforschung* (S. 177-218). München: Oldenbourg.

Weber, M. (1988). *Gesammelte Aufsätze zur Wissenschaftslehre*. Tübingen: Mohr.

Weber, R. P. (1985). *Basic content analysis*. Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-049. Beverly Hills: Sage.

Wernet, A. (2009). *Einführung in die Interpretationstechnik der Objektiven Hermeneutik* (3rd ed.). Wiesbaden: VS Verlag für Sozialwissenschaften.

Yin, R. K. (1989). *Case study research. Design and methods*. Applied Social Research Methods Series, Vol. 5. Newbury Park, CA: Sage.

Zabalza, M. A. (1991). *Los diarios de clase. Documento para estudiar cualitativamente los dilemas prácticos de los profesores*. Barcelona: Promociones y Publicaciones Universitarias, S.A.

About the authors

Prof. i.R. Dr. phil. Dr. h.c. Günter L. Huber (emeritus, former head of Educational Psychology at the Institute of Educational Science, University of Tübingen). His teaching activities, research, and publications include the topics of cooperative learning, problems of teaching and learning, interindividual differences, and analysis of qualitative data. Developer and founder of AQUAD.
Contact: info@aquad.de

Dr. rer. soc. Leo Gürtler (Dipl.-Psych., promov. Erz.wiss.) - many years of activity as a researcher (e.g. humor, addiction, empirical educational research) and lecturer. Work abroad in free economy (health care) as change manager. He works as a systemic coach and scientific consultant.
Contact: info@guertler-consulting.de