

# QualCoder



QualCoder is free software for qualitative data analysis

## Copyright Notice

The MIT License (MIT)

Copyright (c) 2019 Colin Curtain

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1. Edition March 2019

Author: Colin Curtain

ccbogel@hotmail.com

# Table of Contents

1	What is QualCoder.....	4
1.1	Why use QualCoder?.....	4
2	Downloading files and dependencies.....	5
2.1	Preparatory Downloads.....	5
2.1.1	Microsoft Windows.....	5
2.1.2	Linux.....	5
3	Starting QualCoder.....	7
3.1	Linux.....	7
3.2	Windows.....	7
3.3	Backups.....	7
4	How to use.....	8
4.1	Create a new project.....	8
4.2	Settings.....	8
4.3	Files and Cases Menu.....	9
4.3.1	Manage Files.....	9
4.3.2	Manage Cases.....	10
4.3.3	Import survey.....	13
4.3.4	Attributes menu.....	14
4.3.5	Coding text.....	15
4.3.6	Categories and codes.....	16
4.3.7	Coding images.....	16
4.3.8	Coding audio and video.....	17
4.3.9	Codebook.....	18
4.3.10	Node graph.....	18
4.4	Reports.....	19
4.4.1	Coding Reports.....	19
4.4.2	Coding Comparison.....	21
4.4.3	SQL Statements Dialog.....	21
4.5	Multiple dialog windows.....	22
5	Other details about the software.....	24
6	Future plans.....	25
7	About the author.....	26

# 1 What is QualCoder

---

QualCoder is free, open source software for qualitative data analysis.

With QualCoder you can code text and images, write journal notes and memos. You can categorise codes into a tree-like hierarchical categorisation scheme. Coding for audio and video is under development and requires the VLC media player.

Reports can be generated for text coding and for coder comparison using the Cohen's Kappa statistic. A graph displaying codes and categories can be generated to visualise the coding hierarchy. Most reports can be exported at html, open document text (ODT) or as plain text files.

QualCoder is written in python 3 using Qt for the graphical interface. A Sqlite database is used to store the coding data.

QualCoder is available from <https://github.com/ccbogel/QualCoder>. There is also a wordpress site for QualCoder at <https://qualcoder.wordpress.com/>.

## 1.1 Why use QualCoder?

Firstly, QualCoder is free. You do not need to pay for a commercial software license. QualCoder is easy to use and contains the core functionalities required for qualitative analysis. You use Linux. QualCoder has been developed on Linux and also works on Windows. There are many Free/Libre Open Source Software (FLOSS) supporters who are willing to use and support FLOSS development.

## 2 Downloading files and dependencies

---

Download the latest QualCoder from GitHub: <https://github.com/ccbogel/QualCoder>. Unzip the folder. The main program code is in the *QualCoder* folder. The Examples folder contains some example files which can be loaded into a test QualCoder project. There are an *install.sh* and a *QualCoder.desktop* file which are used to create a Linux Launcher.

### 2.1 Preparatory Downloads

#### 2.1.1 Microsoft Windows

you will need to download and install [Python 3](#) and [PyQt5](#) before using QualCoder. You will also need the [lxml](#) module to import from docx files. The [Pillow](#) module is also needed. A Python 3 release can be obtained here: <https://www.python.org/downloads/windows/>. Finally, you need to have the VLC media player installed. You can get this from the Windows Store or from their website: <https://www.videolan.org/vlc/download-windows.html>

Once a Python 3 release is installed you should be able to install the modules [PyQT5](#), [lxml](#) and [Pillow](#) using the following commands in the Windows Command prompt (The prompt is under Windows System):

```
python -m pip install pyqt5
python -m pip install lxml
python -m pip install Pillow
```

#### 2.1.2 Linux

I have created a *QualCoder.deb* package for easy install on Debian/Ubuntu systems.

If you want to install manually follow these instructions:

This method will create a Launcher. Go to your downloads folder. Make the *install.sh* script executable. The example script below assumes you are in your home directory. Open a terminal and type the following to install QualCoder:

```
cd Downloads/QualCoder
```

Run the *install.sh* file. The install process will ask for your permission to install QualCoder into the */usr/share* directory. It will also download the additional python modules.

```
./install.sh
```

The *install.sh* script will also install the `python3-pyqt5`, `python3-lxml` and `python3-pillow` modules as well as the VLC media player. If you do not use the install script, you need to run these commands to install the modules:

```
sudo apt install python3-pyqt5
```

```
sudo apt install python3-lxml
sudo apt install python3-pil
sudo apt install vlc qtwayland5
sudo apt install qpdf
```

## 3 Starting QualCoder

---

### 3.1 Linux

If you have successfully run the *install.sh* script there will be a program launcher. This is shown in the image below.

Alternatively, using the terminal, move to the inner QualCoder folder and open the `qualcoder.py` file:

```
python3 qualcoder.py
```

Soon I will create a flatpak installation for easier installation on Linux.

### 3.2 Windows

In Windows create a shortcut to the `qualcoder.py` file and double click to open. Alternatively, using the command prompt, move to the directory and open the `qualcoder.py` file:

```
python3 qualcoder.py
```

### 3.3 Backups

QualCoder does not automatically create backups. So I recommend that you make a backup copy of the `qda` folder each time you plan to work on a `qda` project. Also make a backup before doing any substantial changes, such as reorganising codes and categories.

## 4 How to use

---

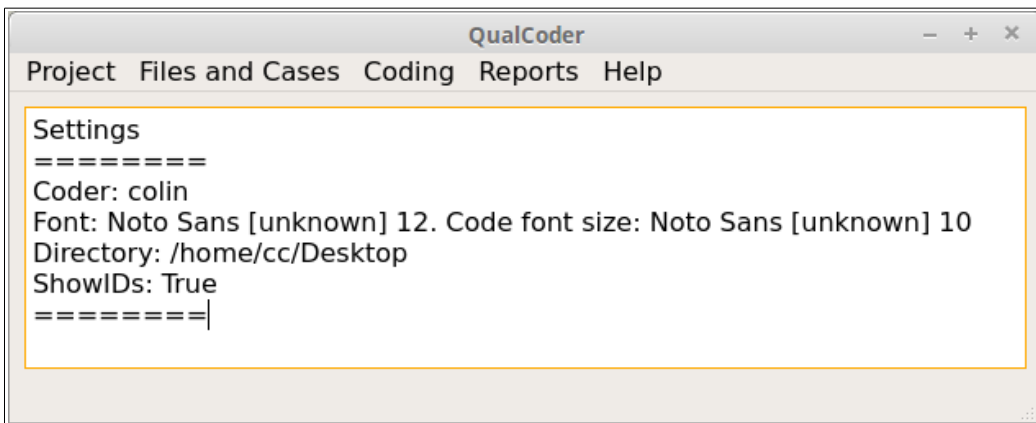
There are a few sample files in the Examples folder. These can be used to test importing files of different document formats and of importing an image. There are also example files to test importing case attributes, and for importing a survey.

### 4.1 Create a new project

Create a new project under the 'Project' menu, call it test. The project will be saved as *test.qda*. Test.qda is actually a folder containing subfolders which hold the database and other files. Additionally, in your home directory QualCoder will create a QualCoder.log file for logging events, including hopefully all software errors and exceptions and a QualCoder\_settings.txt file which will contain the current coder's name, preferred fonts and a preferred working directory.

Under the *Project* menu click on *Project memo*. This is a memo about your project. Type "A test QDA project" and click the OK button.

You will notice the main window displays the following, with a menu bar at the top:

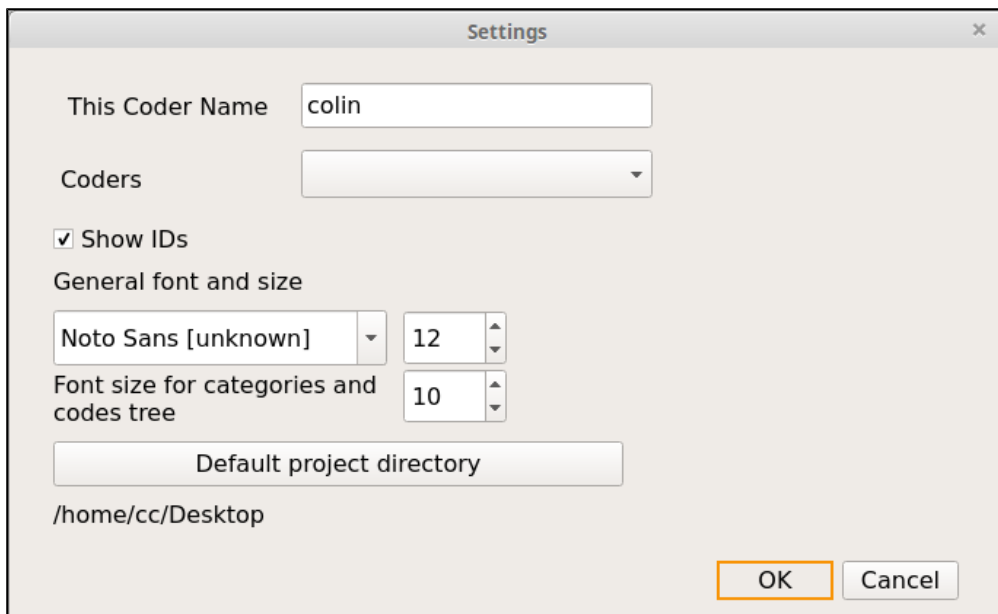


The main window displays various changes that are made when using QualCoder.

### 4.2 Settings

The settings dialog allows you to change the coder name. Multiple coders can code the same text. Checking the Show IDs box displays ID columns for data stored in various tables.





## 4.3 Files and Cases Menu

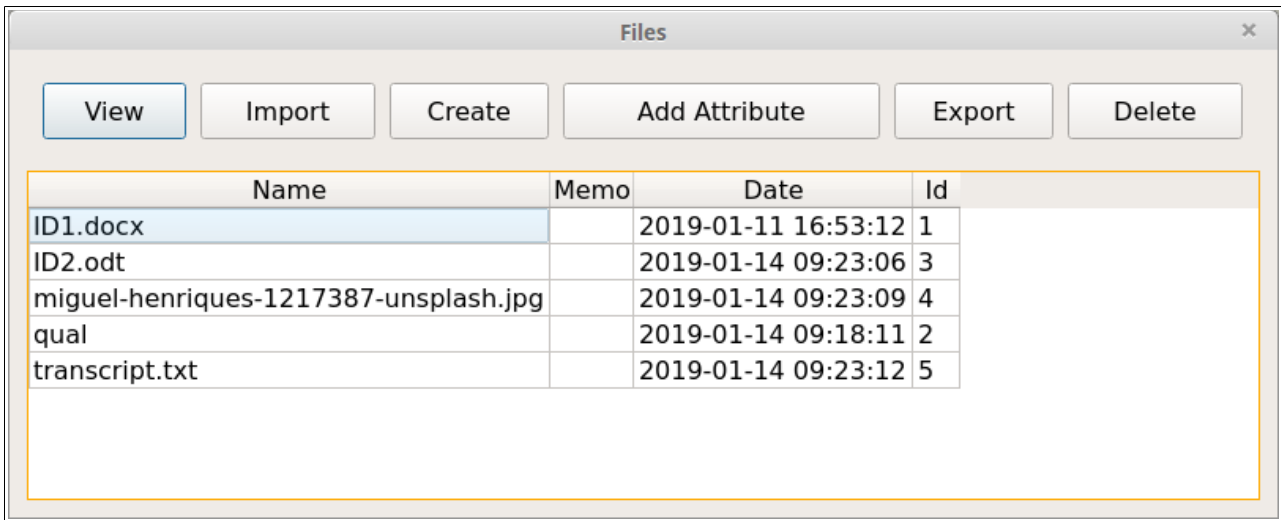
### 4.3.1 Manage Files

You will usually want to load text files into QualCoder prior to coding. You can import text from plain text documents, docx, odt, epub and pdf documents. Pdf importing can sometimes be problematic. Another option is to manually enter text from within QualCoder. You can also load image files in the following formats: jpg, jpeg, png and gif.

Video (mov, mp4, wmv format) and audio files (wav, mp3 format) can be imported. Loading an audio or video file will also automatically create a blank text transcript file. This file will have the same name as the audio or video file, but have a '.transcribed' suffix. Initially, this text transcription will be empty. You can transcribe the file your self or get the file professionally transcribed and copy and paste the text into this file.

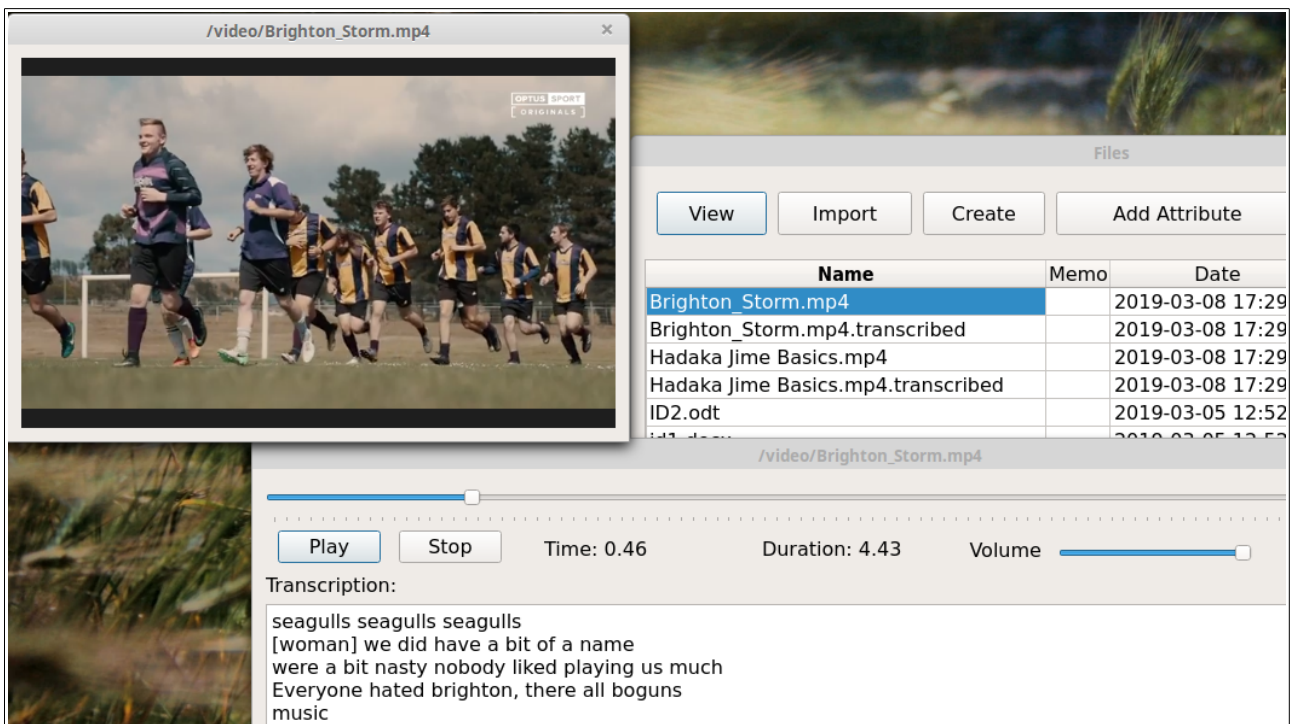
Filenames should only contain typical filename characters, e.g. `a-z 0-9 _ - space ()`. Otherwise, when exporting a text file, unusual characters in the file name will cause an error. Exporting is only available for text files, if you need to copy an image or other media, open the relevant folder inside the .qda folder to copy the media.

Attributes are variables that can be used to describe or classify the files. These can be added here or through the Manage Attributes menu option.



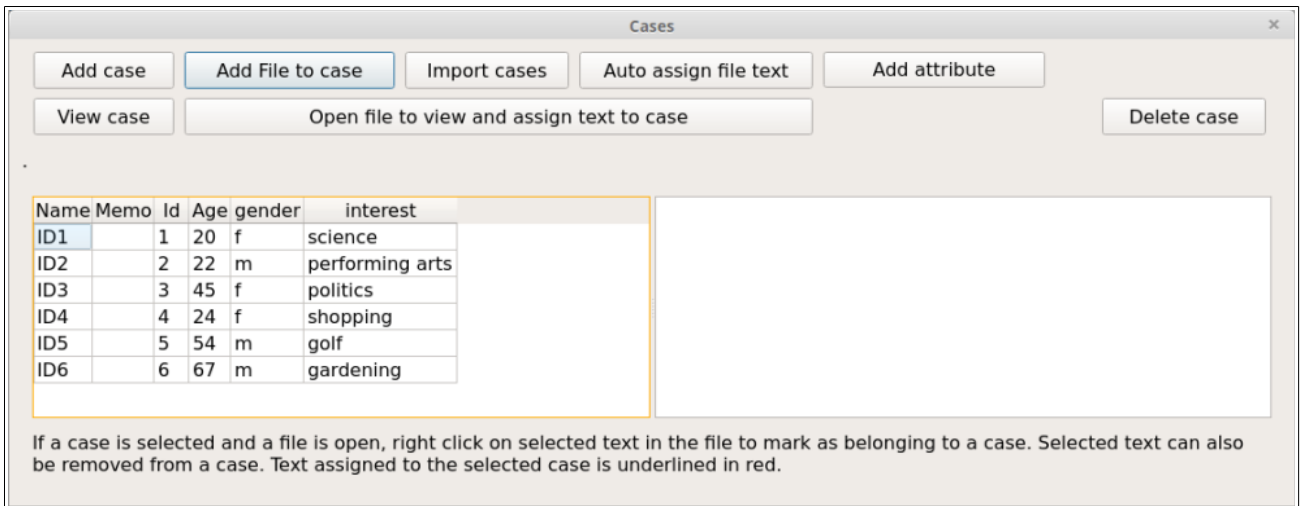
As a practical example: Open the *Manage Files* dialog. In the Examples folder import the following files: ID1.docx, ID2.odt, transcript.txt and the miguel-henriques.jpg.

You can open an audio or video file to view. This opens two windows, one for viewing the video and one for the controls and a transcript, shown below. The transcript text file must have the same name as the video file, but have a '.transcribed' suffix.



### 4.3.2 Manage Cases

Open the *Manage Cases* dialog, shown here:



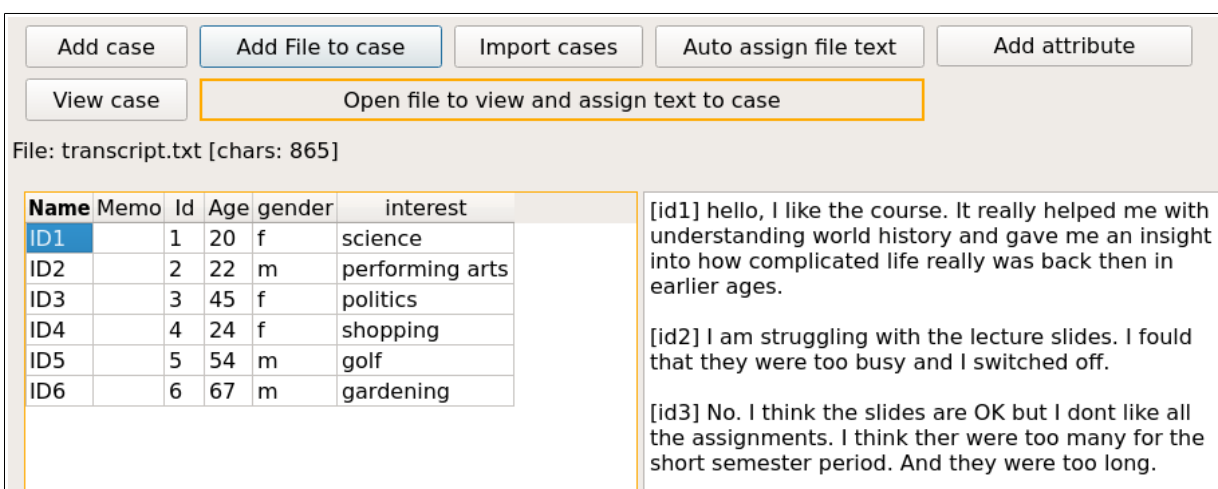
Cases are useful for seeing text and images linked to particular cases and for assigning attributes such as age and gender to interview participants. You can add a memo of notes to each case and rename a case by double-clicking on a case name.

Practical example: Add the three students (or cases) here by clicking the *Add case* button. Call each student: ID1, ID2 and ID3.

Now, add each student's file to each student. Click on a case, say ID1, then click the *Add file to case* button. Select the ID1.docx file. Add the ID2.odt file to ID2 and add the miguel-henriques.jpg to ID3.

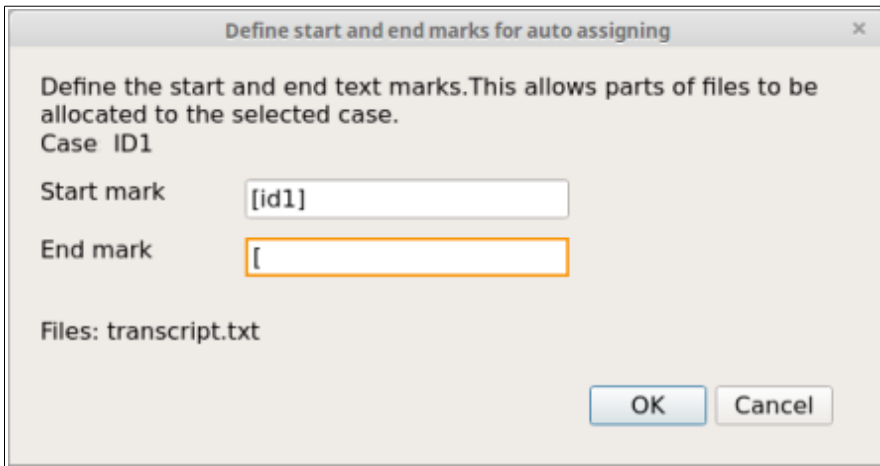
Use the *Open file to view and assign text to case* button and open file ID1.docx and select case ID1. You will see the text will be underlined in red which indicates this text is associated with this case.

Now open the transcript.txt file. Notice the transcript begins with a student id inside square brackets [ ].

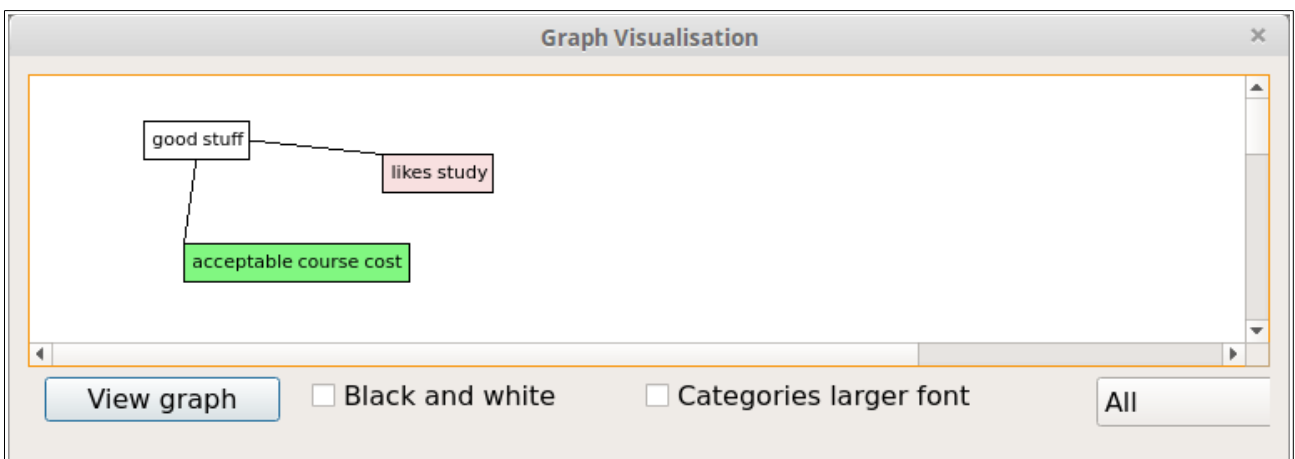


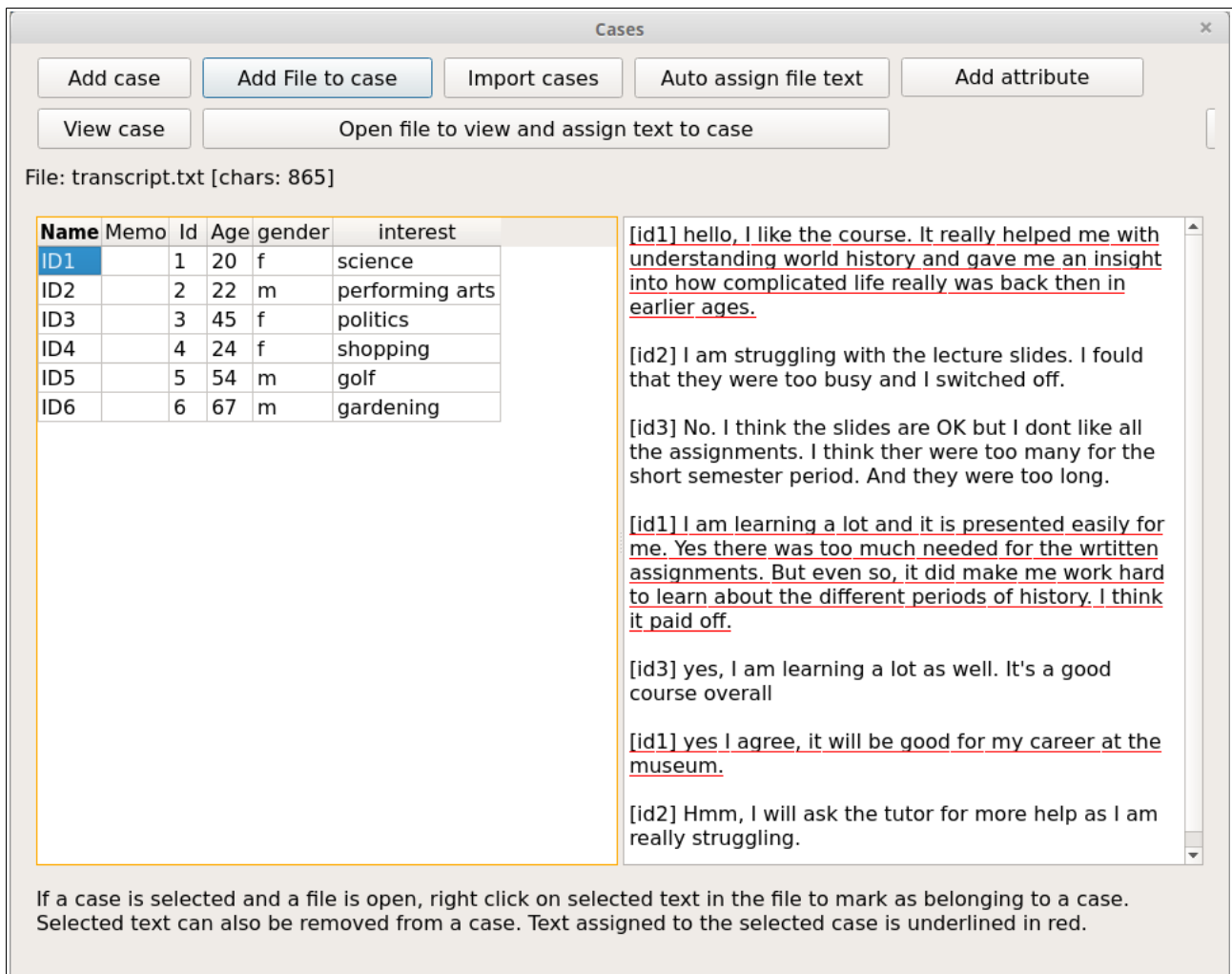
Try automatically assigning text to ID1 by selecting case ID1 then clicking the *Auto assign file text* button. You will be asked which file or files to assign the case to. Select *transcript.txt* from the list.

Next you need to enter the start and end marks. The start mark will be '[id1]' and the end mark will be '[', note this is case sensitive so that is why you use the lower case here.



Now select the case ID1 and see the assigned text. clicking in a code gives a menu of options such as displaying the memo, or displaying all the case or file text that has been coded with the selected code.





You can also select a case, manually highlight text, and right-click to mark – or assign -the text to the selected case.

Importing attributes for the cases. You can import attributes from a CSV file. The first row must contain the attribute headings. The first column must contain matching case names (for this example: ID1, ID2, ID3). Note that this is case sensitive. Open the attributes.csv file in the Examples folder to see how it is laid out.

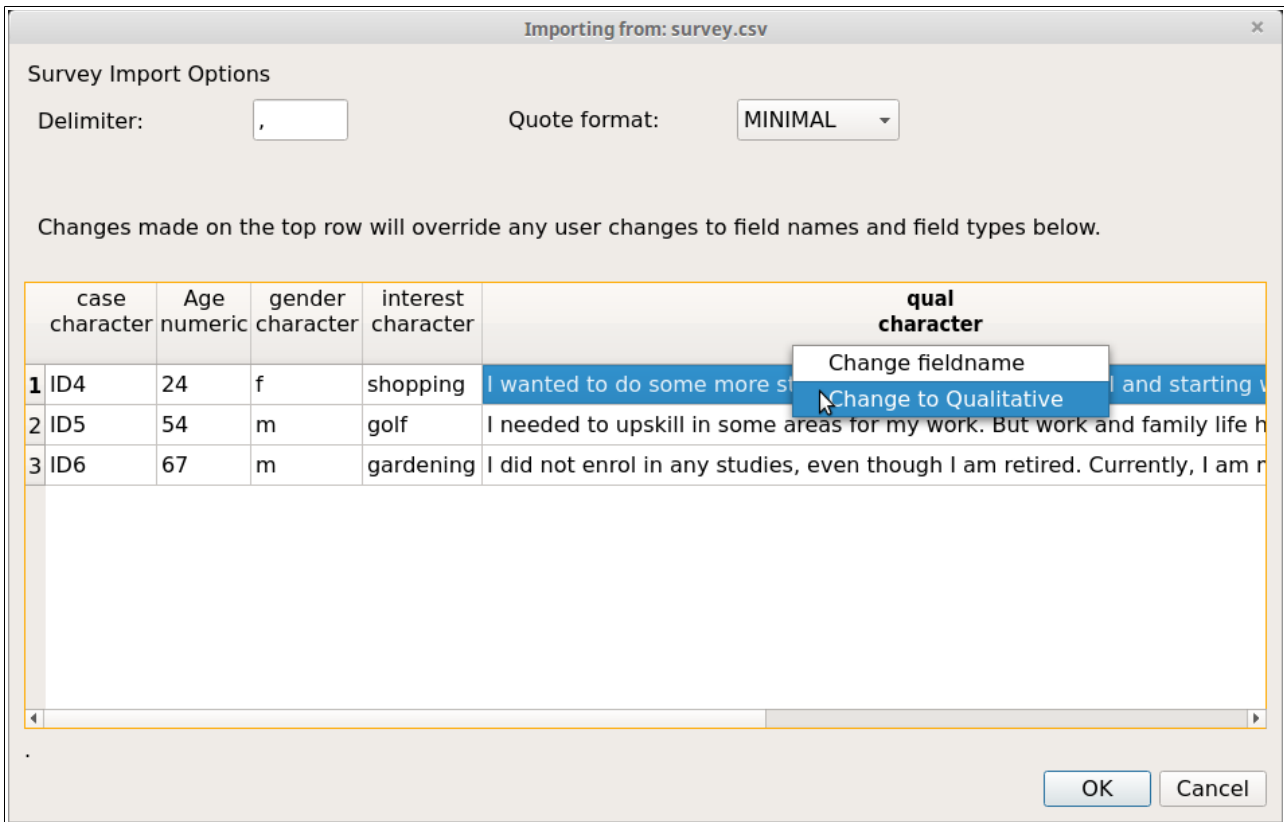
For this example – delete all the cases shown. Then, import the attributes from the attributes.csv file through the *Import Attributes* dialog. You will again need to link the files and file text to each case.

### 4.3.3 Import survey

This asks for a CSV file to be imported. You can try importing the survey.csv file in the Examples folder.

Survey files must be in a CSV format, that is comma delimited format, or another delimiter (tabs are commonly used too). For tab-delimited files type *ta*, *tb*, or *tab* in the Delimiter box so that QualCoder knows the csv file is tab delimited. The first row must contain the headings for attributes. The first column must contain the unique identifiers for each survey respondent.

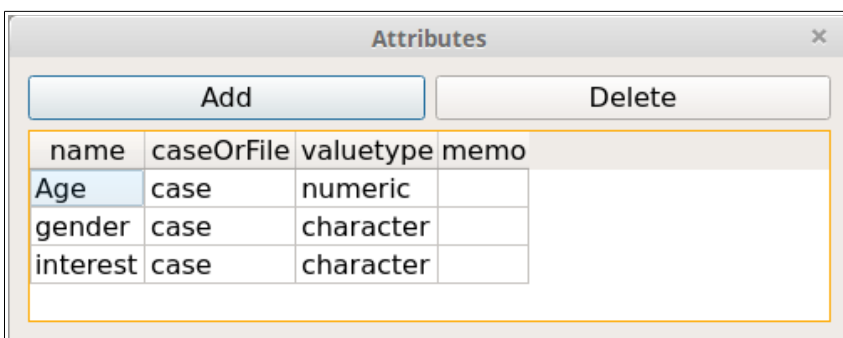
QualCoder will determine if the other columns (attributes) are Numeric or Character. QualCoder cannot determine if an attribute is qualitative data. You must right-click and change the field type from *character* to *qualitative* for those fields that need to be qualitative.



When you click on the OK button, the cases and their attributes will be added. The qualitative column will be converted to a file which will be named with the column name plus the current date and time. Each respondent's row will be prepended with [the unique id] so that you can identify each respondent. Also, the text for each respondent will be automatically linked to the corresponding case.

#### 4.3.4 Attributes menu

Attributes are variables associated with files or cases. They can be useful to add context to the text analysis. Open the *Manage Attributes* dialog. You can add, delete, rename and add memo notes to attributes.

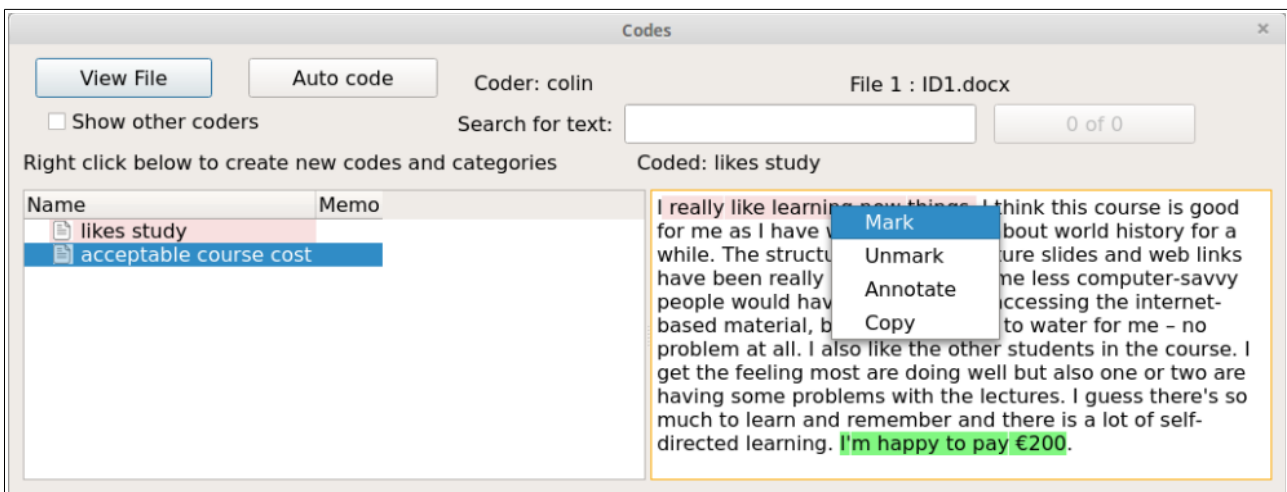


It is not the intention for QualCoder to perform statistical analyses of the attributes. It would be preferable to use dedicated statistical software such as R (<https://www.r-project.org/>) or other such software.

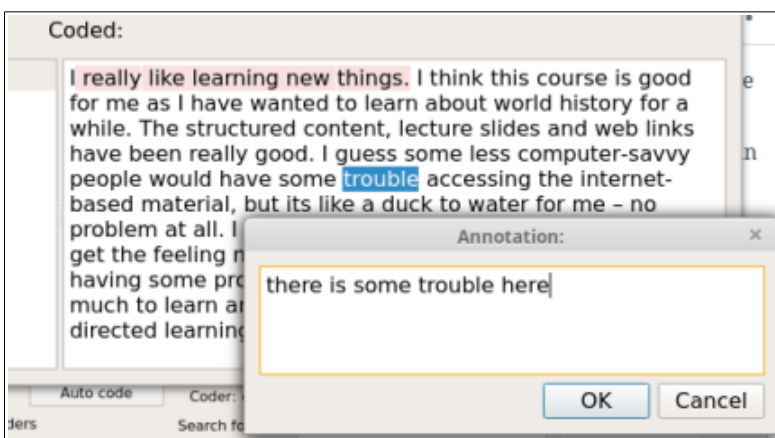
### 4.3.5 Coding text

Select *Code text* from the *Coding* menu. This is the central dialog for assigning codes to text. Once text segments are coded, hovering the mouse over the coding shows the code name as a tooltip. Clicking on the coded segment also shows the code name. Press the *View File* button to select a file to open for coding. Create a new code by right-clicking in the left hand window. Codes can also be assigned a colour by right-clicking on the code and selecting the *change code colour* option. Other options from the right-click menu include adding a memo to the code, deleting the code, renaming, adding a new code, and adding a new category.

Select a code, then select some text. Right-click and mark the text to assign it to the selected code. Hover the mouse pointer over coded text to see a tooltip of the code. Coded text can be uncoded by clicking on the text segment and pressing the *Unmark* button.



Add an annotation (like a memo for a text segment) to a text selection. The text will become **bold** to mark the position of the annotation.

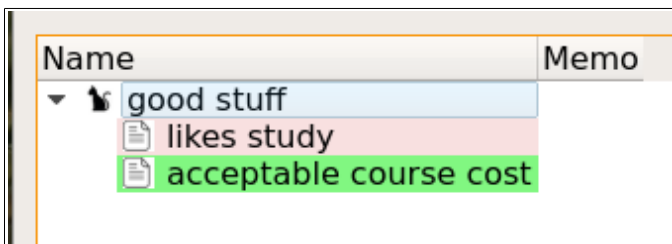


You can also automatically code text segments using the *Auto Code* button. Enter the text you want to autocode. A dialog asks for one or more files to autocode. All matching text will be assigned the selected code.

### 4.3.6 Categories and codes

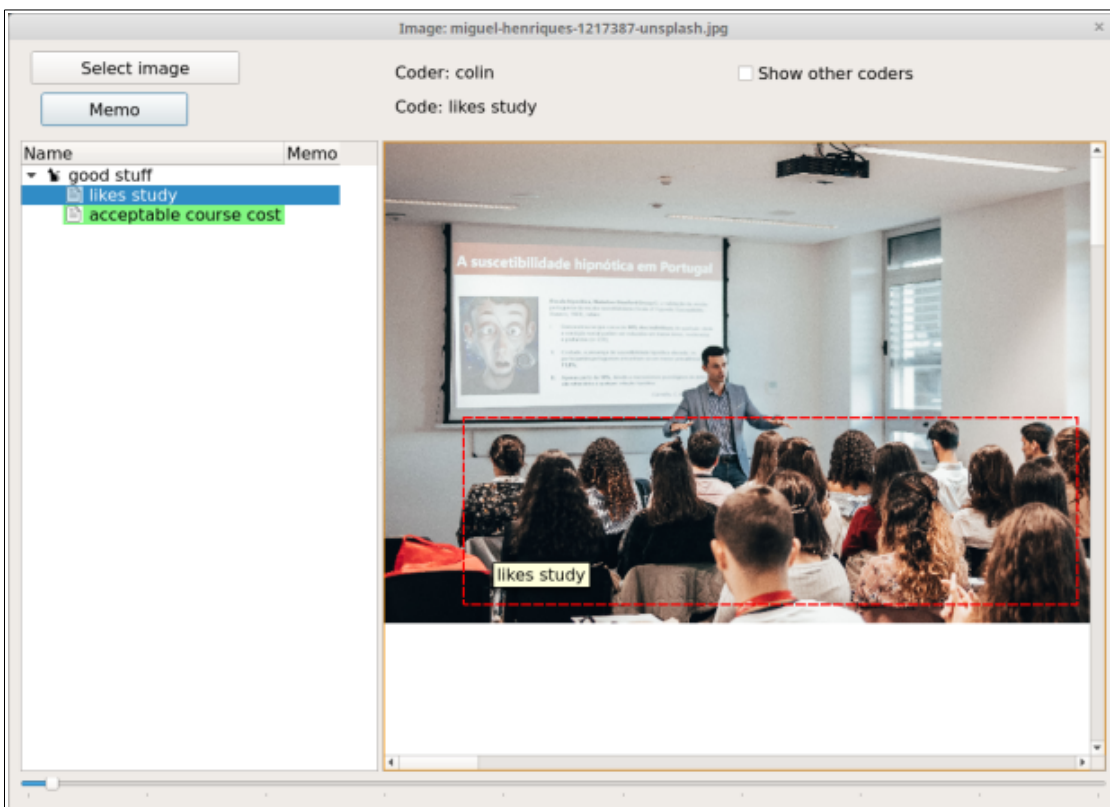
Categories are used to organise codes. Categories are organised hierarchically in a tree structure. The *cat* symbol denotes a category, whereas the *writing-paper* symbol denotes a code.

You can move codes into categories and move categories into larger categories. You can move categories and codes out of their current position. Codes and categories can be merged by dropping a code onto a code or a category onto a category.



### 4.3.7 Coding images

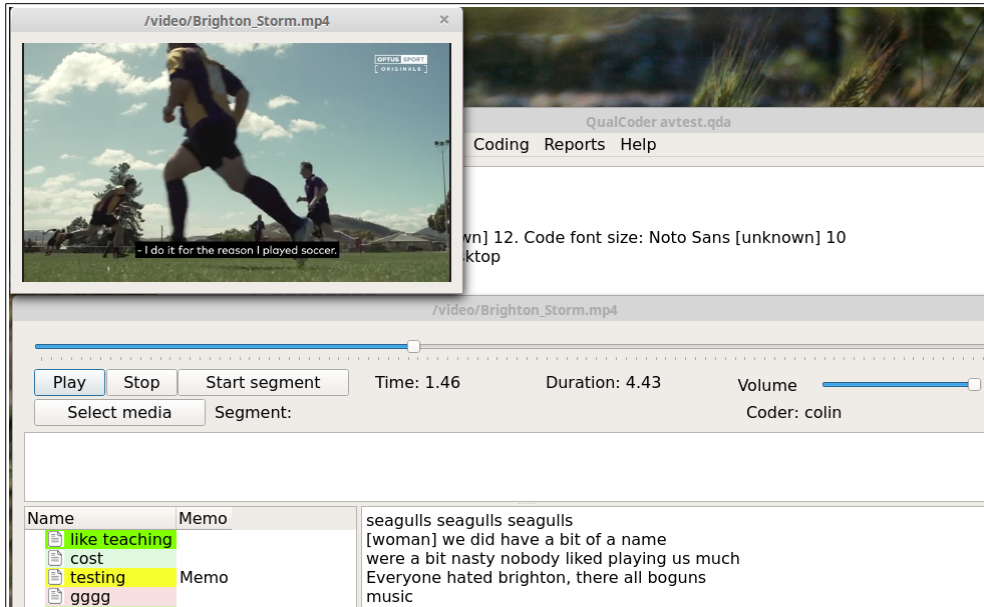
Images can be coded in a similar way to text coding. Select a code. Left-click and drag to highlight the area you want to assign to the code. Right-click will open a menu where you can remove the coding or add a memo. At the bottom of the screen there is a slider control to re-size the image.



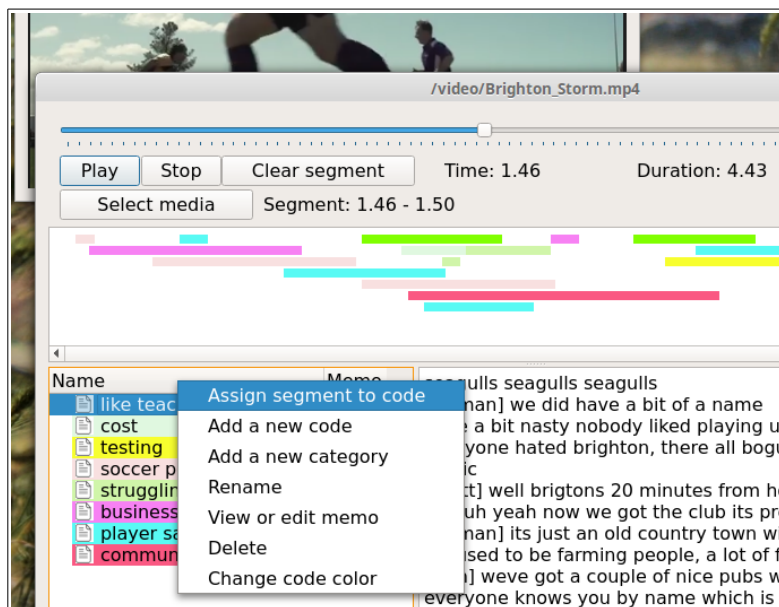


### 4.3.8 Coding audio and video

When coding audio or video, two windows are displayed, shown below. One window has the audio or video playing. The other window has the controls and a button to begin and end a coded segment. The controls window shows the codes in the bottom left pane and any transcript is shown in the bottom right pane.

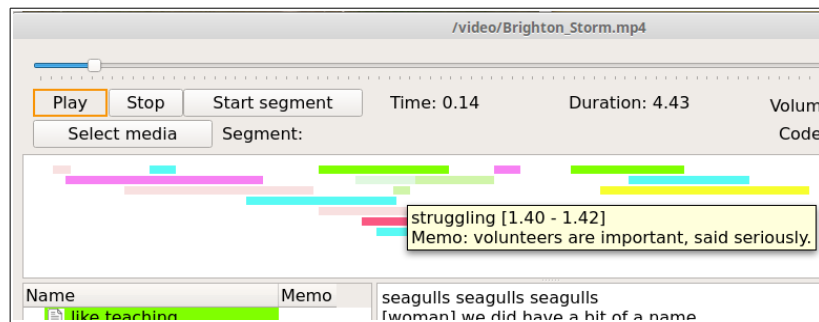


Once a segment is made using the *Start segment/Stop segment* button this can be assigned to a code by right clicking on the relevant code and assigning the segment.



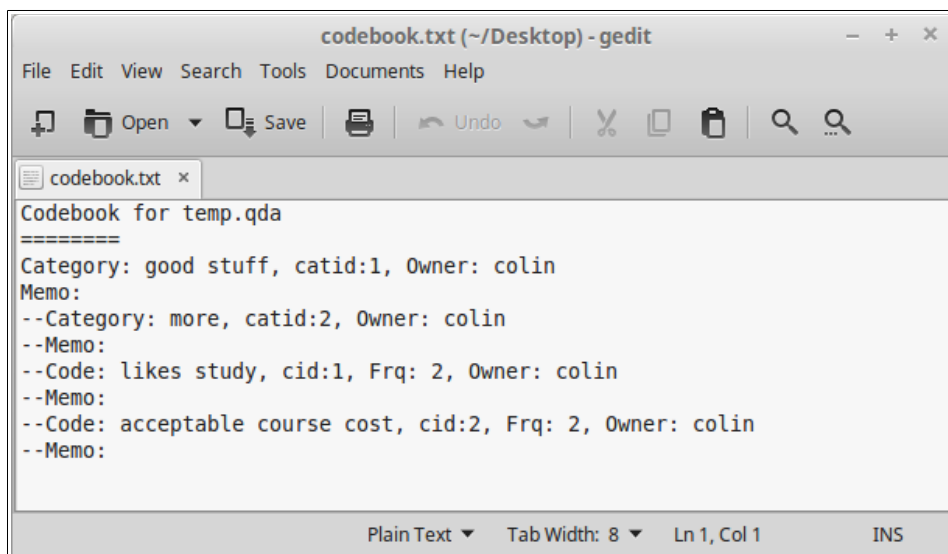
Coding stripes for the coded segments are shown in the upper pane. Hovering the mouse over each stripe shows the code name, time segment and any memo attached to that coded segment. Right clicking on a coded segment stripe shows a menu that can be used to edit the memo or delete the

coded segment. Coded stripes are shown on various lines so that they do not overlap. An example of coding stripes and mouse hover is shown below.



### 4.3.9 Codebook

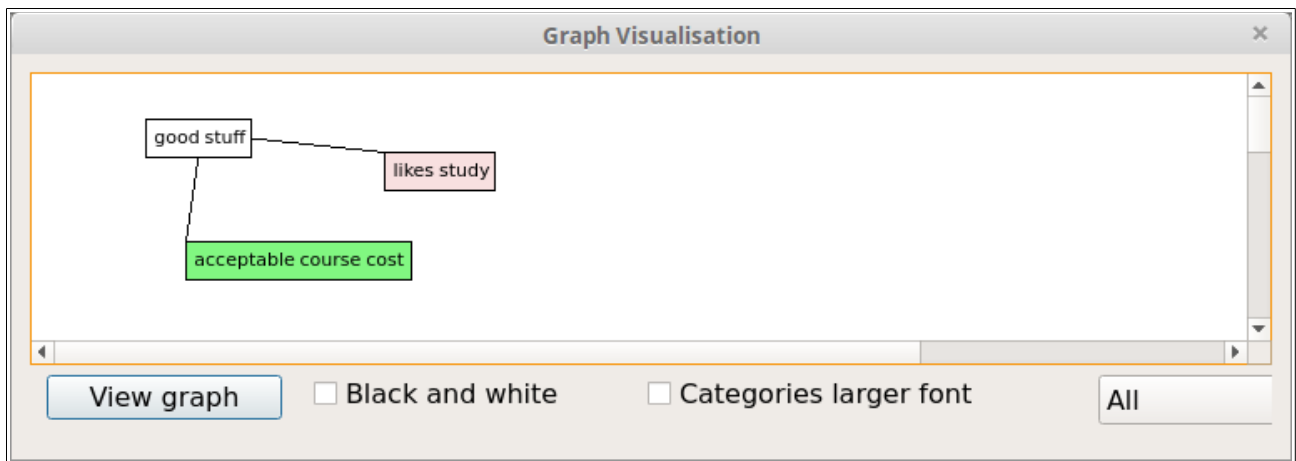
A codebook is a list of your codes. Each code memo should detail the reasons for the purpose of each code. The codebook can be exported to a text file. The codebook also shows the frequency of the codes used (from all coders). Example codebook output is shown below. As codes and categories are put in a tree like structure the double minus ‘--’ indicates the subordinate codes and categories within a category.



### 4.3.10 Node graph

A graph of categories and codes is displayed. There are several options to change what is shown – such as Black and White, All or selected categories.

Each code or category can be moved around by clicking near the edge of the box and dragging it around. Clicking on the text allows you to temporarily change the code or category wording. Right-clicking in a code gives a menu of options such as displaying the memo, or displaying all the case or file text that has been coded with the selected code.



## 4.4 Reports

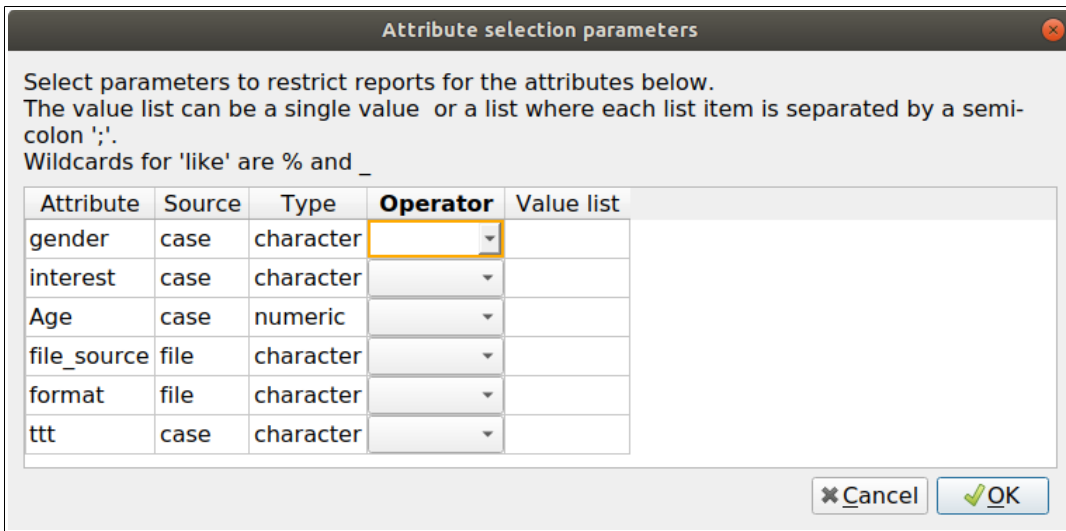
### 4.4.1 Coding Reports

This dialog gives a list of coded text based on your selections. One or more codes need to be selected and one or both of *coder 1* and *coder 2* need to be selected. The *Search* button will present the results. If a category is selected all codes in that category are also selected. Multiple categories can be selected using the mouse and the 'ctrl' button. Codes and categories can be sorted to help find the codes of interest.

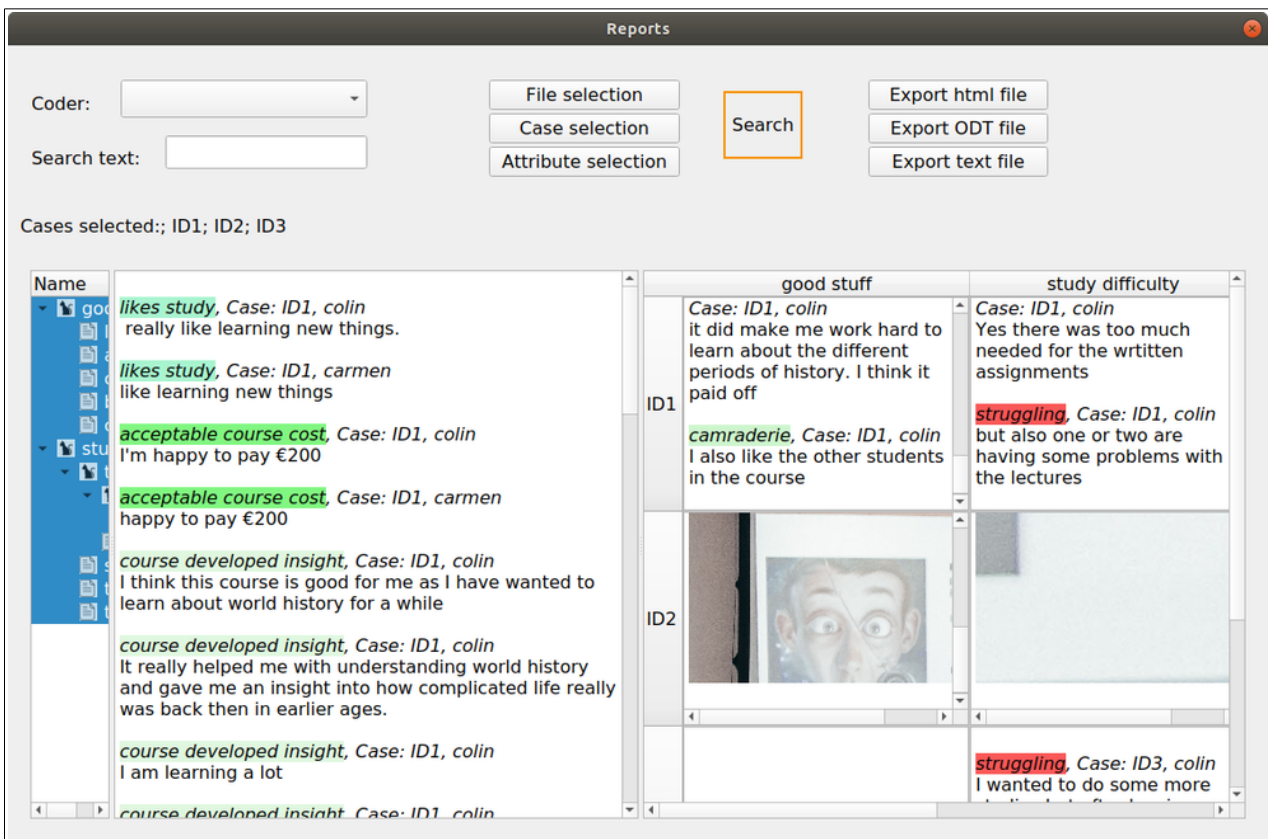
Codings can be narrowed down by using the *File selection*, *Case selection* or *Attribute selection* buttons. Only codings within the selected files or cases will be presented. If text is entered into the *Search text* field, only codings which contain the matching text will be presented. Codings can also be narrowed down to case selections using the *Case selection* button. Note that when case selections are used, file selections are ignored. Also when file selections are used, case selections are ignored.

The *Search Text* box also limits the reported codings. The search text looks for any codings in text that contain the search text. The search text also looks for matching text in memos for coded image areas and coded audio/video segments. The search text function must be used in combination with *File selection*, *Case selection* or *Attribute selection*.

The *Attribute selection* button opens a dialog window where you can select attributes for files and cases, for example: only interview files for people aged > 60.



If you are creating a report based on Case selection, then the code tree pane is hidden. Instead two panes are shown. The left pane shows the codings in a list, the right pane shows a matrix with one case on each row and top-level categories in each column, shown as below:



Each of the three panes can be widened or narrowed by dragging with the mouse on the bar that splits each section.

Reports can be exported to text, open document (ODT) or html files. Currently case matrices are not able to be exported.

## 4.4.2 Coding Comparison

This option shows the similarities and differences between two coders. Select two coders and click the *run comparisons* button. This option is only available for coded text, not coded media files. For each code:

Agreement % shows agreement for a combination of coded and non-coded text characters.

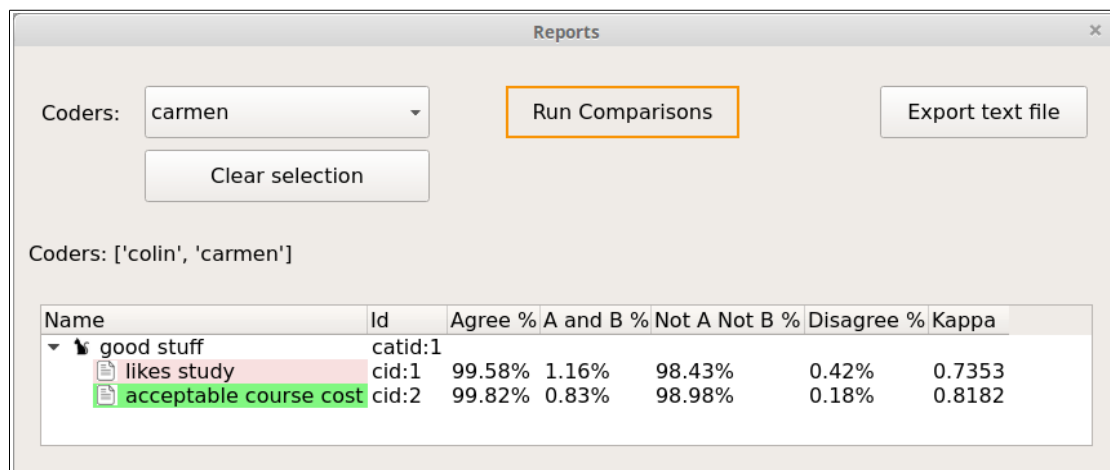
A and B % shows agreement for the only the coded text characters divided by the total characters in the text.

Not A and Not B % shows the total of the uncoded text divided by the total characters in the text.

Disagree % shows the percentage of all the coded and non-coded text that did not match between coders. It is the same as 100 – the Agree %.

Cohen's Kappa is calculated based on the information in Wikipedia

[https://en.wikipedia.org/wiki/Cohen%27s\\_kappa](https://en.wikipedia.org/wiki/Cohen%27s_kappa)

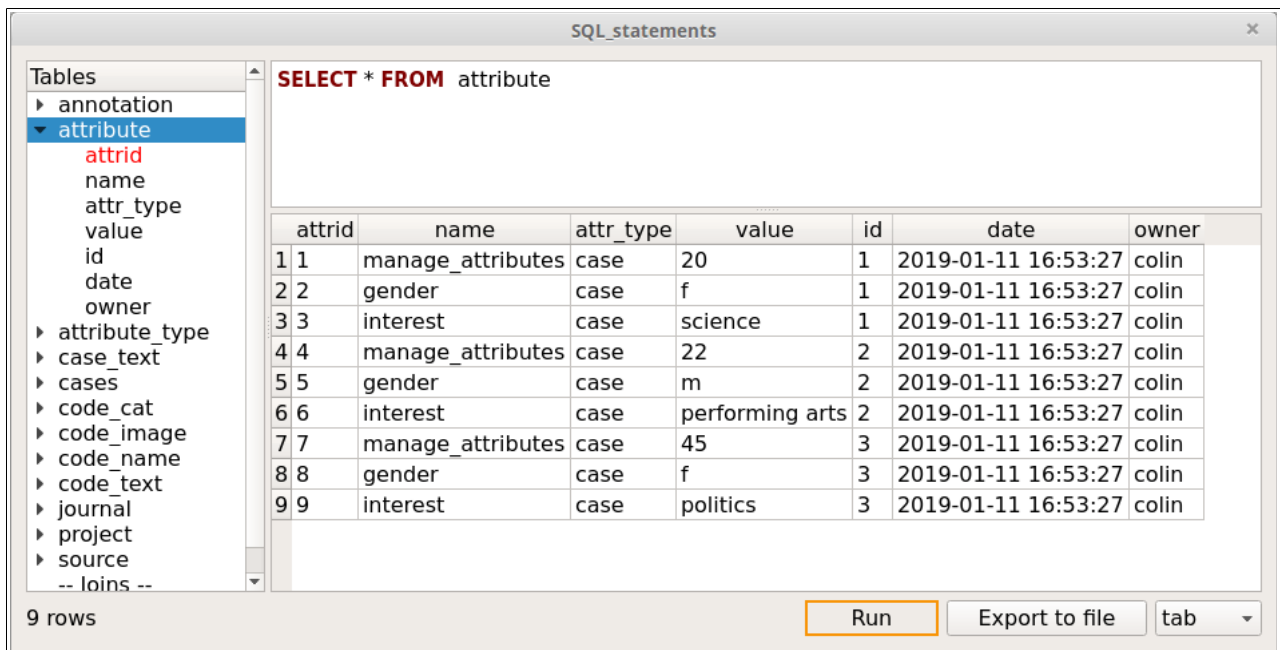


The screenshot shows a 'Reports' dialog box with a 'Coders' dropdown set to 'carmen'. A 'Run Comparisons' button is highlighted with an orange border. Below the dropdown is a 'Clear selection' button. The 'Coders' field shows ['colin', 'carmen']. A table displays the results for a category named 'good stuff'.

Name	Id	Agree %	A and B %	Not A	Not B %	Disagree %	Kappa
good stuff	catid:1						
likes study	cid:1	99.58%	1.16%	98.43%		0.42%	0.7353
acceptable course cost	cid:2	99.82%	0.83%	98.98%		0.18%	0.8182

## 4.4.3 SQL Statements Dialog

This dialog contains three panes. The top pane is where SQL statements are entered and the bottom pane contains the results of queries. The left pane contains tables and field names. Double-clicking on a field name adds it to the SQL statement. Results can be exported to a text file. If you are not familiar with SQL take care as you will be able to update and delete the data as well as select data. Note: Some Unicode symbols are not converted to plain text and are ignored.



Most table fields are text. The following fields are integer: *anid, avid, attrid, caseid,catid, cid, fid, id, imid, jid, pos0,pos1, x1, y1, width, height*.

## 4.5 Multiple dialog windows

It is possible to have multiple dialog windows open. This is useful if you have run a report, or have a file open for coding, and you want to enter your thoughts into a journal entry. However there are some limitations.

Some dialog windows are priority windows, so swapping between windows will not be possible, these restricted windows are: Management of files, cases or attributes, survey import and confirmation dialogs (e.g. confirm delete).

The second limitation is that changes in one dialog will not be reflected in the other dialogs. An example here is adding a code in a coding dialog, the added code will not be shown in other coding or reports windows. If you do make changes to highlighting and codes and categories, it is recommended to close and then re-open the other dialog window that you need. This is to ensure that any changes made show up in the other dialog windows. Similarly, highlighting text on the same file in two opened dialog will not be reflected in the other window. **A warning**, if you have the same journal entry open in two windows, the last window that is closed will be the entry that is saved.

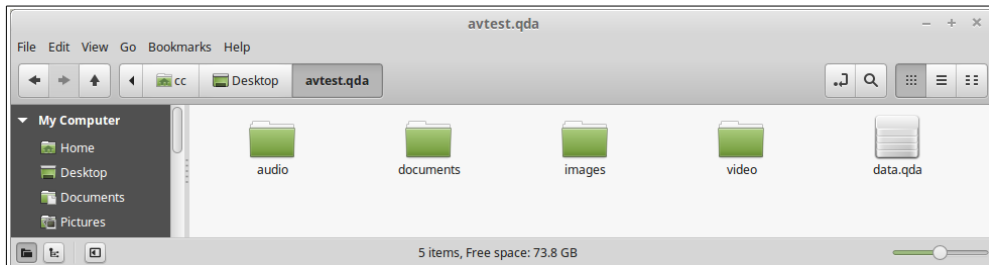
The screenshot displays the QualCoder software interface with several panels open:

- Settings:** Shows configuration for the coder (colin), font (Noto Sans), and directory (/home/cc/Desktop).
- Journals:** A table with columns for Name, Date, and Journal content. One entry is visible: 'first' on '2019-02-26 07:59' with the text 'my first journal'.
- Codes:** A list of codes on the left and a text editor on the right. The text editor shows a paragraph of text with several codes highlighted in different colors: 'likes study' (green), 'course developed insight' (green), 'beneficial to career' (green), 'camraderie' (green), 'study difficulty' (green), 'too hard' (green), 'teaching help' (green), 'more tutor help' (yellow), 'struggling' (red), 'slides too busy' (red), and 'too many' (red).
- Reports:** A panel for generating reports, including a search bar and buttons for 'File selection', 'Case selection', and 'Attribute selection'. It shows 'Cases selected: ID1; ID2; ID3' and a list of selected cases with their corresponding text snippets.
- Graph Visualisation:** A network graph showing relationships between codes. Nodes include 'course developed insight', 'beneficial to career', 'likes study', 'good stuff', 'camraderie', 'acceptable course cost', 'struggling', 'teaching help', 'slides too busy', 'study difficulty', and 'more tutor help'. Lines connect related codes, such as 'course developed insight' to 'good stuff' and 'beneficial to career', and 'likes study' to 'good stuff'.

## 5 Other details about the software

---

The qda data folder contains folders for imported documents, images and audio (although audio is not currently implemented). It also contains the sqlite database, named data.qda, to store coding data.



QualCoder creates QualCoder.log and QualCoder\_settings.txt files in your home directory. The settings file contains the name of the current coder, a default working directory and choice of font. The log file records program errors and some user actions.

QualCoder is written in python 3 using Qt for the graphical interface. A Sqlite database is used to store the coding data.



## 6 Future plans

---

Currently QualCoder has only used on Ubuntu, Linux Mint and by a github contributor (WPFilmmaker) on Lubuntu. Most functionality testing has been performed using Windows 10. In Windows, reports exported in ODT format are okay, but some images may overlap when the ODT file opened with Microsoft Word rather than opened with LibreOffice.

Further testing on different operating systems is required.

Some potential plans for the future are to add functionality.

Coding:

- Possibly add alternative ways to view text coding.

Reports:

- Some statistics: word count, possibly add bar or pie charts for some attributes.
- Possibly look at text mining functionality
- Possibly have an html report with embedded audio/video.

## 7 About the author

---

Hello, my name is Colin Curtain and I am a pharmacist and lecturer from Australia. I have many interests including clinical pharmacy, computer programming, research, statistics and clinical decision support. I completed a PhD evaluating computerised clinical decision support in 2014. When doing my PhD I used [R](#) as the statistics program of choice. This is where my interest in qualitative data analysis and the use of [RQDA](#) came from, which ultimately led to this project.

Originally when doing my PhD I analysed qualitative survey data via a thematic approach using RQDA. I then thought this could be reproduced in Python, so I scripted an earlier version called PyQDA which worked OK at the time. This has been forked and is currently developed by [Ronggui Huang](#).

I thought I would share QualCoder in the hope that it may help others. This is still an early version so bugs are possible and functionality could be further extended. I also generally work on Linux machines, and sometimes Windows, so this software probably needs further tweaking for best use on Windows. I work on the programming for this in my spare time, it is not my full time job for me.

If and when you use QualCoder and publish your results, I would really appreciate it if you let me know the bibliographic information of your work.