

**Programs for the Collection and Analysis of Observational Data:
Manual for Version 2.75.5**

Dr. William L. Roberts¹
Thompson Rivers University
21 July 2012

The public domain programs described in this manual can be downloaded from
<http://sourceforge.net/projects/behavioraldata/>

¹ Mailing address: Psychology Department, Thompson Rivers University, 900 McGill Drive, Kamloops, BC V2C 0C8, Canada. Email: wroberts@tru.ca. DOS versions of these programs were presented at meetings of the Society for Research in Child Development, Seattle, April 1991, and New Orleans, March 1993.

Copyright

The public domain programs described in this manual are Open Source under the GNU General Public License (see <http://opensource.org/licenses/gpl-2.0.php>), but the bound (*.exe) versions include proprietary libraries copyrighted by True Basic, Inc. (see www.TrueBasic.com). The programs may be used for purposes of research and teaching free of charge, or copied and distributed to others. Please include this notice and documentation. *If you find these programs useful, I would be grateful if you cited this Manual in your publications.*

If you modify these programs, please send copies of the new source code to me at wlroberts@tru.ca, so that I can distribute the code and programs to others.

Acknowledgements

I acknowledge with gratitude the support of the Social Sciences and Humanities Research Council of Canada and York University in the development of the DOS version of these programs, and Thompson Rivers University for its support for the current Windows version. I would like to thank Dr. Debra Pepler of York University, Dr. Wendy Craig of Queens University, and my research assistants at York (especially Lisa Kelner and Anne Stephenson) for their advice and feedback when using earlier versions of these programs. Thanks also to Susanne Denham of George Mason University and her research assistants for suggestions on the current Focal interface and the diligent reporting of problems.

Table of Contents

Introduction..... 8

 The programs: an overview..... 8

 Steps in collecting and analysing observational data..... 12

FOCAL: Collecting Focal Individual Data..... 15

 Overview..... 15

 Program files..... 16

 Running the program..... 16

 Collecting data..... 18

 The Number Pad..... 19

 Quick Entry..... 20

 Pause..... 20

 Ending data collection..... 20

 Do another focal sample..... 20

 Edit data file..... 21

 Placing comments in data files..... 21

 Make session notes..... 21

 Edit behaviour or person codes..... 22

 A sample data file..... 23

Summary..... 24

SCAN: Checking and summing FOCAL data..... 26

 Overview..... 26

 Getting Started..... 26

Summary..... 30

KAPPA: Reliabilities for FOCAL data..... 32

 Overview..... 32

 Running the program..... 33

 Selecting data files to be analysed..... 33

- Creating a new list of files. 33
 - Select two files. 33
 - Select multiple files. 34
 - Select multiple files of the same type. 34
- Using an earlier list of files. 34
- View Menu. 35
 - Results. 35
 - Sample output from KAPPA. 36
 - Finding a given code in all data files. 38
 - Matched Events. 39
 - Sample output I: Focal data files matched by time. 40
 - Sample output II: Focal data files matched by event sequences. 41
 - Recoding (combining) behaviours. 42
 - Files to be analysed. 43
- Analysis Menu. 43
 - Begin analysis. 43
 - Set time tolerances. 43
 - Set parameters for event-matching. 44
 - Combine behaviour codes. 44
 - Use row totals for percent agreement. 44
 - Use mean comparisons for percent agreement. 44
 - Use time to coordinate comparisons. 45
 - Use event sequences and matched events to coordinate comparisons. 45
- Options Menu. 45
 - Report matched events. 45
 - Show Table of Comparisons. 45
- How KAPPA compares events. 45
 - Using time to compare events. 46
 - Using sequences of events to make comparisons. 46
 - Which method is best?. 47
- Accuracy, false positives, and false negatives. 48
- Exporting and importing a Table of Comparisons. 49

Final thoughts about KAPPA. 51

Summary.. . . . 52

TIMES: Frequency, rate, and proportional times for FOCAL data. 54

Overview. 54

Running the program.. . . . 54

Combining behaviour codes. 55

Pooling data across subjects. 55

Should identical adjacent codes be combined?. 56

Restricting the Analysis to Certain Contexts. 56

‘Stop’ codes: Coding simultaneous behaviours.. . . . 57

Designating, using, and deleting 'stop' codes. 59

Summary.. . . . 60

LAG: Sequential analyses for FOCAL data. 61

Overview. 61

Running the program.. . . . 62

Searching for antecedent events. 62

Combining behaviour codes. 63

Combining actor-target codes. 63

Should identical adjacent events be combined?.. . . . 63

Using all focal samples.. . . . 63

Restricting the Analysis to Certain Contexts.. . . . 64

Instructions for a log-linear analysis. 64

Removing ‘stop’ codes. 64

Sample output for a lag sequential analysis.. . . . 65

Some thoughts on z-scores, Yule’s Q, and Phi. 67

Summary.. . . . 71

SEARCH: Patterns of contingent behaviour in FOCAL data. 72

Overview. 72

Running the program.. . . . 72

- Combining behaviour codes. 73
- Combining actor-target codes. 73
- Should identical adjacent codes be combined? 73
- Restricting sequence length. 73
- Setting the level of significance. 73
- Removing 'stop' codes. 73
- Sample output for Search. 74
- Summary.. 76**

- COUNT: Patterns of behaviour in FOCAL data. 77**
- Overview. 77
- Running the program. 77
- Combining behaviour codes. 78
- Combining actor-target codes. 78
- Should identical adjacent codes be combined? 78
- Removing 'stop' codes. 78
- Sample output for Count. 80
- Summary.. 81**

- INTERVAL ANALYSIS: Identifying episodes of behaviour. 82**
- Overview. 82
- Identifying intervals between criterion behaviours. 82
- Detailed information on intervals. 84
- Modeling the distribution of episodes. 84
 - Measures of deviation. 88
 - A note on zero values in the simulated and observed distributions. 89
- Refining the search for a cut point. 89
- Generating labeled versions of the data files. 91
- Locating all episodes. 91
- Summary. 92

- References. 93**

Appendix.	94
Installation..	94
Uninstalling the programs.	94
Program limitations.	94
Known issues..	95
Technical Support.	95

Introduction

These programs allow you to collect and analyse observational data in natural (field) settings, using a laptop or netbook computer. I have tested them on Windows 7, Vista, Windows XP, and Windows 98. They also run on Linux under *Wine* (I have tested them on Kubuntu and Mandriva Linux), and they should work on the Macintosh, using *Crossover* (<http://www.codeweavers.com/products/>). Macintosh and Linux applications can be used to edit the data and output files. The programs display properly at low screen resolutions (800 x 600), and of course do well when higher resolutions are available.

In this Manual, an *event* has three elements – an *actor* who preforms a *behaviour*, and a *target* to whom the behaviour is directed. The next event in the data set may record the reaction of the target to the actor. Thus FOCAL data are interactive data, although they can also record states. A *criterion* (or *criterion event*), specified by the user, serves as the starting point for assessing contingency in a data set (analyses of contingency are often referred to as *lag* analyses).

The programs: an overview

The first program in this suite is FOCAL, a program for data collection. Although designed for coding continuous, focal-individual interactions, it can also be used to collect instantaneous scan sampling data (Altmann, 1974). With the use of “stop” codes (see the chapter on TIMES), simultaneously occurring behaviours can be coded. FOCAL allows users to specify different contexts (e.g., playground; classroom; home) for each focal-data sample. The analysis programs can examine these contexts separately or in combination.

Up to 100 user-defined numeric codes for behaviours and persons (actors or targets) can be displayed on-screen. Normally, users enter codes for actors, behaviours, and targets, and durations are recorded automatically; but when coding dyadic interactions, "Quick Entry" mode allows FOCAL to insert the most recently used codes for actor and target as defaults, so that only behaviour codes need to be entered. Quick Entry can either alternate the actor and target codes, as these individuals interact, or keep them constant. Errors in data entry can be corrected

when they are made, or flagged for later. Corrections to the data file can be made at the end of each trial or sample. Narrative session notes can also be made between trials. (These will be headed by information for the data sample just recorded).

When data files are edited and corrected, comments can be placed at the end of data lines. (They must be preceded by a “#” or “!”).

SCAN checks FOCAL data files for errors that may have been introduced during editing. It will correct simple errors (saving the original data to a backup file), and flag others. SCAN also reports total time observed and total events recorded. If multiple data files are specified, SCAN will report results for each file individually and for the entire set. If multiple files are used and their names saved, this list will be loaded automatically when you begin a new work session. (The list can also be used by the analysis programs). Because it checks data file integrity, SCAN should always be used before running KAPPA or the analysis programs.

KAPPA assesses reliabilities, reporting percent agreement, kappa, and the 95% confidence interval for kappa for the entire taxonomy. It also reports percent agreements and partial kappas for individual codes, as well as five codes most often confused with the individual code. To assess reliability of event durations (important for time-budget analyses), KAPPA correlates durations across observers as an overall measure of reliability, and also identifies problematic individual codes with standardized residuals.

Because observers coding in real time will necessarily be slightly out of synchrony even when they agree about which codes to use, KAPPA allows users to specify how much time discrepancy to tolerate, both when identifying matching sequences of events, and when matching events by time. As an aid in training or in understanding problematic levels of agreement (especially for rare behaviours), KAPPA can also mark and display agreements, matching events and sequences of events in copies of the data files being compared.

Reliability can be assessed for sets of observers as well as for individual pairs. Thus KAPPA can report overall reliability at a given time (say, at the end of training, or at a given point during data collection), or across an entire project.

LAG assesses contingency, and contains five modules or programs. The first, LAG, reports conditional probabilities for all events following (or preceding) a specified actor-behaviour-target event (the criterion event), at any given distance, or lag. LAG also reports rates and frequencies. Contingencies between a specified criterion and the events that follow it are assessed by exact binomial probabilities. LAG also tests the omnibus null hypothesis that *all* lag probabilities are zero.

LAG can generate copies of the data files, substituting descriptive names for the numeric codes, using the code-name files from FOCAL. Thus episodes can be identified and reliability established without the necessity for research assistants to learn numeric codes.

LAG is capable of producing output for log-linear analyses, if desired. This output is in the form of a BMDP instruction set, with data. It can be run directly by BMDP 4F, or modified for use with SPSS. A STATXACT import file may also be created, so that exact chi-square probabilities can be assessed when expected probabilities are low. Thus contingency can be assessed generally, across an entire taxonomy, as well as for specific events and their sequelae.

Using techniques described in Bakeman and Gottman (1997), SEARCH, the second module of LAG, finds sequences of contingent behaviour that begin at a specified criterion, based on output from LAG. SEARCH will automatically generate needed LAG output, or it will examine existing output.

When comparing sequences of behaviour across groups (for instance, of aggressive and non-aggressive children), it is necessary to know how often a sequence that was statistically significant in one group occurs in the other. The COUNT module of LAG can read sequences directly from SEARCH output for one group, and report their frequency for a second group. Group differences in frequency are assessed by chi-square, using Yate's Correction, which produces *p*-values that are more accurate than the usual asymptotic estimates (which are also reported). Users may specify sequences. These will be counted, but their statistical significance will not be assessed.

The time-budget module, TIMES, calculates time-budget results for focal-

individual or instantaneous scan samples (that is, it reports durations as proportions of total time observed, and events as proportions of total events observed). Rates (as events per minute) and mean durations (in seconds) are also reported. With proper coding, TIMES can handle multiple simultaneous events.

The fifth module, INTERVAL ANALYSIS, contains several tools that are useful for establishing criteria for the end of episodes. Episodes (of conflict, conversations, etc.) are characterized by codes that occur frequently for a short period of time, followed by intervals when they do not occur at all. Thus examining interval lengths can inform us about the nature of episodes. The interval analysis module displays a graph of the distribution of the intervals between user-specified events and provides basic descriptive statistics as well as a sorted list of all intervals. The program can also generate random distributions indicating the probability that any given focal sample will contain n episodes, a feature that can be used to test various criteria for defining episodes. (Good criteria generate random distributions.) The use of these tools will be illustrated by a specific example in the appropriate section later in this manual.

For all programs, options set by menu are remembered in a configuration file, so that they do not need to be re-entered at the next work session.

KAPPA and the five programs in LAG all permit behaviour codes to be combined to form new, more inclusive categories. All except KAPPA also permit combining actor-target codes in order to pool data across individuals in a data file (e.g., all same-sex peers who interacted with a particular focal child). Recoding information is kept in small text files to facilitate use across programs. The original data are left unchanged.

Because data for each subject are kept in separate files, it is possible to pool across several subjects (e.g., all girls) by selecting the appropriate group of files. When multiple files are specified at the beginning of an analysis, KAPPA, LAG, SEARCH, COUNT, TIMES and INTERVAL ANALYSIS will analyse the pooled data as well as

report results for individual files. When multiple files are specified, the list of file names can be saved and used other analyses, as noted earlier.

Results of analyses are placed in output files and displayed on screen. They may be printed, if desired. Because output files are text files, they can be imported into a word processor (such as Word, WordPerfect, or Open Office Write) for editing and printing, or they can be read by text editors such as NoteTab (www.notetab.com) or Kate (<http://kate-editor.org/>). Thus it is usually a straightforward matter to transfer rates or conditional probabilities (for example) from LAG or TIMES output files to BMDP or SPSS for further analysis.

Prior to version 2.70.2, I included a program called COMPARE, which, as its name suggests, could be used to compare edited and backup versions of a data file, displaying discrepant lines in context (that is, with the lines immediately preceding and following). I have not maintained this program since I found the public domain program KDIFF3 (<http://kdiff3.sourceforge.net/>), which I prefer. If you would like to look at COMPARE , please download one of the earlier packages from my SourceForge site.

I'll be glad to answer any questions you have about using the programs (I can be contacted at wloberts@tru.ca). I am interested in hearing suggestions for improving the user interface or for including additional capabilities. **Bug reports are especially welcome**, as I would like to programs to work under a wide range of conditions. I hope you find the programs useful.

Steps in collecting and analysing observational data

This section is only a brief summary. There are several excellent books (e.g., Bakeman & Gottman, 1997) that describe this process in detail.

(1) After developing a set of numerical codes for persons and behaviours and placing them in text files (the program comes with examples that you can modify for your own use), FOCAL is used to collect data on interactions between a focal person and those others with whom the focal person interacts. FOCAL can also be used to

collect scan sampling data.

(2) Errors in data entry can be corrected when they are made, at the end of each focal sample, or, if you prefer, after data collection is over, using your own word processing program or a text editor like NoteTab. If you keep back-up versions of files as well as corrected versions, you may find it necessary at times to compare them in order to see what changes have been made. This can be done using Kdiff3 (<http://kdiff3.sourceforge.net/>), mentioned above.

(3) Use SCAN to ensure that editing hasn't accidentally altered the format of the data file. SCAN also reports total time observed and total events recorded for each data file.

(4) In order to assess reliability, two individuals must simultaneously code the same focal person. KAPPA will compare their data files, reporting per cent agreement and kappa (which corrects percent agreement for chance concurrence). As an aid in understanding when coders are agreeing or failing to agree, KAPPA can also copy the data files, marking matching event sequences in each file.

(5) LAG is used to investigate contingency in behaviour. The LAG module reports conditional probabilities for all behaviours that follow a selected ('criterion') actor-behaviour-target event, along with their exact binomial probabilities (which indicate whether these conditional probabilities are occurring more (or less) often than one would expect on the basis of chance). LAG can also report on events which precede a specified criterion.

LAG also reports frequencies and rates for the trials for which the focal person was either the initiator or target specified in the criterion event. Thus if a file contains data on a single family in which father, mother, and child served in turn as focal individuals, a criterion that specifies a mother-child interaction will produce an analysis based on only those trials in which either the mother or the child was the focal individual. (In contrast, the TIMES module of LAG can report frequencies and rates based on all trials in a data file – see below.)

(6) The SEARCH module of LAG can be used to find entire sequences of behaviour that follow a criterion event. SEARCH will utilize existing LAG output, or it will generate its own lag output as needed.

(7) If you are comparing sequences of behaviour in different groups, it is

necessary to know how often a sequence that was significant in one group occurs in the other. The COUNT module of LAG gives this information. It will read sequences from SEARCH output, or it will count user-specified sequences. Once the number of occurrences is known, proportions in the two groups are tested by chi-square.

(8) The TIMES module of LAG reports frequencies and rates for all behaviours, as well as mean durations and durations as a proportion of total time observed (time-budget data). If you have coded simultaneous sequential events, TIMES will give the proper percentage of observed time for these events.

If you have collected scan sampling data instead of sequential data, then analyse your data with TIMES rather than LAG. Percentage of events in a scan sample is equivalent to percentage of observed time in sequential data, and can thus give reliable time-budget information on states – if enough observations have been made.

FOCAL: Collecting Focal Individual Data

Overview

FOCAL is designed to collect sequential interactions, coded continuously in real time, using focal individual samples and exhaustive, mutually exclusive codes (see Altmann, 1974; Bakeman and Gottman, 1997). However, by using 'stop' codes, FOCAL can be used to record multiple simultaneous behaviours (see page 57.) Focal samples can be of any time length. At the end of each sample, the data file can be edited to correct coding errors, codes for persons or behaviours can be added or corrected, and any pertinent information or narrative description can be entered in a "session notebook". All FOCAL files are text files and thus compatible with other text and word-processing programs.

While FOCAL is intended for data collection in the field, it can also be used for coding videotapes in real time. Note, though, that while data collection can be suspended during a focal sample – see Pause below – there is no provision for reading videotape time directly into FOCAL.

Observations are recorded using a three-element syntax: the initiating individual, the behaviour, and the target individual. Each element can be coded with either one or two digits, 0 to 99. The target's response can be recorded, if desired, in the next three-element line. For example, if '10' codes 'talks' and '20' codes 'replies', a talk-reply sequence between persons 3 and 4 would be recorded as:

```
3 10 4
4 20 3
```

The program records this input along with the duration of the event and total duration since the beginning of the sample. Codes for behaviours and persons are, of course, user specified.

All coding categories can be displayed on screen. During data entry, codes for individuals and behaviours are checked as they are typed in, and non-matches cause the program to prompt for re-entry.

For the sake of speed during data collection, FOCAL temporarily stores data in RAM. At the end of each focal sample (typically 5 or 10 minutes long), all data are transferred to disk. Therefore, if you suffer power loss during data collection, you will lose the few minutes of data currently in RAM; but data from earlier sessions will be

safe. If you are concerned about power loss, data can be saved to disk during data collection – see the Save option, below.

Program files

FOCAL uses two ancillary text files, one for “person” (or actor-target) codes and another for behaviour codes . You should modify them to suit your own needs, rename them, and place them in a convenient directory of your choosing. Once Focal has used your new files, it will remember their names and where to find them.

The first text file contains the default listing of individuals to be coded, and is called DEFAULT CODES FOR PERSONS.TXT. Up to 100 individuals can be listed in this file, but you will probably want to list only those that will be common across families or groups. You can identify this file as the default file in the *Preferences* menu. Will

The program will allow you to name new family or group members at the beginning of each session (see below). These changes must be saved in order to become effective. In future coding sessions, FOCAL will look for and load this file . Therefore, additional family or group members only have to be specified once.

The second file, “DEFAULT Behaviour CODES.TXT”, contains a default set of behaviour codes. It should be replaced by your own set of codes, renamed, and placed in a convenient directory. As noted, FOCAL will look for and load this file in future sessions.

Note that, as in the sample files, numeric codes must be separated from names or behaviours by at least one space. Blank lines and lines with numbers only will be displayed on screen (and may make the display more readable), but they will be ignored by the routine that checks entries for errors.

Running the program

After starting the program, you will be prompted for identifying information for the current data session.

Subject ID

Because the program will use this ID to label data and session note files, you should select characters that Windows can use for file names. If the subject ID has been used in an earlier session, FOCAL will append data from the current session to

the end of the data file created earlier.

Session

You can enter additional numbers or characters to identify the session. The program will use this as header information in the data file. (Date and time will be added automatically.)

Focal person

Enter the numeric “person” code (actor or target – not the subject ID) that identifies the focal person for the current data sample.

Length of focal sample (in minutes)

Enter the duration of your focal sample. (If you wish, the program will make a brief ringing sound to signal when this time has elapsed. You can set this option later, in the Options menu of the Data Collection window). You may continue to collect data, of course, for as long as you want.

Observer

Enter your name or other ID, as the coder or observer for this session.

Context

You may optionally list up to four different contexts (e.g., “playground”, “classroom”, “home”, “school”). The analysis programs in LAG will be able to analyse data by context (or sets of contexts), allowing you to assess possible differences in behaviour. Contexts can be listed in any order (that is, you don’t need to be consistent about the ordering from session to session – but *do* be consistent about spelling and the use of upper and lower case). Because the programs use commas internally to delimit or separate contexts, do not use them when naming contexts. Thus “home morning” will be treated as one context if it is entered in one box, but “home, morning” will be treated as two, and should be entered in separate boxes – if you want them each to be a context.

Working directory

This is the directory where your Focal data files will be placed. If you wish to change it, click on the “Change directory” button, navigate to the directory that you would like to choose, and select any file in it. (The program will save the directory information and discard the file name).

After clicking “Continue”, you will be taken to the data collection window. You

can change any of the values just entered by clicking on *Analysis Menu, Edit Header information*, in the main window at the top of the screen. If this is the first time you've started Focal, you should identify your default person-codes file in the *Preferences* menu. Later, you will be able to upload this file from the person-codes window in the lower right portion of the screen. (You may view your current defaults from the *Preferences-Information* menu).

Collecting data

The header, or top line in the top window, reports the subject and session ID supplied earlier; the date and time; the trial number (focal sample number); and the focal individual for the current sample. This information is also sent to the output data file (see below).

Codes for persons and behaviours appear in the two windows on the lower right. You may alter these codes at this point, although the program will only recognize changes when the files are saved (be sure to use the menu in the window in which you were working). As mentioned earlier, the program will check codes during data entry, and will not permit codes other than those listed. Therefore, if you are observing in circumstances in which others may unexpectedly join the focal individual, you may wish to enter one or two "dummy" names in the "person" codes. You will be able to change or delete them at the end of each focal sample. Remember, code files must be saved in order for changes to become effective.

To begin data collection, chose "Begin coding" from the *Data Entry Menu*, or use the keyboard shortcut when in the *Data Entry window* or the main (top) window.

If your person or behaviour taxonomy only has codes numbered 0 to 9, FOCAL will disregard the leading zeros shown in the code files. However, if any code has two digits, then all codes, even those numbered less than 10, must have two digits (that is, codes 1 to 9 must be entered as 01, 02, etc.). This constraint exists because FOCAL keeps track of actor, behaviour and target codes by counting digits, formatting the in-coming data-stream appropriately, and moving on to the next line automatically (no need to hit the <Enter> key). The program takes care of these house-keeping chores so that you can give your whole attention to observing the

scene before you. This desire to make data-collection as streamlined as possible also underlies some non-standard key assignments in the number pad, as described in the next section.

The Number Pad

On many laptops, the numeric keypad overlaps the right hand portion of the alphabetic keys. This keypad is automatically activated by the program. (The 'home key', 5, is the <i> key).² If your laptop has a separate keypad, it will work as expected.

In addition to the <0> key on the top numeric row and the <O> key at the lower left of the number pad (located on the <m> key), "0" is also entered when you press the <n> key (which is adjacent to <m>) and the space bar (to better mimic a standard numeric keypad).

Errors in the current data line can be cleared or earlier errors noted without moving one's hand from the 'home key' in the centre of the number pad. Speed is of the essence. Thus the semicolon key <;> as well as the backspace key clears the last *element* in the current line, not just the last character. For example, if you note that the two-digit behaviour code that you just entered is wrong, either the <backspace> or <;> key will delete both digits and the spaces that followed them, so that you are ready to enter a new behaviour code immediately. The apostrophe key <'> will clear the entire line.

While previous data lines are displayed on the screen in a running record of the session, they cannot be corrected during data collection. (Because coding is occurring during real time, this would result in a loss of data.) If you notice an error in an earlier line, strike either the <e> or <?> (</>) keys to drop an error flag (>>ERROR<<) in the data file. (The unintuitive <?> </> key is activated so that error flags can be dropped without moving your hand from the number pad home key.) After the focal sample is over, you can edit and correct your data (see below). Because the data file is a text file, you can also edit it after the session using your word or text processor. The error flag will help you locate the line that needs correcting.

² Note that this doesn't work for Dvorak keyboards, as the keys are distributed differently; but I could implement it if there is a need. Please contact me.

Quick Entry

In "Quick Entry" mode, FOCAL will insert actor and target codes; all you need to enter are behaviour codes. This is convenient when coding dyadic interactions.

You can toggle Quick Entry on or off from the Data Entry Menu before you begin collecting data (a check mark appears by the menu item when it is active) or by pressing the <F9> key. This can also be done during data collection.

If you activate Quick Entry before entering any codes, FOCAL will expect a complete first line of data, from which it will extract default actor and target codes. If you have been entering data before you toggle on, the last complete data line supplies the default actor and target codes. Actors and targets are alternated (as you would expect when coding interactions or exchanges), unless this option is toggled off (Data Entry Menu - Options). (If you do change this option, I suggest that you first turn off Quick Entry).

Once Quick Entry is "on", it stays on until you turn it off. Thus if you quit a data trial with Quick Entry on, it will still be on when you start the next data trial. (You will need to start the new trial with a complete data line, however, to supply the default actor and target values for the new trial). When you restart Focal for a new session, it will remember your last settings (Quick Entry on or off; actors and targets alternating or not). If this is a nuisance, please contact me, and I will change the defaults!

Pause

If you are coding from videotapes, you may wish to pause during coding to consider an ambiguous sequence of behaviour. Strike the <F5> key or chose Pause from the menu. A message will appear indicating that the program is waiting and the clock will stop. When you are ready to continue, simply resume entering data.

Ending data collection

To end the current sample, strike <q> (quit) or <Esc> or chose 'Save and Quit' from the menu. The program will save the data to disk. After this, you can use any of the windows, and you have several options.

Do another focal sample.

If the next data sample will be on a different focal person or a different subject, select "Edit header information" from the *Analysis* menu in the top window. You will

be returned to the opening information box, where you can alter any of the fields that you entered earlier. When you return to the data collection window, chose “Begin coding” from the Data Entry Menu or on the keyboard.

If the parameters are unchanged, you can begin the new data sample directly, by selecting “Begin coding” from the Data Entry Menu or .

Edit data file.

Selecting this option (also available in the top window) will load the current data file from disk. You are in a standard, but basic, Windows text editor, with the usual cut, copy, paste, and find capabilities. “Undo” will return you to the last saved copy of the file.

Here, and with the session notes editor, below, the keyboard often won’t respond when the file is first opened -- the cursor will be stuck at the beginning of the text. This is a system bug in True Basic; but a mouse click or touch pad tap anywhere in the text restores keyboard input to normal.

Placing comments in data files. You may place comments at the end of data lines (for instance, to note where codes have been corrected). Any text entered after an exclamation mark <!> or a pound-sign <#> will be ignored. Comments must always occur at the end of data lines, headers, or end-of-trial markers (“ -1”) – other uses will generate errors in the analysis programs.

Be sure to Save any changes before you Exit. *The text editor will be closed automatically at the beginning of the next data collection session*, in order to prevent over-writing new data with the data displayed in the text editor.

Make session notes.

Selecting this option opens a file named “[Case ID] SESSION NOTES.TXT”, with header information for the focal sample just ended placed at the end of the file. Again, you are in a standard, but basic, Windows text editor, with the usual cut, copy, paste, and find capabilities, and the same system bug noted above, which requires a mouse click or a touchpad tap. In contrast to the data editor, text will be wrapped at the end of the line.

Because this file is a text file, you may view your notes later using any word or text processor.

This window will close automatically at the beginning of the next data session, so

be sure to save any changes when you are done.

Edit behaviour or person codes.

When you click into either of the code windows, you will again find yourself in a standard, but basic, Windows text editor. You may change or add names or behaviours as you wish (remember to leave at least one space between the number and the code.). You may even replace the current file with your default person codes (*File- load default codes*). The program will be updated when you Save the modified file. Be sure to use the menu for the window in which you are working.

Quit.

To leave the program altogether, choose *Exit* from the *File* menu in the top window.

A sample data file

The program will create a file using the subject ID you provided: [Case ID] DATA.TXT. Output files are text files. A sample, with comments, follows:

```

Case 9127. Session 1a. 30 MAY 2001 at 9:31 a.m. Trial 1, Focal person 2. Observer Bill
Context = playground, morning, city
3 02 3 4.3 4.3
2 20 3 5.4 9.8
3 27 2 11.6 21.4
3 02 3 3.1 24.4
2 20 3 1.9 26.3
3 20 2 3.6 29.9
3 20 2 15.6 45.5
*
*
*
3 05 3 33.8 395.3
3 04 3 3.5 398.9
3 02 3 7.0 405.9
3 78 2 7.3 413.1
3 05 3 196.1 609.2
-1
Case 9127. Session 1a. 30 MAY 2001 at 9:45 a.m. Trial 2, Focal person 2. Observer Bill
Context = playground, morning, city
2 04 2 9.9 9.9
2 20 9 2.2 12.1
9 20 2 3.1 15.2
2 04 2 90.9 106.1
2 14 3 3.5 109.6
*
*
*
2 78 3 3.9 585.8
3 78 2 3.5 589.3
2 20 3 3.7 593.0
3 02 2 2.8 595.8
2 02 3 7.4 603.3
2 20 3 4.2 607.4
-1

```

Summary

To use FOCAL, you MUST:

- **Supply background information** for the data file header: Subject ID; Session; numeric code for the Focal person; length of the current observation sample, in minutes; observer name or ID; Contexts. Contexts are optional, but they can be used by the other programs in this suite to include or exclude trials for analysis. All values entered become defaults for subsequent trials and sessions. They can be changed by clicking on the *Analysis - Set Header Information* menu.
- **Have a set of numeric codes** for the behaviours and persons that you will be coding. If these are stored in the active directory using the default file names, FOCAL will load them automatically and display them on screen once it knows the subject ID for the current session. Otherwise, you must Open these files in the appropriate windows yourself.
- **To begin collecting data, click on Data Entry - Begin Coding, or choose the keyboard shortcut when the Data Entry window is highlighted.**

During data collection, the following keys are active:

- The numeric keypad, if it is separate; or the numeric keypad overlapping the letter keys on the right-hand side of a standard QWERTY keyboard. (Sorry, this isn't implemented (yet) for Dvorak keyboards).
- **<Esc> or <Q> to Save and Quit data collection**
- <F5> to Pause data collection. (Resume data collection by entering new data).
- <F9> to start or stop Quick Edit mode.
- <S> to save current data and continue collecting data.
- <e>, <?>, or </> to drop an error flag into the data file.
- <;> clears the last actor or behaviour code entered.
- <'> clears the current line.

The menus for the data collection window are active during data collection.

When using FOCAL, you MAY:

- **Edit your codes for behaviours or persons *before* you start collecting data.** (If you do make changes, be sure to Save them.) *Do not try to edit codes when data collection is active!*
- **You may change the working directory** by clicking the “Change directory” button in the setup box or the *Preferences* menu . Output files will be placed here. Note that if you change directories during a session, part of your data file will be in the first directory and part in the second, as Focal writes data to disk at the end of every focal sample (trial).
- **Make notes** about the trial just completed (*File - Open Session Notes*). Be sure to save any changes to this file! It will be closed when data collection begins.
- **Edit the data file for** the trial just completed (*Data Entry - Edit Data File*; or *File – Edit current data file*). Be sure to save this file if you change it! It will be closed when data collection begins.
- **Start or stop Quick Edit mode** from the Data Entry Menu.

Remember:

- Files with Behaviour Codes and Person Codes can have only one number and code per line, with the number and code separated by at least one space. The code may consist of several words and spaces, as in the sample code files shown in the Manual.
- If your taxonomy has only codes numbered 0 to 9, FOCAL will disregard the leading zeros shown in the code files. However, *if any code has two digits, then all codes must have two digits* when entered during data collection.
- FOCAL will ignore blank lines in your code file and lines with numbers only but no codes.
- Although the sample code files have all numbers from 0 to 99, you may delete unused numbers, if you wish, to make your code files easier to read.

SCAN: Checking and summing FOCAL data

Overview

SCAN reads data files produced by FOCAL, reporting the number of focal samples for each case (file), total number of events, total time observed, and the number of days across which observations were made.

SCAN automatically checks for a number of errors that may result from editing data files. It looks for error flags (which must be removed before running KAPPA or LAG). It checks that data lines have the required number of elements and that durations and cumulative times total properly³, and that End-of-Trial markers (" -1") are present. It notes the presence of duplicate lines (that is, adjacent lines with the same actor-behaviour-target codes). SCAN removes any blank lines that have been inserted in the file, and it rejoins header lines that have been split. (This may happen if the data file was edited using a word processor with too-narrow margins: the header is 'wrapped' on to the next line.) SCAN also checks for nonprinting characters (i.e., ASCII Text characters with values less than 32), replacing them with spaces. (If a tab character, for example, occurs at the end of a data line, it will be invisible – but it will cause the analysis programs to malfunction.) If SCAN encounters these errors, it will write a corrected data file. The original file will be saved with a .BAK extension.

Other checks are optional, and will be described below.

Getting Started

From the Files Menu, click on 'Select Files for Scanning'. Use the Open Files dialogue box to find the drive and directory with your FOCAL data files. Choose the first one to scan. A list of files will appear, with the first file already selected. If you want to scan only this file, click the 'OK' button. Alternatively, you may select multiple files to scan; click 'OK' when done. (Sorry for this 2-step process; it is a limitation of the programming language that I hope to remove in a subsequent

³ Note that when a defective line is encountered (e.g., if there are too few elements or non-numerical characters are present as a result of editing), SCAN is unable to read duration and cumulative time. It will thus generate two error messages: one for the defective line itself, and another indicating an error in duration or cumulative time for the line following the defective line. This second error message may or may not be appropriate.

upgrade.) You will be given a chance to save this list in a file, which you can use in later work sessions.

By default, only files with the same extension as the first file are presented for selection. If you want a list of all files to choose from, click *Select from all files* in the *Options* Menu.

In subsequent analyses, you may choose a file set you defined earlier and saved by choosing *Select a previous list of files* in the *Files* Menu, or by Opening this file when viewing the files to be analysed (*View Menu – Files to be analysed*). If you make any changes in your list, be sure to Save them – this will make them active. If you wish to re-scan only those files that had errors, choose *Select files with errors* in the *File* Menu.

Begin the scan by clicking *Start Analysis* in the *Analysis* menu. When it is complete, you may see the details by clicking *Results* in the *View* menu. Included in the output is an alphabetized list of all contexts noted in the data files. (Note that if you used a comma in a context name, the name has been split and treated as two different contexts. *Do not use commas when naming contexts.*) If one of the listed contexts is misspelled, you can locate it by choosing *Find Context* from the *Edit* Menu. In the sub-menu, selecting *Search Files Most Recently Analysed* will cause SCAN to search the last set of files analysed. If you want to specify a different set of files, choose *Search Other Files*. (In fact, you can use this option to search for any word or phrase in any set of text files.)

If you want to review these results after leaving SCAN, they are kept in a file named SCAN REPORTS.TXT, located in the same directory as the scanned data files. If you wish, you may rename this file while viewing it in SCAN – simply Save it with the new name. Results from subsequent runs will be placed in SCAN REPORTS.TXT, not the renamed file.

Results can also be sent to the printer (File menu – Print). If your system has more than one printer installed, you will be able to choose among them at the printer setup dialogue box. (Note that trying to Cancel the print job at this point will result in a system error, crashing the program. The problem is in the programming language; I hope to have it corrected at some future point.)

If SCAN found errors, you may see these details by clicking 'Error Reports' in

the *View* Menu. If you want to review these results after leaving SCAN, they are kept in a file named SCAN ERROR REPORTS.TXT, located in the same directory as the scanned data files. As above, you may Save and rename this file while viewing it in SCAN, and subsequent output will be placed in SCAN ERROR REPORTS.TXT.

If, after viewing the error reports, you want to edit one of your data files, select *Open Data File for Editing* from the *File* Menu in the main window.

If you wish to replace (overwrite) the output in SCAN REPORTS.TXT and SCAN ERROR REPORTS.TXT, choose this in the *Options* Menu. (A checkmark indicates the active option.)

Options in the *Analysis* menu disable some of the error-checking done by SCAN. I do not recommend this; but you may find it useful if you are working with data files created by earlier versions of FOCAL. Note that if you disable 'Check Durations and Time', time budget analyses may not be accurate. Because of the way FOCAL records time, durations for individual events should sum exactly to the total durations reported for the trial (apparent small discrepancies are due to rounding error). Thus time discrepancies indicate that events have been edited out of the data file.⁴

If you are collected focal-individual data, the focal individual should be present in every interaction, either as the initiator or target (or both, if engaged in solitary behaviour; Altmann, 1974). By default, SCAN checks for the presence of the focal individual and reports data lines where the focal individual is absent. You can disable this in the *Analysis* menu. (If 'stop' codes are present in your data, you should enter these (*View* menu – *Stop Codes*), as they replace actor-target codes – see page 57.

The directory you are working in, the files that you have used, and the options that you have chosen, will be saved in a configuration file, and will be loaded the next time you run SCAN. (Please do not erase this file, unless you are uninstalling the

⁴ If this has happened, you will notice that the total time reported in SCAN REPORTS.TXT will not match the sum of the individual file times. The time totalled at the bottom of the report is the sum of all individual event durations, whereas time credited to individual files is the sum of 'cumulative time' at the end of trials. If the sum of the individual file times does not equal the given total time, data may be missing from your files.

programs. It will be used by other programs in this suite, as well as by SCAN).

Summary

To run SCAN, you MUST:

- **Choose the files to be examined.**
 - Click on *File Menu - Select new files for scanning*. Use the dialogue box to find the drive and directory with your FOCAL data files. Choose the first one to analyse. Click *Open*. A list of files will appear, with the first file already selected. Select one or more additional files to analyse; click *OK* when done.
 - You may also select a previously saved list (*File Menu - Select a previous list of files*).
 - The list of files that you last used is loaded by default when you start the program.
- Start the analysis. (*Analysis Menu - Start Analysis*)

You MAY:

- Disable error-checking for event durations and cumulative time (*Analysis Menu - Check durations and time*)
- Disable error-checking for the presence of an observer name (*Analysis Menu - Check for observer in Headers*)
- Disable error-checking for duplicate lines (*Analysis Menu - Check for duplicate lines*)
- Disable error-checking for the presence of the focal individual (*Analysis Menu - Check that focal person is always an actor or target*).
- Select files with the same extension as the first file chosen, or you may select from a list of all files (*Options Menu - Select from all files*).
- Select only files that had errors in the previous Scan (*File Menu - Select files with errors*)
- Select files used in a previous Scan (*Files Menu - Select a previous list of files*).
- Have output from the current analysis either replace earlier output or be appended to it. (*Options Menu*). A checkmark tells you which option is currently active.
- View or print results of an analysis you have completed (*View Menu - Results*).

Included in the output is an alphabetized list of all contexts noted in the data files. If one of these is mis-spelled, you can locate the its file (*Edit Menu - Find Context*).

- View, edit, or print a data file (*File Menu- Open data file for editing*).
- View or print errors identified by an analysis you have completed (*View Menu - Error Reports*).

KAPPA: Reliabilities for FOCAL data

Overview

KAPPA assesses observer reliability by examining a set of FOCAL data files two at a time and computing for each pair and for the entire set:

- percent agreement for each code and the entire set of codes.
- kappa, its standard error, and 95% confidence interval.
- partial kappa for each individual code.⁵
- Reliability of event durations is assessed by a matched *t*-test and correlation across observers, with problematic events identified by standardized residuals.
- Trimmed tests are also computed, so that the influence of out-lying cases can be evaluated.
- If more than two files have been specified, then all statistics are provided for the set of files as a whole, as well as the individual pairs of files. Mean kappa is provided as an index of integration across multiple coders (Conger, 1980), as well as pooled kappa.
- Taxonomies of up to 100 behaviour categories and 100 actor-target codes can be analysed.
- Codes can be combined without altering the original data. After one analysis, the recoding file can be edited prior to its use in another analysis. This recoding file can be used for later sequential (lag) and time-budget analyses.

⁵ This is modelled on the formula for the standard table-wise kappa. Let *i* = an element of the diagonal of the Table of Comparisons (that is, the number of agreements for code *i*). Then $\text{kappa}(i) = \text{Table}(i) - \text{expected}(i) / (1 - \text{expected}(i))$, where $\text{expected}(i) = (\text{row total}(i) / \text{Grand Total}) * (\text{column total}(i) / \text{Grand Total})$.

As I note later, when code *i* is relatively rare, its expected value will be close to zero – observers are unlikely to agree by chance – and the partial kappa will be very close to the percent agreement for that code. Consequently, partial kappas are only useful for the most frequently occurring codes in your taxonomy.

Please note that I am not aware of any formal justification for computing kappas for individual codes, nor for the formula that I have used to do so. If you know of any relevant published discussions, I would be grateful if you sent me the references at wlroberts@tru.ca.

- Results are sent to text files which can be viewed at the end of a run, or later with any word processor.
- As an aid in training observers and a guide in evaluating the reliability of infrequently occurring behaviours, KAPPA can report (in separate output files) the events it has matched and the agreements it has found.
- The program lists up to five other codes most frequently confused with each code, as a guide for recombining codes to obtain satisfactory reliabilities.
- Because observers coding in real time will necessarily be slightly out of synchrony even when they agree about which codes to use, KAPPA allows the user to specify how much time discrepancy to tolerate when matching individual events or searching for sequences of identical events.

Running the program

Before running KAPPA, FOCAL data files should be examined by SCAN, to insure that error flags have been removed and that editing has not introduced errors.

Selecting data files to be analysed.

KAPPA requires at least two data files. However, it is possible to specify multiple data files (say, all reliability files for a certain point in time, or all reliability files for a given project), in order to assess reliability across observers or across time. When specifying multiple files, you will be presented with a list of the files that you have chosen. You will need to inspect this list to be sure that files are paired correctly (copy, cut, and paste are available). The altered list can be saved for future use.

Creating a new list of files. From the *File* Menu in the main window, click *Select new files for analysis* and then select one of the three options. All options begin with an Open Files dialogue box, which you can use to navigate to the drive and directory with your FOCAL data files. (Kappa will remember this location.) Choose the first file to analyse. How the next file is chosen depends on which option was selected.

If *Select two files* was chosen, a second Open Files dialogue box opens immediately and you can select the second file, which will be compared with the first. Both files must be in the same directory.

If you would like to save the file names for later use, go to the *View* menu and click on *Files to be analysed*. The names of the two files you just selected will be listed

in a new window. Chose *Save* in the window *File* menu.

If *Select multiple files* was chosen, a list of files will appear, with the file chosen from the Open Files dialogue box already selected. Select one or more additional files to compare, using the Shift or Ctrl keys⁶. Click 'OK' when done. You will be given a chance to save this list in a file, which you can use in later work sessions.

After deciding whether or not to save the file list, you will be presented with the list in a simple text editor. Inspect the list to be sure that files will be compared as you wish. **The first and second files on the list will be compared, the third and fourth, and so on.** Pairs of files can be separated by blank lines to improve readability. You may cut, copy, and paste, to rearrange or extend the list. For example, if three observers have coded the same set of interactions, the names of their three data files will need to be copied and rearranged to make all possible comparisons – observers A and B, B and C, and A and C. If you toggle on 'Quick Move' in the Edit Menu, you can cut and paste simply by clicking on the line you want to move, and then the place where you want it to be.

If you make any changes in the list of files to be analysed, be sure to *Save* it when you are done, in order to make the changes active in the program.

When *Select multiple files of the same type* is chosen the list of files will include only those files with the same extension as the first file chosen. (This is convenient if you have named your files using a template like *observer.case_id*). As before, you will be given a chance to save this list in a file, and then to check and correct the list in a text editor, as just described.

Using an earlier list of files. If you saved a list of files in an earlier session, it will loaded by default when you next start KAPPA.

You may select a different saved list of files in two ways. (1) from the *File* Menu choose *Select a previous list of files*. (2) From the *View* menu, click on *Files to be analysed* and then in the new window, choose *Open* from the *Files* menu. If you make any changes in the retrieved list, you must *Save* it in order for the changes to become

⁶ The usual Windows conventions apply. Using the mouse and the Shift key allows you to select contiguous file names; holding down the Ctrl key allows you to select file names anywhere in the list.

active. (You may choose a new file name for the altered list, if you like.)

View Menu

Results. When the analysis is complete, you can view (and edit) the results by clicking *View Menu - Results*. If you wish, you may *Save* them under a different file name. Subsequent analysis will still be sent to KAPPA REPORTS.TXT, the default output file. Output will be appended to the end of this file, if you toggle on *Append results to output file* in the *Options* Menu in the main window (this item has a check mark by it when it is active). When *Save only current results* is toggled on, current results will replace (overwrite) previous output.

Be aware that it is easy to generate a MB or more of output if you choose to examine a range of time tolerances. The Results text editor in KAPPA can only handle small files with ease. Therefore you will want to examine large output files with your own word processor or with a text editor like NoteTab or Kate.

Sample results for Kappa are shown beginning on the next page.

Sample output from KAPPA

Reliabilities: Percent agreement and kappa
 Friday, September 19, 2003 1:00 p.m.

Data from file JANICE.101, collected 27 JUL 1989. Observer = Janice
 Data from file LISA.101, collected 27 JUL 1989. Observer = Lisa

Reliabilities calculated over 51.6 minutes across 5 trials
 Permitted error in timing when matching events: 10 seconds

Table of Comparisons with marginal totals:

Rows: JANICE.101
 Columns: LISA.101

Code	0	1	3	9	10	11	13	16	20	25	26	31	42	45	50	62	63	70	
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12
3	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28
9	0	0	1	19	0	0	0	0	0	1	0	0	0	0	0	0	0	0	21
10	0	0	0	0	30	0	0	0	0	1	0	0	0	0	0	0	0	0	31
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	1	0	5	0	0	0	0	0	0	0	0	0	0	0	6
16	0	0	0	0	0	0	0	20	1	2	0	0	0	0	0	0	0	0	23
20	0	0	1	1	0	0	0	0	64	3	0	0	0	0	0	0	0	0	69
25	0	0	0	0	2	1	0	0	2	135	2	0	0	0	0	0	0	0	142
26	0	0	0	1	0	0	0	1	1	1	6	0	0	0	0	0	0	0	10
31	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	11
42	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
45	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	4
50	0	0	0	0	0	0	0	0	0	1	0	0	0	0	3	0	0	0	4
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	4
63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
70	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	3	5
	0	12	30	22	33	1	5	21	68	146	8	11	1	4	3	4	1	3	373

(Example continues)

The optional Table of Comparisons has been printed, along with marginal totals. Values along the diagonal indicate observer agreement; off-diagonal elements indicate disagreements.

Comparisons = agreements + average disagreements across coders.

Code	Agreements/ Comparisons	Proportion agreement (kappa)	Most frequent disagreements: code (frequency), code (frequency)				
0	0 / .5	.000 (.000)	9 (1)				
1	12 / 12	1.000 (1.000)					
3	28 / 29	.966 (.965)	20 (1)	9 (1)			
9	19 / 21.5	.884 (.883)	26 (1)	25 (1)	20 (1)	3 (1)	0 (1)
10	30 / 32	.938 (.937)	25 (3)	13 (1)			
11	0 / .5	.000 (.000)	25 (1)				
13	5 / 5.5	.909 (.909)	10 (1)				
16	20 / 22	.909 (.909)	25 (2)	26 (1)	20 (1)		
20	64 / 68.5	.934 (.932)	25 (5)	26 (1)	16 (1)	9 (1)	3 (1)
25	135 / 144	.938 (.927)	20 (5)	26 (3)	10 (3)	70 (2)	16 (2)
26	6 / 9	.667 (.666)	25 (3)	20 (1)	16 (1)	9 (1)	
31	11 / 11	1.000 (1.000)					
42	1 / 1	1.000 (1.000)					
45	4 / 4	1.000 (1.000)					
50	3 / 3.5	.857 (.857)	25 (1)				
62	4 / 4	1.000 (1.000)					
63	1 / 1	1.000 (1.000)					
70	3 / 4	.750 (.750)	25 (2)				

Disagreements are averaged over both observers. For code 16, for example, there were 4 disagreements, 3 by one observer and 1 by the other (see Table of Comparisons on the preceding page).

 Total agreements / total comparisons = 346 / 373
 Proportion agreement = .928
 Kappa = .909
 Standard Error of Kappa = .022
 Approximate 95% confidence interval = .864 to .954

Because many of the individual codes are relatively rare, the probability that observers will agree by chance is extremely small. Therefore the individual kappas are very close to the proportion agreements.

Analysis of event durations

Pooled t(17) = .12, p = .909. Variance accounted for by coders: .001
 Trimmed pooled t(15) = -.215, p = .833. Variance accounted for by coders: .003

The t-test is for coder differences; therefore, a small effect size for t indicates coder similarity across the set of event durations.

(Example continues)

For event durations, $r(16) = .999$, $p = .000$

Trimmed $r(14) = .997$, $p = .000$

A large correlation indicates coder similarity across the set of event durations.

Total durations for each event code, in seconds and as a proportion of total time.

A large standardized residual indicates coder differences for that particular event.

Code	Janice		Lisa		Standardized Residuals	
	Seconds	Proportion	Seconds	Proportion	whole set	trimmed set
0	5.3	.002	.0	.000	.05	.05
1	102.9	.033	105.9	.034	.43	.38
3	563.4	.182	584.4	.189	.95	.59
9	106.9	.035	114.8	.037	.83	.80
10	131.4	.042	123.1	.040	.57	.68
11	.0	.000	12.3	.004	1.43	1.49
13	29.1	.009	27.0	.009	.16	.16
16	104.5	.034	72.6	.023	2.49	2.65
20	1,106.9	.357	1,139.9	.369	.78	
25	745.7	.241	742.1	.240	1.50	
26	38.4	.012	33.2	.011	.12	.14
31	65.7	.021	46.0	.015	1.39	1.48
42	4.7	.002	4.9	.002	.41	.43
45	25.2	.008	22.9	.007	.15	.15
50	11.0	.004	14.8	.005	.69	.72
62	36.1	.012	30.3	.010	.16	.18
63	4.8	.002	4.6	.001	.37	.39
70	14.8	.005	11.1	.004	.06	.06
	3,096.8		3,089.9			

Finding a given code in all data files. If you wish to examine a certain code in your data files (perhaps to understand why percent agreement was low) you can locate all occurrences by

Standardized residuals are similar to z scores. We ignore their sign; only absolute values are tabled.

The program trims the upper and lower 5% (durations with z-scores exceeding ± 1.645). We wish to see how much these extreme values influence t and r . (In the current example, we conclude that they are not influential – the values for t and r are stable.) Which values were trimmed is apparent in the table of residuals: because of their long durations, codes 20 and 25 were excluded).

choosing “Find Across Files” from the Edit Menu. By default, Kappa will search the set of files just analyzed. If you would like to search a different set of files (for example, the matched event files, if you have generated them), you can specify them from the File Menu (Select Files) or the View Menu (Files to be Analyzed). Results of the search are appended to the end of the results

file that you are viewing.

Matched Events. If you clicked on *Report Matched Events* in the *Analysis Menu* (it will have a check mark by it), you will be able to view them by clicking on *View Menu - Matched Events*. Examining these files makes it obvious when coders have disagreed. They are, therefore, valuable tools for training. They are also very helpful when examining the reasons for problematic reliabilities of infrequently occurring codes, and for identifying agreements that the program has overlooked. (In my own data files, Kappa identifies better than 95 percent of all possible agreements on average; but the overlooked agreements may be critical for infrequently occurring codes. In any case, you will want to examine the sequence files to ensure that the time tolerances and matching technique that you have chosen are producing reasonable results.)

In the sequence files, matched events (that is, events used by Kappa to synchronize comparisons across the two files) will be indicated by "T 1", "T 2", etc., if you matched by time, or "E 1", "E 2", if you matched by events. Sequences of events are marked with "S1", "S2", etc. In all cases, matched events have the same actor, behaviour, and target codes in both files. (A more detailed description of how the program matches events is given in a later section of this chapter.) Finally, as shown in the sample output on the next two pages, agreements (based on comparing behaviour codes only) are marked "A 1", "A 2", etc.

Sample output I: Focal data files matched by time

Tuesday, October 7, 2003 6:05 p.m.
 Tolerance when matching events: 5 seconds.
 Family 3, Home Session 2. 09 FEB 1989
 at 19:16 Trial 2, Focal person = 2,
 Observer = Kelly

2	02	0	2.7	2.7	[T 1]	A 1
3	20	2	1.3	4.0	[T 2]	A 2
2	20	3	2.9	6.8	[T 3]	A 3
2	02	0	14.5	21.4		A 4 A 5
2	20	0	6.6	28.0	[T 4]	A 6
0	22	2	7.8	35.8		
2	20	5	5.8	41.6	[T 5]	A 7
5	27	2	1.8	43.3	[T 6]	A 8
2	02	0	3.6	47.0	[T 7]	A 9
2	20	0	4.1	51.0	[T 8]	A 10
3	20	2	4.2	55.2	[T 9]	A 11

-1
 Agreements / comparisons = 11 / 14.
 Percent Agreement = 78.6

Summary of individual event codes for this trial:

code	hits / comparisons	mean	percent agreement
2	4 / 5		80.0
20	6 / 7		85.7
22	0 / 1		.0
27	1 / 1		100.0

End of summary.

The 11 agreements have been marked (A1 to A11), as well as 9 events used to synchronize comparisons across the two files.

Tuesday, October 7, 2003 6:05 p.m.
 Tolerance when matching events: 5 seconds.
 Family 3, Home Session 2. 09 FEB 1989
 at 18:24 Trial 2, Focal person = 2,
 Observer = Anne

2	02	0	3.0	3.0	[T 1]	A 1
3	20	2	1.0	4.0	[T 2]	A 2
2	20	3	2.3	6.3	[T 3]	A 3
3	02	2	7.1	13.4		A 4
2	02	3	7.9	21.3		A 5
2	20	0	2.9	24.2	[T 4]	A 6
3	20	2	3.0	27.2		
2	02	0	11.9	39.1		
2	20	5	2.1	41.2	[T 5]	A 7
5	27	2	2.0	43.2	[T 6]	A 8
2	02	0	3.6	46.9	[T 7]	A 9
2	20	0	2.8	49.6	[T 8]	A 10
3	20	2	2.0	51.6	[T 9]	A 11
2	02	0	3.5	55.0		

-1
 Agreements / comparisons = 11 / 14.
 Percent Agreement = 78.6

Summary of individual event codes for this trial:

code	hits / comparisons	mean	percent agreement
2	4 / 5		80.0
20	6 / 7		85.7
22	0 / 1		.0
27	1 / 1		100.0

End of summary.

Sample output II: Focal data files matched by event sequences and matched events

Tuesday, October 7, 2003 6:30 p.m.

Event sequences and matched events were used to make comparisons across files. Tolerance at sequence onset: 20 seconds.

Family 3, Home Session 2. 09 FEB 1989 at 19:16 Trial 2, Focal person = 2, Observer = Kelly

2	02	0	2.7	2.7	[E 1]	A 1
3	20	2	1.3	4.0	[E 2]	A 2
2	20	3	2.9	6.8	[E 3]	A 3
2	02	0	14.5	21.4		A 4
2	20	0	6.6	28.0	[E 4]	A 5
0	22	2	7.8	35.8		
2	20	5	5.8	41.6	[S 1]	A 6
5	27	2	1.8	43.3	[S 1]	A 7
2	02	0	3.6	47.0	[S 1]	A 8
2	20	0	4.1	51.0	[S 1]	A 9
3	20	2	4.2	55.2	[S 1]	A 10

-1

Agreements / comparisons = 10 / 14.
Percent Agreement = 71.4

Summary of individual event codes for this trial:

code	hits / comparisons	mean	percent agreement
2	3 / 4.5		66.7
20	6 / 8		75.0
22	0 / .5		.0
27	1 / 1		100.0

End of summary.

In addition to agreements, 4 matching events and one 5-event sequence have been marked. A summary follows the end of the trial. Note that in this case, the event algorithm produced only 10 agreements, whereas the time algorithm produced 11.

Tuesday, October 7, 2003 6:30 p.m.

Tolerance at sequence onset: 20 seconds. Event sequences and matched events were used to make comparisons across files.

Family 3, Home Session 2. 09 FEB 1989 at 18:24 Trial 2, Focal person = 2, Observer = Anne

2	02	0	3.0	3.0	[E 1]	A 1
3	20	2	1.0	4.0	[E 2]	A 2
2	20	3	2.3	6.3	[E 3]	A 3
3	02	2	7.1	13.4		A 4
2	02	3	7.9	21.3		
2	20	0	2.9	24.2	[E 4]	A 5
3	20	2	3.0	27.2		
2	02	0	11.9	39.1		
2	20	5	2.1	41.2	[S 1]	A 6
5	27	2	2.0	43.2	[S 1]	A 7
2	02	0	3.6	46.9	[S 1]	A 8
2	20	0	2.8	49.6	[S 1]	A 9
3	20	2	2.0	51.6	[S 1]	A 10
2	02	0	3.5	55.0		

-1

Agreements / comparisons = 10 / 14.
Percent Agreement = 71.4

Summary of individual event codes for this trial:

code	hits / comparisons	mean	percent agreement
2	3 / 4.5		66.7
20	6 / 8		75.0
22	0 / .5		.0
27	1 / 1		100.0

End of summary.

The default sequence filename is based on the name of Focal data file. For example, data file *9127.dat* will have its sequences placed in *9127 Sequences.txt*. If you wish, you may Save them under a different file name. Subsequent analysis will still be sent to the default output files. New output will be appended to sequence files

if you have chosen ‘Append results to output file’ in the Options Menu, above. (Using the ‘Set time tolerance’ option in the Analysis Menu, it is possible to specify multiple analyses on the same set of data files.)

Recoding (combining) behaviours. If you want to combine (recode) some of your behaviour codes, click *View Menu - Recoding File*. KAPPA will attempt to load the default behaviour code file used by FOCAL; if it is not present, you will be asked to select the file that has your behaviour codes in it. (You will have such a file from using FOCAL.) You are now ready to convert your list of codes into a list of recodes. When you are done, KAPPA will save it under a new name ("Recoded Behaviours.txt" – or you can save it under another name of your choice), leaving your behaviour codes file intact.

A recoding file is like a simple, two-level outline: codes which are sub-topics

00 talks, neutral
01 talks, positive
02 talks, sad
03 talks, angry
04
05 ignores; no response
10 solitary, neutral
11 solitary, positive
12 solitary, sad
13 solitary, angry
20 parallel play, neutral
21 parallel play, positive
22 parallel play, sad
23 parallel play, angry

will be re-coded as the topic (or ‘parent’) code. In the example in the box, behaviours were originally crossed with emotions to make a large number of distinct categories for observation; now we wish to combine the ‘neutral’ and ‘positive’ codes for ‘talks’ and ‘solitary activity’ and, because of low frequency, we wish to combine all varieties of parallel play into a single category. When recoding is activated (*‘Do combine codes’* in the *Analysis* Menu), KAPPA will read this recoding file and replace codes of 01 with 00, 11 with 10, and 21, 22, and 23 with 20. If any adjacent actor-behaviour-target events are identical after recoding, the program will merge them into a single event lasting for their combined duration.

As the example indicates, ‘parent’ codes without ‘children’ (topics with no sub-topics) are permissible, as are blank lines and lines with numbers but no codes.

You may use Cut, Copy, and Paste to arrange and indent your list (2 or more

spaces). If you activate Quick Move, you can cut a line (code) by clicking once on it. When you click again, it will be pasted and indented.

When you are satisfied with your recoding file, be sure to Save it before leaving the editor. You may use the default file name or another of your choice; KAPPA will remember it.

Don't forget – when you want the program to use the recodes you have specified, choose 'Do combine behaviour codes' in the Analysis Menu.

Files to be analysed. As mentioned earlier, click here to edit the list of files currently chosen, or to load a list of files previously saved.

Analysis Menu

Begin analysis. When you are ready to start the analysis, click here.

Set time tolerances. Kappa offers two different methods of matching events across files – time, or sequences of identical events (see page 39). Here you can choose the timing discrepancy (or tolerance) that will be used by each method. For the time method, the tolerances indicated here will affect both the search for identical, matching events in the two files, as well as the comparison of behaviour codes. For the sequences method, the tolerances specified here are used when searching for sequence onsets, that is, the first event in a sequence of events. When one method is being used, tolerances for the other method are ignored.

If you wish to change the default values (10 seconds for both methods), click 'Set time tolerances' in the Analysis Menu. In the screen that appears, you may specify a range of tolerances to be applied to your data. For example, entering *4 to 10 step 2* in the set of boxes will cause the program to do 4 analyses, applying tolerances of 4, 6, 8, and 10 seconds. If you wish to specify only a single tolerance (say, 10 seconds), enter it in both boxes (*10 to 10 step 2*). The step value will be ignored.

When you are satisfied with the values you have set, click 'OK'.

The default values may not be appropriate given the type of data you have collected and the research questions that interest you. In general, smaller tolerances result in fewer matching events being identified, more comparisons (and therefore more disagreements), and lower values of kappa. However, large tolerances can also produce very poor reliabilities, especially if one observer uses more codes than the other. In deciding on appropriate values, you should consider how critical time is to

the research questions that concern you, and the average time differences that actually exist between your two coders. Results should be checked by examining the matched event files.

Set parameters for event-matching. These parameters all affect how matching events and sequences are identified.

- *Minimum Length* specifies the minimum number of identical actor-behaviour-target events needed to identify a sequence. The default value is 4.
- *Search-distance* sets how many events ahead the program will search for a matching event. Values of two or three usually give good results.
- *Events For Match:* This is the number of events needed to identify a matching event. For example, if this parameter is set to 2, the program will look for two-event sequences and designate the first event in the sequence as a matched event. If it is set to a value greater than or equal to *Minimum Length*, then no matched events will be found, only sequences. If this parameter is set to 1, the program may not make appropriate matches across files; but you can check this by examining the matching event files (see View Menu, above).

The impact of these parameters is explained more fully in the next section.

Combine behaviour codes. Click here to have the program combine (recode) behaviour codes, following the recoding instructions that you gave in the View - Recode Behaviour File menu (above). When this option is active, the item will be checked.

Use row totals for percent agreement. If you click on this (a checkmark will appear), then the row total – all occurrences of a given code by a single observer – will be used to calculate percent agreement for that code. An important drawback of this method is that the row total will differ depending on which file is arbitrarily designated to form the rows in the Table of Comparisons. An additional difficulty is that this method only notes a disagreement when the first observer uses a given code and the second observer does not. It ignores disagreements that occur when the second observer uses the code and the first observer fails to.

Use mean comparisons for percent agreement. This alternative method avoids the difficulties just noted by averaging the number of disagreements across the two coders, so that the number of comparisons = agreements + mean disagreements. By taking into account all disagreements, percentage agreement for individual codes

remains the same regardless of the order in which files are named for analysis.

Use time to coordinate comparisons. When you select this alternative, cumulative time is used to coordinate comparisons across the two files and to match behaviour codes for agreement/disagreement. The discrepancy (in seconds) to be tolerated across the two files can be set by the user (see Setting tolerances, above).

Use event sequences and matched events to coordinate comparisons. When you select this alternative, event sequences and matched events are used to coordinate comparisons across the two files. Because time is only used to coordinate the beginning of sequences, this method can produce high rates of agreement when time discrepancies are very large.

Options Menu

Append or replace results. You may choose whether output from the current analysis replaces earlier output or is appended to the end of the output file.

Report matched events. Click this option to have KAPPA record the sequences and events that it is able to match, with agreements. When this option is active, the item will be checked. After the analysis is finished, sequences can be viewed using the View Menu.

Show Table of Comparisons. When this item is active (check mark present), the Results file will contain the entire Table of Comparisons constructed by KAPPA. Kappa presents so much summary information from this table (percent agreement, codes most frequently confused) that it is usually not necessary to invoke this option.

How KAPPA compares events

The usual method for determining reliability is to compare judgments between two judges or coders, and compute both kappa and an "agreements divided by agreements + disagreements" statistic. If one's interest is primarily sequences of behaviour and the consequences and contexts of certain actions, then it is clearly correct to give priority to correctly coding what is happening, and to be less concerned about when it is happening (or who is doing it). Although the traditional approach in essence ignores the issue of time (coders are assumed to be in synchrony, or close enough for practical purposes), clearly a large lag between coders is a matter of concern. How much discrepancy should be tolerated is unclear, and probably varies

depending on the behaviours coded and the research questions of interest. You can specify the amount of tolerance the program uses when matching events from different judges, and you can choose between two methods, one of which emphasizes synchrony and the other of which minimizes it.

Using time to compare events. When using time, KAPPA selects events for comparison using two different rules:

(1) If cumulative time (the last value in the data line generated by FOCAL) is the same for both events (within the tolerance specified), the program moves to the next event for each coder. If, on the other hand, coder A is behind coder B, the program moves to coder A's next event and makes a comparison. This is repeated until the cumulative times are within the tolerance specified.

(2) Comparisons are synchronized at the beginning of matching events (marked with "T"s in the Matched Event Files). As above, if one file is behind the other, this file is brought forward one event at a time until both files are at the matched event.

This process continues until the end of the trial. If one coder reaches this before the other, the 'extra' events are paired by the program with the last event in the shorter trial. Thus files are also synchronized at the end of each trial.

Using time to select synchronization points (matching events). The program takes events one at the time from the smaller of the two trials being compared and attempts to find matching events from the larger trial, that is, events with identical actor, behaviour, and target codes. Only events that have occurred within *tolerance* seconds are examined in the larger trial. For example, if tolerance was set at 5 seconds and the event from the first file ended at 20 seconds cumulative time, only events whose cumulative times fell between 15 and 25 seconds would be examined in the second file. If more than one identical event is found in the larger file, then the one with the least time discrepancy is chosen.

Using sequences of events to make comparisons. When using matching events and sequences, KAPPA makes comparisons using two different rules:

(1) If neither of the two events is a synchronization point, the comparison is made and the program moves on to the next event in each file, regardless of cumulative time or whether the events agreed or disagreed.

(2) Comparisons are synchronized at the beginning of sequences (marked with

“S”s in the Matched Event Files) and at matching events (marked with “E”s in the Matched Event Files). If one file is behind the other, this file is brought forward one event at a time until both files are at the matched event. Once both files are synchronized, the program compares the next event in each file.

As with the time method, this process continues until the end of the trial. If one coder reaches this before the other, the 'extra' events are paired by the program with the last event in the shorter trial. Thus files are also synchronized at the end of each trial.

Using events to select synchronization points (matching events and sequences).

The program takes events one at a time from the smaller of the two trials being compared and attempts to find matching events from the larger trial, that is, events with identical actor, behaviour, and target codes. As with the time method, only events that have occurred within *tolerance* seconds are examined in the larger trial. If the program finds a match in the second file, it then sees how many identical events follow in each file. For these subsequent events, time is ignored. If a series of four or more identical events are found in each file, they are designated as a sequence. (This minimum number can be reset in the *Analysis Menu*.) If the program finds several initial matches in the second file, it chooses the one that generates the longest sequence.

If the program has found sequences of only two or three events in length, the first event in each file is designated as a matched event. (The reduced sequence length needed to designate a matched event can be set in the *Analysis Menu*.)

Matched events must also satisfy a second criterion: they must occur within *place + n* events in both files. For example, if $n = 2$ and the program is examining the third event since the end of the last sequence in the smaller file, a matching event in the larger file can be no more than five events ($3 + 2$) from the end of the last sequence. (The value is higher in the second file, because the larger file has “extra” events and we wish to look beyond these interpolated events – but not too far beyond – to find a match.) The value of n can be set in the *Analysis Menu*.

Which method is best? Which method works best depends on the characteristics of the data files being compared. If time discrepancies are large (a problem in my own data, as we have coded family and playground interactions in real

time, rather than from videotapes) then the event sequences method generally gives better results. On the other hand, when time synchrony is good, the time method can give more accurate comparisons, as it does in the sample output presented earlier. It is important to examine the matched event files to confirm that the method and parameters that you have specified are the most appropriate for your data.

Accuracy, false positives, and false negatives.

Based on my own research, in which I have used complex taxonomies which give ample opportunity for disagreement, KAPPA can find 96% or more of all agreements. Despite this high “hit rate”, the program will make mistakes when one observer has coded more events than the other, as in the example below.

Observer 1							Observer 2								
3	04	3	14.6	14.6	[S 1]	A 1	3	04	3	14.0	14.0	[S 1]	A 1		
2	24	3	7.5	22.1	[S 1]	A 2	2	24	3	11.6	25.6	[S 1]	A 2		
3	25	2	3.4	25.5	[S 1]	A 3	3	25	2	3.7	29.3	[S 1]	A 3		
3	04	3	9.8	35.3	[S 1]	A 4	3	04	3	5.8	35.1	[S 1]	A 4		
2	11	3	1.7	37.0	[S 1]	A 5	2	11	3	6.5	41.6	[S 1]	A 5		
3	27	2	5.6	42.6	[S 1]	A 6	3	27	2	2.8	44.4	[S 1]	A 6		
3	22	9	54.2	96.8	[S 1]	A 7	3	22	9	52.1	96.5	[S 1]	A 7		
3	14	2	7.9	104.8	[S 1]	A 8	3	14	2	6.3	102.7	[S 1]	A 8		
3	04	2	5.4	110.2	[S 1]	A 9	3	04	2	18.3	121.0	[S 1]	A 9		
2	20	3	1.6	111.8			2	03	2	2.1	123.1				
3	20	2	10.0	121.8			3	22	2	23.3	146.4	[E 1]	A 10		
2	03	3	2.9	124.7			2	20	3	4.6	151.1	[S 2]	A 11	A 12	A 13
3	22	2	21.6	146.3	[E 1]	A 10	3	20	2	5.2	156.2	[S 2]	A 14		
2	20	3	1.1	147.3		A 11	2	03	3	1.8	158.0	[S 2]	A 15		
3	20	2	1.6	148.9		A 12	3	22	2	11.6	169.6	[S 2]	A 16		
2	20	3	1.1	150.0	[S 2]	A 13	2	20	3	1.3	170.9	[S 2]	A 17		
3	20	2	6.6	156.6	[S 2]	A 14	3	20	2	6.0	176.9	[S 2]	A 18		
2	03	3	2.1	158.8	[S 2]	A 15	...								
3	22	2	10.6	169.4	[S 2]	A 16									
2	20	3	1.4	170.8	[S 2]	A 17									
3	20	2	2.0	172.8	[S 2]	A 18									
...															

In this example, Observer 1 has coded two extra pairs of code 20s (“talks”), one pair at the end of sequence 1 (at cumulative times 111.8 and 121.8), and another pair after matched event E 1 (cumulative times 147.3 and 148.9). Consequently, when observations were matched by sequences of events, the program fails to note an agreement for code 03 (cumulative time 124.7 in file 1 and 123.1 in file 2).⁷ Instead, the extra code 20 in the first file (at time 111.8) is compared with code 03 in the

⁷ Code 03 is not designated as a matched event (see discussion above) because the behaviours that immediately follow it are different in the two files.

second file (both events immediately follow sequence 1), while code 03 in the first file is compared with code 22 (E 1) in the second file.

In addition to this false negative for code 03, false positives are generated for the second pair of extra code 20s (at times 147.3 and 148.9). Because the program reaches the beginning of sequence 2 in the second file before it reaches sequence 2 in the first file, the extra code 20s in file 1 are compared with the first behaviour code in sequence 2 (cumulative time 151.1 in the second file), as described earlier. Because this code happens to be a 20, the extra code 20s in file 1 are scored as agreements even though there are no corresponding events in the second file. (The false positives – A11 and A12 – are readily apparent in file 2.)

If code 03 occurs infrequently, the presence of extra, “interpolated” events can seriously affect the level of reliability reported by the program. Not only have the interpolated events resulted in missing an agreement, they have resulted in two disagreements being credited against code 03. In the current example, this is clearly the wrong thing to do, as the observers have not disagreed about code 03, but about code 20. In such cases as this, it seems reasonable to me to code an agreement for the rare code (03) and to score the two disagreements with code 20 against the residual category (“not codable”) rather than against 03 itself.

Such judgments are beyond the scope of the software. However, it makes these cases easy to spot in the matching event files, as in the example above. If one exports a summary Table of Comparisons (see the sample output presented earlier) to a spreadsheet program, appropriate changes based on an examination of the matching event files are easily entered. The altered spreadsheet can then be imported into Kappa, where it will be analysed. This process is described in the next section.

Exporting and importing a Table of Comparisons.

To export a table of comparisons to a spreadsheet, check *Export Table of Comparisons* in the File menu. A comma-delimited text file with a .CSV extension will be created during the next analysis. If several files are being analysed, export files will be created for each pair and for the summary analysis. CSV files can be imported into any spreadsheet program.

Codes	1	10	15	20	21	25	26	31	45	60	63	72
1	26	0	0	1	0	0	0	0	0	0	0	0
10	0	10	1	0	0	3	0	0	0	0	0	0
15	0	1	5	2	0	0	0	0	1	0	0	0
20	5	1	1	41	0	5	0	0	1	0	0	1
21	0	0	0	0	2	0	0	0	0	0	0	0
25	3	0	0	1	0	65	1	0	2	0	0	0
26	0	0	0	1	0	2	7	0	1	0	0	0
31	0	0	0	0	0	0	0	2	0	0	0	0
45	0	0	1	2	0	5	0	0	13	0	0	0
60	0	0	0	0	0	0	0	0	0	3	0	0
63	0	0	0	1	0	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0	0	0	0	0	2

As the figure illustrates, Kappa creates an export filename based on the names of the files that were compared, and reports these original names in the spreadsheet. Codes are used to label the columns and rows. Agreements and disagreements can then be altered by viewing the corresponding matched event

files, as described earlier.

New values for percent agreement and kappa can be calculated by importing the spreadsheet back into Kappa. Be sure to use the *Save As* function in your spreadsheet to save the file in CSV format. (Kappa will import CSV spreadsheets from Linux or Macintosh applications, as well as Windows programs). Click on *Import CSV file* in the File Menu and select the file using the dialogue box. The analysis will be done and presented automatically.

Remember that one must examine rare codes in their context in order to accurately judge their reliability. In this task, the matched events and agreements identified by KAPPA and reported in the matched event files can be invaluable.

Reliabilities: Percent agreement and kappa for an imported Table of Comparisons
Thursday, November 24, 2011 11:38 a.m.

Data imported from file JANICE.102 AND TALLY.102 COMPARISONS.CSV

Table of Comparisons with marginal totals:

Code	1	10	15	20	21	25	26	31	45	60	63	72	
1	26	0	0	1	0	0	0	0	0	0	0	0	27
10	0	10	1	0	0	3	0	0	0	0	0	0	14
15	0	1	5	2	0	0	0	0	1	0	0	0	9
20	5	1	1	41	0	5	0	0	1	0	0	1	55
21	0	0	0	0	2	0	0	0	0	0	0	0	2
25	3	0	0	1	0	65	1	0	2	0	0	0	72
26	0	0	0	1	0	2	7	0	1	0	0	0	11
31	0	0	0	0	0	0	0	2	0	0	0	0	2
45	0	0	1	2	0	5	0	0	13	0	0	0	21
60	0	0	0	0	0	0	0	0	0	3	0	0	3
63	0	0	0	1	0	0	0	0	0	0	0	0	1
72	0	0	0	0	0	0	0	0	0	0	0	2	2
	34	12	8	49	2	80	8	2	18	3	0	3	219

Comparisons = agreements + average disagreements across coders.

Code	Agreements/Comparisons	Proportion agreement (kappa)	Most frequent disagreements: code (frequency), code (frequency)	
1	26 / 30.5	.852 (.850)	20 (6)	25 (3)
10	10 / 13	.769 (.768)	25 (3)	15 (2)
15	5 / 9.5	.526 (.528)	20 (3)	45 (2)
20	41 / 52	.788 (.776)	25 (6)	1 (6)
21	2 / 2	1.000 (1.000)		
25	65 / 76	.855 (.836)	45 (7)	20 (6)
26	7 / 9.5	.737 (.736)	25 (3)	45 (1)
31	2 / 2	1.000 (1.000)		
45	13 / 19.5	.667 (.664)	25 (7)	20 (3)
60	3 / 3	1.000 (1.000)		
63	0 / .5	.000 (.000)	20 (1)	
72	2 / 2.5	.800 (.800)	20 (1)	

Total agreements / total comparisons = 176 / 219
Proportion agreement = .804
Kappa = .751
Standard Error of Kappa = .032
Approximate 95% confidence interval = .688 to .815

Final thoughts about KAPPA

Although KAPPA uses actors and targets to coordinate comparison of the two files, percent agreement and kappa are based on behaviour codes only. KAPPA does not currently permit 'person' codes to be combined. For purposes of matching sequences, it is usually more useful to distinguish individuals than to pool across them.

Similarly, KAPPA ignores contexts entered in FOCAL, on the assumption that issues of reliability cut across contexts and should be assessed across contexts. Moreover, if the contexts are so different that different coding taxonomies are used (for instance, to assess interactions in families and on playgrounds), then independent observers will be used as well. Each set of observers will generate its own data files, and these separate sets can easily be handled by KAPPA's ability to store and retrieve lists of file names for analysis.

The default value of ten seconds for tolerances has no formal justification. For the time method, 10 seconds seems reasonable to me given my own data and research questions. Because tolerance affects the identification of matched events, it has an important effect on how comparisons are carried out, and consequently on percent agreement and kappa. By setting the range and step values carefully in 'Set Time Tolerances' in the Analysis Menu and reviewing the summary statistics, it is fairly easy to identify the tolerance values producing the greatest overall agreement for a given set of files. Matched event files can then be generated and files with marginal reliability examined to understand why agreement is sometimes poor.

Please let me know if there are any aspects of KAPPA that I haven't made clear, or if there are program features that you would like to see added or changed.

Summary

To run Kappa, you MUST:

- Choose at least two files to compare.
 - Click on *File Menu - Select new files for analysis* and choose one of the three options. All options start with an Open Files dialogue box allowing you to navigate to the drive and directory with your FOCAL data files. Choose the first file to analyse and click 'Open'.
 - If you chose the *Select two files* option, a second Open Files dialogue box will appear and you can choose another file. (Both files must be in the same directory.)
 - If you chose a *Select multiple files* option, a list of files will appear, with the first file already selected. Select one or more additional files to analyse, using the mouse and the Shift or Ctrl keys; click 'OK' when done. The list of file names will be placed in a simple text editor so that you can confirm that they are paired correctly. (The first file will be compared with the second, the third with the fourth, and so on.) If you make any changes to the list, be sure to *Save* it.
 - You may select a previously saved list from the *File* menu - *Select a previous list of files* or from the *File* menu in the window listing the files to be used (*View* menu - *Files to be analysed*).
 - The list of files that you last used is loaded by default when you next start the program (if it was *Saved*).
- Start the analysis (*Analysis* Menu - *Start Analysis*)

You MAY:

- Recombine behaviour codes (*View* menu - *Recoding file*; and *Analysis* menu - *Do combine behaviour codes*).
- Choose either time or event sequences to coordinate comparisons across files (*Analysis* menu).
- Change the minimum number of events needed to define a sequence or a matched event (*Analysis* menu).
- Set a range of time tolerances (*Analysis* menu - *Set time tolerance*).

- Specify whether percent agreement is to be computed using row-totals or the average number of comparisons across both observers (*Analysis* menu).
- Request that matching events be reported (*Options* menu - *Report matching sequences*).
- Request that your new results (and matching sequences) replace or be appended to earlier results. (*Options* menu)
- Request that the entire table of comparisons be shown in the output (*Options* menu - *Show Table of Comparisons*)
- Request that the entire table of comparisons be exported (*File* menu – *Export Table of Comparisons*].
- Import a Table of Comparisons as a .csv file from a spreadsheet (*File* menu – *Import .csv file*). It will be analyzed automatically.

TIMES: Frequency, rate, and proportional times for FOCAL data

Overview

Unlike FOCAL and KAPPA, which are stand-alone programs, TIMES is a module of LAG. To do the analyses described below, start LAG, select the FOCAL data files to be analysed, and then select *Start Analysis* from the *Time Budget Analysis* option in the *Analysis* menu.

For each actor-behaviour-target combination, TIMES reports frequency, rate (as events per minute), mean duration (in seconds), duration as a proportion of total time observed, and frequency as a proportion of all events. If simultaneous events have been coded using user-designated ‘stop’ codes (see below), proportional times will reflect this simultaneity. As with LAG, behaviour codes can be recombined and data pooled across persons, and identical adjacent events may be combined or not.⁸ Results are presented for each case (file) and summarized across cases (for behaviours only, sorted by code or frequency, and optionally for each actor-behaviour-target combination as well). The summary report for behaviours (pooled across actors and targets) is presented when files are analysed one at a time, as well as in groups.

Running the program

TIMES is one of five analysis modules in LAG. Start LAG by clicking on it or its shortcut.

Selecting data files

Click on *Select new files for analysis* in the *Files* menu to choose a new set of files to analyse. Use the Open File dialogue box to find the drive and directory with your FOCAL data files, and choose the one you wish to use. If you are selecting multiple files (to analyse simultaneously), a list of files will appear, with the first file

⁸ Note that frequency, rates, and proportional times may differ from those reported by LAG if the data file analysed contains focal samples for several individuals (as it might, for instance, if all members of a family had served in turn as focal persons) and if the Use All Focal Samples option (page 63) was ‘on’ for one analysis and ‘off’ for the other. Also, of course, if events were coded to indicate simultaneous occurrence, and this option was used in the analysis, proportional times reported by TIMES will differ from those reported by LAG. LAG treats all events as sequentially occurring, mutually exclusive behaviours.

already selected. (Depending on the option you chose, this list will contain only files with the same extension as the first file, or all files in the directory.) Using the mouse with the Shift and Control keys, you may select multiple files as usual in a Windows environment. When you are done, click 'OK'. You will have the option of saving this list. If you do, it will be available for use in later work sessions (and will be loaded by default when you re-start the program).

If you saved a list of files earlier (in a previous work session or perhaps when using Scan), you may select it by clicking on *Select a previous list of files* from the *File* menu.

You may also retrieve an earlier list when viewing *Files to be Analysed* from the *View* menu – just Open the file containing the list that you want. If you make any changes in the list of files, be sure to Save them. This will make the changes active.

Combining behaviour codes.

The procedures for recoding behaviours are the same as those described earlier for KAPPA (page 42).

If you already have a recoding file as a result of reliability analyses with KAPPA, you may load it by selecting *Behaviour Recoding File* from the *View* menu. If you make any changes, be sure to Save them.

Once you have loaded your recode file, you must toggle on *Do Combine Behaviour Codes* in the *Analysis* menu in order for them to take effect (a check mark will appear by the item). The recoding information that you supply will be applied to the current analysis without altering the original data file.

Pooling data across subjects (combining actor-target codes)

Actor-target codes can be combined in the same way as behaviour codes (page 42). Select *Actor-Target Recoding File* from the *View* menu. If a recoding file does not exist, the program will construct one by reading and collating all the default actor-target code files in the current directory. As with behaviour codes, this recoding file must be arranged as a simple two-level outline, with actor-target codes to be combined as the indented sub-topics of the code which forms the super-ordinate category, one code per line. Be sure to Save any changes you make, and to toggle on *Do Combine Actor-Target Codes* in the *Analysis* menu in order for them to take effect.

Note that this option permits you to pool across person codes *within* each file. If you wish to pool across subjects (for example, all members of a treatment or a control group), this can be done by specifying the appropriate set of files (or by copying the relevant data to one file). Normally, data for each subject is kept in its own file, and so the programs treat the file as the unit to be analysed.

In order for TIMES to work properly, the codes you provide must match the codes as they appear in your FOCAL data files. Thus if some of your subject codes are two-digit numbers, all subject codes must contain two digits.

Should identical adjacent codes be combined?

After recoding, it is possible that adjacent actor-behaviour-target events will be identical. Normally, one would want to consider the now-identical adjacent events as a single event, with a duration equal to the combined durations of the original events. This is the default strategy followed by TIMES (and KAPPA, LAG and SEARCH as well).

Under certain circumstances, however, one may want to treat identical events as unique, separate events, whether recoded or not. For example, one may be recording sequences of decisions or choices, or sequences of outcomes from a game like "Prisoners' Dilemma". In this case, toggle off *Do Combine Identical Adjacent Events* in the *Analysis* menu (the check mark will disappear).

Restricting the Analysis to Certain Contexts.

If you wish to analyse only certain contexts, you may specify them by selecting *Contexts* in the *View* menu. If more than one context is named, their relations must be specified by placing "and" or "or" between them. For example, if "classroom", "playground" and "other" are contexts specified in your FOCAL data files, you could analyse only the classroom focal samples by naming only "classroom" in Contexts dialogue box. (Be sure that *Do Use Contexts* in the *Analysis* Menu is toggled on – a check mark will appear next to it.) "Classroom OR playground" would analyse focal samples from either of these contexts, but not "other". "Classroom AND playground" would result in no focal samples being analysed, since these two contexts would not occur together for any given focal sample. Data from all three contexts would be analysed if all three were named and linked by OR, or (more simply) if *Do Use Contexts* in the *Analysis* Menu was toggled off.

Displaying summary output

Results are presented for each case (file) and summarized across cases for behaviours. If you would like to view summary output for each actor-behaviour-target combination, choose this option from the Time-Budget Analysis menu. A check mark will indicate that your choice is active.

‘Stop’ codes: Coding simultaneous behaviours

Stopping specified behaviours.

If one code is used to designate the cessation of on-going activities, it is possible to 'nest' codes, that is, to code simultaneous behaviours. Consider the following example, in which two children (coded 1 and 2) engage in a conversation (coded 10) and then begin to play (coded 20). Talking persists for a while, but then lapses, while playing continues. '9' is used to indicate that a particular behaviour has stopped, but any code may be designated.

1	10	2	10.5	10.5	(1 speaks to 2)
2	10	1	5.0	15.5	(2 speaks to 1)
1	20	2	15.5	31.0	(1 plays with 2)
9	10	9	30.2	61.2	(speaking stops)
9	20	9	1.3	62.5	(play stops)

Note that in the last two lines of our example, the stop code replaces the actor and target codes. When TIMES encounters a 'stop' code, it searches for the first preceding occurrence of the stopped behaviour, and adds any intervening durations to the duration originally noted. Thus in the example above, when 9 10 9 is encountered, TIMES finds the last occurrence of behaviour 10 (namely, 2 10 1) and adds to its original duration (5.0 seconds) the intervening duration (61.2 - 15.5 = 45.7), for a total of 50.7 seconds: the coding indicates that event "1 speaks to 2" began 10.5 seconds into the focal sample, and continued until 61.2 seconds. Note that the durations of the intervening behaviours are unchanged; thus 1 20 2 is still credited with a duration of 15.5 seconds. In the same way, when 9 20 9 is encountered, TIMES finds the last occurrence of behaviour 20, in this case, 1 20 2, and adds 31.5 seconds to

the original duration of 15.5, for a new duration of 47.0 seconds.⁹

Proportional times are still calculated on the basis of total time actually observed (in this example, 62.5 seconds), so that when stop codes are used, proportional times will sum to more than 1.0.

Stopping all interactions for a specified dyad

If desired, a second stop code can be designated to signal the end of all dyadic interactions. Note that this code must be used in conjunction with a stop code for events. For example, if 9 has been designated as the stop code for events and 99 as the stop code for dyads, then 'stop all interactions for the current dyad' MUST be coded as 9 99 9.

Because TIMES does not know what your codes mean, it treats this option much like the 'stop behaviour' code just described. Consider the following modification of our first example, in which 9 and 99 have both been designated as stop codes:

1	05	2	10.5	10.5	(1 attends to 2)
2	10	1	5.0	15.5	(2 speaks to 1)
1	20	2	15.5	31.0	(1 plays with 2)
9	10	9	30.2	61.2	(speaking stops)
9	20	9	2.3	63.5	(playing stops)
9	99	9	1.5	65.0	(dyadic interaction ends)

When TIMES encounters 9 10 9, the duration of 2 10 1 is incremented as described above (to 50.7). Similarly, when it encounters 9 20 9, the duration of 1 20 2 is incremented to 48.0 seconds (1 20 2 began at 15.5 seconds and ended at 63.5 seconds). When 9 99 9 is encountered, TIMES increments the duration of the first preceding code that is not a stop code itself, nor affected by a stop code. In our

⁹ In early versions of TIMES, the duration of the stop code itself was ignored, that is, the on-going behaviour was treated as terminating at the beginning of the stop code rather than the end. If you are using this option and have suggestions about how it should work, please get in touch with me at WLRoberts@tru.ca.

example, this is event 1 05 2, and its duration will be incremented to 65.0 seconds.¹⁰

As this example implies, it is possible to produce "nonsense" results. In particular, the use of stop codes produces distorted output for behaviours that inherently have an element of turn-taking, such as talking, or are time-limited in nature, such as ignoring, in contrast to behaviours such as playing, which are truly simultaneous in nature. Care is needed when coding simultaneous behaviours, and in interpreting the results of TIMES.

Designating, using, and deleting 'stop' codes

If your FOCAL data files contain 'stop' codes to identify simultaneously occurring behaviour, enter one or both by selecting *Stop Codes* from the *View* menu. Note that a stop code for all interactions cannot be entered by itself; it MUST be accompanied by a behaviour stop code.

In order for the program to use the stop codes you have specified, toggle on *Do Use Stop Codes* in the *Time-Budget Analysis* menu in the main *Analysis* menu. A check mark will appear next to the item. Naturally, you can ignore your stop codes by turning off this option, in which case the analysis programs will treat stop codes just like any other event.

You may also chose to *Delete Stop Codes* (an option in the *Sequential Analyses* menu in the main *Analysis* menu). In this case, the analysis programs will treat stop codes as if they were deleted from the data file(s). (Naturally – as with the recoding options – the original data file isn't actually altered.) This reduces total time by the durations of the excluded stop codes, thus affecting the values for proportional time. However, it may give you a clearer view of some of your behaviours.

¹⁰ In early versions of TIMES, 'stop all interaction' codes could increment the durations of several behaviours, not just one. The current method seems less likely to me to produce distorted results; but if you are using this option and would like it changed, please let me know.

Summary

To do a time-budget analysis, you **MUST**:

- Specify one or more data files to analyze.
- Start the analysis (*Analysis Menu - Time-budget Analysis - Start Analysis*)

You **MAY**:

- Combine behaviour or actor-target codes (*View Menu - ... Recoding File and Do Combine...* in the Analysis Menu)
- Specify that identical adjacent events be analyzed as separate, not single, events (*Analysis Menu*)
- Restrict the analysis to one or more specified contexts (*View Menu - Contexts and Analysis Menu - Do Use Contexts*)
- Use all focal samples (*Analysis Menu*), or you may restrict the analysis to only those focal samples for which the criterion actor and target were focal persons.
- Request that your new results replace or be appended to earlier results (*Options Menu*)
- Request that 'stop' codes be used, to yield durations for simultaneously occurring events (*View Menu - Stop Codes and Do Use Stop Codes* in the Time-Budget Analysis Menu in the Analysis Menu.)
- Request that 'stop' codes be treated as if they were deleted from the data. (*Delete Stop Codes* in *Sequential Analyses Menu* in the Analysis Menu)
- When analysing multiple files, restrict the summary report to behaviours only, or include a section reporting actors, behaviours, and targets (*Time-Budget Analysis Menu*).
- When analysing multiple files, sort the summary report of behaviours by frequency or by code (*Time-Budget Analysis Menu*).

LAG: Sequential analyses for FOCAL data

Overview.

The LAG module of LAG performs sequential (or lag) analyses on data collected by FOCAL. The user specifies a criterion event (that is, codes for an initiator, behaviour, and target) and a lag interval (i.e., the number of events following the criterion at which a response will be counted). Behaviour categories can be recombined or data pooled across persons without altering the original data file. (If you like, you can have the recoded data saved in a separate file.) LAG can report on events that precede as well as follow a designated criterion. Long sequences of events can be identified by SEARCH.

LAG reports simple frequencies and rates for all behaviours, as well as conditional probabilities for all responses found at the requested lag intervals. For each of these probabilities, an exact binomial test of significance is reported and three measures of effect size (the odds ratio, Yule's Q , and ϕ) based on the presence or absence of the criterion and response behaviours. Yule's Q and ϕ can be used as scores for subsequent analyses in SPSS or BMDP. LAG also reports an exact binomial test for the omnibus null hypothesis that all lag probabilities are zero. Z scores are also reported. (A positive z score indicates that the response is more likely to follow the criterion than its simple probability would suggest; a negative z score indicates that the response is inhibited, that is, less likely to follow the criterion than its simple probability would suggest.) If more than one file is being analyzed, these statistics are reported for the entire set as well as for the individual files.

Conditional probabilities and their binomial tests of significance are measures of contingency between specific events. If one is interested in general contingency, that is, contingency across the entire taxonomy at a given lag, a log-linear analysis is appropriate (Bakeman & Gottman, 1997). While LAG will not do such an analysis, it will, if requested, generate a log-linear instruction set with data. This file can be run directly by BMDP 4F, or modified for use by SPSS.

Finally, LAG reports cumulative time for the criterion event as a proportion of total time for all events in the focal trials that it has used (those in which the criterion actor or target served as the focal person). This proportion can be used in time-budget

analyses. (See Altmann, 1974).

Running the program

LAG is one of four analysis programs in LAG. Start this program by clicking on it or its shortcut.

Select one or more data files to analyze.

You may select new files or use a list from an earlier analysis (see page 54.)

Entering the criterion behaviour.

From the **Analysis Menu - Sequential Analyses**, choose **Set Criterion Event**, and enter codes for the actor (initiator), behaviour, and target. This event is the starting point (lag 0) for your lag analysis.

The lag interval.

When the lag interval is one, the first event following each occurrence of the criterion event is examined. When the interval is two, the second event following each occurrence of the criterion is examined, and so forth. From the **Analysis Menu - Sequential Analyses**, choose **Set Lag Interval**. You may specify a single lag, or a range of lag values. For example, to do an analysis at lag 1 (the default), set the first and last lag values to 1. "Step" will be ignored. To do analyses at lags 1, 2, and 3, set "first lag" to 1, "last lag" to 3, and "step" to 1. To do analyses at lags 1 and 3 (omitting lag 2), set "first lag" to 1, "last lag" to 3, and "step" to 2. Negative lag values will cause the program to examine events which precede the criterion.

Searching for antecedent events.

Normally, LAG reports on the events which follow a specified criterion. However, it may sometimes be useful in exploratory analyses to ask which events precede a criterion. For example, if we have coded social interactions, including aggressive or agonistic behaviours, we may be interested in knowing which events precede or precipitate an aggressive interchange. LAG will perform such backwards lag analysis if you specify negative values for the lag interval.

Needless to say, the results of such a backwards analysis are subject to the distortions attending any retrospective analysis. Just because code X often precedes an aggressive interchange, we cannot conclude that behaviour X is usually followed by aggression. Nevertheless, backwards lag analyses may be useful for identifying earlier links in chains of contingent behaviours that culminate in events that are

either rare or of central interest. (SEARCH is specifically designed to find such chains, given specified starting behaviours.)

Combining behaviour codes.

Choose **View Menu - Behaviour Recoding File** and **Analysis Menu - Do Combine Behaviour Codes**. See p. 42 for more details.

If you would like to view the recoded data, select **Save recoded data in new file(s)** from the **Options Menu**. LAG will generate the recoded data and save it in a file called "Recoded *original_file_name*". (For example, data from file *Class 39.dat* will be saved in *Recoded Class 39.dat*.)

Pooling data across subjects (Combining actor-target codes)

Choose **View Menu - Actor-Target Recoding File** and **Analysis Menu - Do Combine Actor-Target Codes**. See page 42 for more details. The recoded data can be saved, as described in the preceding paragraph.

Should identical adjacent events be combined?

After recoding, adjacent events may be identical and should usually be combined, but under some circumstances you may want them to be distinct (see page 56). Choose **Do Combine Identical Adjacent Events** from the **Analysis Menu**. Toggle this option on and off by clicking on it (a check mark = yes). This option affects how expected values are calculated for z-scores (see page 67).

Using all focal samples.

Usually lag analyses should only be carried out on focal data trials for which the criterion actor or target were the focal individuals. LAG examines the focal individual specified in the header of each data trial to ensure that this condition is met. However, you can now force LAG to analyse all data trials in a file by clicking on this option in the Analysis Menu. I think this may be useful in two circumstances. One may want to get rates and raw frequencies across all members of a family, each of whom served in turn as the focal individual; or one may wish to do a lag analysis using 'stop' codes in the criterion event. (These substitute for the actor and target codes, as explained on page 57, so that the DOS versions of LAG were unable to analyse them.)

This option affects only whether or not focal individuals determine whether a data trial is analysed. It does not affect the impact of context.

Restricting the Analysis to Certain Contexts.

If you wish to analyse only certain contexts, you may specify them by selecting **Contexts** in the **View Menu** and **Analysis Menu - Do Use Contexts**. See page 56. To use all contexts, turn off **Do Use Contexts** (the check mark will disappear).

Log-linear analyses

Instructions for a log-linear analysis can be generated by clicking on **Analysis Menu - Sequential Analyses - Write Log-Linear Instruction File**. This option is useful for assessing general contingency at a given lag (i.e., contingency across the taxonomy as a whole). When toggled on, LAG will place in file “*subject ID* LogLinear input.txt” the instructions for a log-linear analysis using BMDP 4F, complete with data. (This will be a matrix of frequencies whose row labels are events at lag 0 and whose column labels are events at the lag requested. It does not matter what criterion behaviour is specified. The lag table is determined by the lag requested; the criterion only determines which row of this table is reported in the main output to LAG.)

Because lag tables often contain many cells with frequencies of zero or with small expected values, the probabilities generated by BMDP, SPSS, etc., which are asymptotic estimates, may not be accurate. There are now programs available that calculate exact probabilities for chi-square, and LAG generates an import file for one such program, STATXACT. If you have STATXACT and wish to confirm the probabilities produced by your log-linear analysis, you may directly import this file (“*subject ID* StatXact input.txt”) into STATXACT. This file is produced along with the log-linear file. (Note that the log-linear instruction file must be requested in order to generate the StatXact import file.)

Removing ‘stop’ codes.

If you have identified stop codes (page 57), you may chose **Ignore Stop Codes (Analysis Menu - Sequential Analyses Menu)**. In this case, the analysis programs will treat stop codes as if they were deleted from the data file(s) – see page 59.

Sample output for a lag sequential analysis

Lag sequential analysis
 Thursday, April 25, 2002 at 9:38 a.m.

Data read from file 9127.DAT

Data for this analysis came from focal samples for persons 1 and 5.
 No behaviour codes were recombined.
 Actor-Target recodes read from file Recoded Actors-Targets.txt

Identical adjacent events were combined.
 Contexts were ignored; all contexts were used for this analysis.
 Frequencies and rates for case 91272111

Recoding files and other options are listed here. Frequencies follow.

CODES	N	RATE (N / min)	CODES	N	RATE (N / min)
1 0 1	17	.375	1 72 5	1	.022
1 1 1	1	.022	1 73 5	3	.066
1 1 5	16	.353	1 74 5	2	.044
1 2 1	3	.066	1 94 5	1	.022
1 3 5	9	.199	5 11 1	1	.022
1 4 5	3	.066	5 11 5	1	.022
1 5 1	16	.353	5 12 1	4	.088
1 6 1	1	.022	5 12 5	1	.022
1 6 5	1	.022	5 13 1	37	.816
1 11 5	1	.022	5 13 5	22	.485
1 12 5	6	.132	5 14 1	9	.199
1 13 1	1	.022	5 14 5	9	.199
1 13 5	37	.816	5 24 1	2	.044
1 14 1	2	.044	5 42 1	2	.044
1 14 5	16	.353	5 43 1	4	.088
1 14 8	1	.022	5 52 1	1	.022
1 16 5	1	.022	5 53 1	2	.044
1 24 5	1	.022	5 62 1	1	.022
1 34 5	1	.022	5 63 1	2	.044
1 42 5	2	.044	5 72 1	1	.022
1 43 5	5	.110	5 73 5	1	.022
1 63 5	3	.066	5 74 1	6	.132

Total Events = 257

[example continues on next page]

Lag results for case 91272111, criterion 1 13 5 at lag 1.
 Total pairs of observations = 255

"Lag p" is the lag or conditional probability of the response, given the criterion. The binomial probability indicates the statistical significance of this conditional probability. Negative scores for z, Q, and phi indicate that the response was inhibited (occurred less often than expected by chance). Yule's Q, phi and the odds ratio are measures of effect size for the lag probability. Phi is the correlation between the criterion and response, dichotomized as present or absent; it is equivalent to Cramer's V.

Response	frequency	lag p	Z	exact binomial p	Yule's Q	phi	odds ratio
1 0 1	2	.0556	-.48	.5661	-.111	-.018	.80
1 1 5	2	.0556	-.39	.6833	.007	.001	1.01
1 4 5	1	.0278	.74	.3469	.512	.060	3.10
1 5 1	1	.0278	-1.03	.3309	-.440	-.058	.39
1 12 5	1	.0278	.02	.5756	.100	.011	1.22
1 24 5	1	.0278	2.08	.1319	1.000	.155	*****
1 43 5	2	.0556	1.33	.1568	.618	.105	4.24
5 13 1	17	.4722	4.90	.0000	.798	.377	8.90
5 14 1	2	.0556	.45	.3644	.281	.045	1.78
5 24 1	2	.0556	2.95	.0325	1.000	.219	*****
5 52 1	1	.0278	2.08	.1319	1.000	.155	*****
5 53 1	1	.0278	1.19	.2468	.723	.092	6.23
5 63 1	1	.0278	1.19	.2468	.723	.092	6.23
5 74 1	2	.0556	1.05	.2075	.519	.086	3.16

36

The probability of observing 2 significant lag events in 14 comparisons is .0742 with alpha = .0325

Total time = 45.34 minutes across 2 trials
 Proportion of criterion time / total time = .08705

*** When the response occurs only after the criterion, or only the response follows the criterion, Yule's Q = 1 and the odds ratio = infinity.

Two responses followed the criterion event more often than would be expected by chance (alpha = .05): event 5 13 1 and event 5 24 1. The binomial probability of finding two or more such significant events in 14 comparisons is greater than .07, so we might retain the omnibus null hypothesis that all lag probabilities are zero. (For this omnibus test, the program sets alpha at the largest observed lag p value that is ≤ .05; in this case, .0325.) In this example, however, if event 5 24 1 is excluded and the omnibus test recalculated for event 5 13 1 alone, omnibus p < .001 (14 * .00005). We would conclude that the criterion event is followed reliably by one event, and only one event, at this lag – event 5 13 1.

Some thoughts on z-scores, Yule's Q, and Phi

When adjacent codes must differ (**Analysis Menu - Do combine identical adjacent events** is toggled on), expected values and z are calculated as recommended by Sackett (1979). This is the familiar z -test for proportions, found in most introductory statistics textbooks and functions in this context as the normal approximation to the binomial test. When adjacent codes can be the same, expected values and z are calculated according to Bakeman & Quera (1995), formula 6. This is the Allison-Liker z , which is similar to the adjusted standardized residual (Bakeman & Quera (1995), formula 5; BMDP Manual, vol 1, p. 275). In either case, there are several noteworthy problems.

(1) As the expected (chance) probability departs from .5 and becomes more extreme (either large or small) Sackett's z score is not normally distributed (e.g., DeVore, 1982; Hayes, 1988), and therefore *cannot be used to assess the null hypothesis* of no contingency between criterion and response. The same is true for the Allison-Liker z at lags greater than 1 (Bakeman & Quera, 1995, p. 279). Even at lag 1, the number of observations required becomes very large as the expected value of p becomes small.

It is for these reasons that LAG and SEARCH report the exact binomial probability as a test of the null hypothesis between criterion and response events. When z is positive, the binomial probability is for the upper cumulative binomial distribution, that is, it is the probability of observing *n or more* responses at the lag specified. When z is negative, the test is for the lower cumulative distribution, that is, the probability of observing *n or fewer* responses. In both cases, the expected probability is an unadjusted simple proportion in the lag table: the total frequency of the response event divided by the total number of entries.

(2) Bakeman and Gottman (1986) suggested using z scores as substitute (transformed) scores, replacing conditional probabilities in supplementary analyses. However, because they are derived from a z -test of proportions, these z scores are affected by the number of observations. This is simply a consequence of the increase in power that accompanies an increase in sample size. Because the z -value for a given conditional probability becomes larger as the number of events observed

becomes larger, *it cannot function as a standard score*: it does not in fact adjust for variation across subjects in number of events or total time observed. Bakeman & Gottman (1997) noted this problem in the second edition of *Observing Interaction*.

Bakeman & Gottman (1997) now recommend that measures of effect size, such as Yule's Q or ϕ , be substituted for conditional probabilities in supplementary analyses when the independent variables are continuous in nature (e.g., for correlation, regression, or path analyses). LAG reports both Q and ϕ . If the independent variables are categorical, the statistical strategy of choice is a log-linear analysis using the lag frequencies (Bakeman & Gottman, 1997; Bakeman & Quera, 1995) – if the table is not too sparse.

Although Q and ϕ are reported to be functionally equivalent (Bakeman & Gottman, 1997, p. 130), there are several conditions under which the use of Q as a score in additional analyses appears to be problematic. First, Yule's Q equals 1 when *only* the lag event ("A") follows the criterion event *or* when lag event "A" *never* follows a *non-criterion* event. These conditions may not be uncommon when frequencies are low. ϕ gives a less exaggerated measure of effect size in these cases. A second problem arises more often. If frequencies for the criterion event and lag event "A" are small relative to the *combined frequencies* for the remaining codes, Q will always be large (for example, $> .90$). ϕ , in contrast, does not display this behaviour. Consequently, scatter plots suggest that the two indices are not functionally equivalent under these conditions. For these reasons, it is probably preferable to use ϕ in additional analyses.

(3) There is disagreement about the proper way to calculate the expected probabilities used in computing z-scores and exact binomial probabilities. Traditionally, the expected probability of a given code is the frequency of that code in the entire data set divided by the total number of events in the data set. (This value is reported by TIMES.) Sackett (1979), however, points out that if adjacent events cannot be identical, then at lag n we *cannot* find the events that just occurred at lag $n - 1$. It follows that the denominator for expected probabilities is not the total number of events, but rather the total number of events minus the frequency of the criterion event in the lag analysis. Thus at lag 1, we know for certain that we cannot find any of the criterion codes: these are all occurring at the previous lag, lag 0. Therefore the

criterion codes should be excluded from the grand total of events which is used to calculate expected probabilities at lag 1.

In contrast to Bakeman & Gottman (1986), Sackett argues that the same considerations apply at any lag. At lag 2, for example, there will still be c events (where c = the frequency of the criterion code) that occurred at the preceding lag (lag 1) that cannot occur at the given lag (lag 2). Therefore c still needs to be deducted from the grand total of all events, the denominator used to calculate expected frequencies at the given lag. Moreover, *the numerator for the expected frequency also needs to be adjusted*. If code A occurs t times across the total data set and p times at the prior lag, it follows that it can only occur $t - p$ times at the current lag.¹¹

While Bakeman & Gottman (1986) accept Sackett's argument for calculating expected probabilities at lag 1, they make no adjustments at higher lags.

Bakeman & Gottman's recommendations differ from Sackett's in another way as well. While Sackett takes the grand total of all events in the data set, Bakeman & Gottman take the grand total of events in the lag table (that is, the table of all possible event pairs at the lag specified). However, the total events in the lag table decreases as the lag interval increases, because at lag n , the last $n - 1$ events in each trial are not included as criterion events, while the first n events are not included as responses. Thus for a seven-event trial, A B A C D A B, there are five pairs of lag 2 events (AA, BC, AD, CA, and DB) and consequently five entries in the lag table. If the criterion event was "C", then by Sackett's method, the grand total would be $7 - 1 = 6$; the frequency of A = 3, and the expected probability of A would be $3/6 = .5$. In contrast, by the strategy of Bakeman & Gottman, the grand total is 5 and the frequency of A = 2. (A has occurred twice as the first member of a pair of events and twice as the second.) The expected probability is $2/5 = .4$.

Naturally, as the number of events increases, the results of the two methods rapidly converge. Nevertheless, it appears that in certain cases, as in the example

¹¹ Naturally, if you have specified that adjacent events can be identical, no adjustments are made. At lag 1 and at every lag, the expected probability of a code is simply its frequency divided by the total number of events. For programming convenience, the same holds when a "backwards lag analysis" is done (see page 62).

above, the methods of Bakeman & Gottman (1986) do not precisely give the raw frequency for some codes, and do not yield a reasonable expected probability.

Moreover, because the size of the lag table decreases as lag increases, the z score for a given conditional probability for a given pair of codes will differ depending on the lag.

(This follows from the fact that the expected frequency will change as the lag changes.) In contrast, Sackett's methods yield a constant expected probability, so that z is independent of the lag examined. It is for this reason that I have used it when adjacent events must differ.

Summary

To do a lag analysis, you **MUST**:

- Specify a criterion actor-behaviour-target event (**Analysis Menu - Sequential Analyses - Set Criterion Event**)
- Specify one or more data files to analyse (page 54).
- Start the analysis (**Analysis Menu - Sequential Analyses - Start Lag Analysis**)

You **MAY**:

- Specify a new lag value or range of lag values (page 62)
- Combine behaviour or actor-target codes (page 42)
- Save recoded data to a separate file. (**Options menu**)
- Specify that adjacent events that are identical be analysed as separate events, rather than as a single combined event (page 56)
- Restrict the analysis to one or more specified contexts (page 56)
- Restrict the analysis to only those focal samples for which the criterion actor and target were focal targets, or you may **Use All Focal Samples (Analysis Menu)**.
- Request that 'stop' codes be ignored (treated as if they were deleted from the data; page 64)
- Request that a BMDP log-linear instruction file (and optionally, a StatXact import file) be written (page 64).
- Request that your new results replace or be appended to earlier results (**Options menu**)
- Request that, for each analysis, results for all cases be placed in the same file (identified by the criterion event) or that results be placed in separate files, one for each case. (**Options menu**)

SEARCH: Patterns of contingent behaviour in FOCAL data

Overview

The SEARCH module of LAG finds sequences of contingent behaviour beginning at any specified actor-behaviour-target criterion. It will find any number of simple or branching patterns, each with a maximum length set by the user (the default is 10 events). Users can also specify the alpha level for the significance of the individual lag analyses that SEARCH examines (the default value is .01). A test of significance for each full sequence is also reported, along with the total number of actual occurrences in the data file. SEARCH will examine existing LAG output, or it will generate needed LAG output if none exists.

SEARCH uses a strategy first devised by Sackett and described by Bakeman and Gottman (1986). In order for a chain of events to be considered statistically significant, all its lag permutations must be significant. For example, for the four-event chain A-B-C-D to be significant, all its lag 1 conditional probabilities (A-B, B-C, C-D) must be significant, as must all its lag 2 probabilities (A-C, B-D), and all its lag 3 probabilities (A-D).

Following Sackett (1979), a binomial test assesses the significance of the sequence as a whole. This is the probability of observing N or more sequences in t trials (= the number of criterion events), under the null hypothesis that the sequence occurs by chance. The null or expected probability is simply the product of the respective lag 1 conditional probabilities.

If one wishes to assess general orderliness across the entire taxonomy, a log-linear analysis would be appropriate – see page 64.

Running the program

SEARCH is one of four analysis modules in LAG (from the *Analysis* menu select *Sequential Analyses*, then *Begin search for sequences*). Start LAG by clicking on it or its shortcut. Your first step is to select a data file to analyse.

Select one or more data files to analyse.

You may select new files or use a list from an earlier analysis. See page 54.

Entering the criterion behaviour.

From the Analysis Menu - Sequential Analyses, choose Set Criterion Event, and enter codes for the actor (initiator), behaviour, and target. This event is the starting point (lag 0) for the contingent sequences reported by Search.

Combining behaviour codes.

Choose View Menu - Behaviour Recoding File *and* Analysis Menu - Do Combine Behaviour Codes. See p 42 for more details.

Pooling data across subjects (Combining actor-target codes)

Choose View Menu - Actor-Target Recoding File *and* Analysis Menu - Do Combine Actor-Target Codes. See page 42 for more details.

Should identical adjacent codes be combined? (Analysis Menu)

After recoding, adjacent events may be identical and should usually be combined, but under some circumstances you may want them to be distinct (see page 56). Choose Do Combine Identical Adjacent Events from the Analysis Menu. Toggle this option on and off by clicking on it; a check mark = yes. This option affects how expected values are calculated for z-scores (see page 67).

Restricting the Analysis to Certain Contexts.

If you wish to analyse only certain contexts, you may specify them by selecting Contexts in the View Menu *and* Analysis Menu - Do Use Contexts. See page 56.

Restricting sequence length.

You may set the maximum sequence length to any value you like (Analysis Menu - Sequential Analyses - Set Maximum Sequence Length). It is worth noting that long sequences are not necessarily optimal solutions. In general, the longer a sequence, the fewer times it occurs in a data set. If it is possible to truncate a sequence at a theoretically appropriate point, this shorter sequence may occur more frequently than its longer form.

Setting the level of significance for individual lag analyses.

As discussed on page 72, SEARCH examines all possible lag links in a given sequence. You can set the level of significance required for these links. (Analysis Menu - Sequential Analyses - Set Alpha Level for Search). The default value is .01.

Removing 'stop' codes.

If you have identified stop codes (page 57), you may chose Delete Stop Codes

(Analysis Menu - Sequential Analyses Menu). In this case, SEARCH treats stop codes as if they were deleted from the data file(s). This reduces total time (by the durations of the excluded stop codes), thus affecting the values for proportional time, but it may give you a clearer view of the sequelae of your criterion event.

Sample output for Search

Data read from file home.18
 Sunday, July 22, 2001 at 12:46 a.m.
 Required alpha = .01
 Maximum sequence length = 5 lags.

Recoding files and other options would be listed if they had been used.

Sequence 1 of 12
 This sequence followed the criterion 19 of 470 times (.040)
 expected probability = .048, z = -.752, binomial p = .2662

Although the individual lags are highly significant (see below), the sequence as a whole occurs less often than expected by chance.

Lag	Response	frequency	lag p	Z	exact binomial p
0	2 20 3				
1	3 20 2	263	.5187	28.97	.0000
2	2 20 3	281	.5854	31.66	.0000
3	3 20 2	263	.5187	28.97	.0000
4	2 20 3	281	.5854	31.66	.0000
5	3 20 2	263	.5187	28.97	.0000

Lag probabilities and frequencies for each event in the sequence are taken from the Lag 1 analysis for the prior event. The expected probability of the sequence is the product of these lag 1 probabilities.

-----.... [output omitted here]

Sequence 7 of 12
 This sequence followed the criterion 0 of 508 times (.000)
 expected probability = .018, z = -3.076, binomial p = .0001

Lag	Response	frequency	lag p	Z	exact binomial p
0	2 20 3				
1	3 22 2	83	.1637	13.67	.0000
2	3 1 3	20	.1117	7.34	.0000

It is possible for a sequence to have a complete set of significant lag probabilities and yet never occur in the data set, as Sackett (1979) points out. (This follows from the fact that the lag probabilities are less than 1.) It is for this reason that SEARCH verifies sequences by counting their actual frequency and performing a binomial test of significance. In this case, the test is significant, indicating that the sequence is inhibited.

Sequence 8 of 12

This sequence followed the criterion 2 of 506 times (.004)
 expected probability = .000, z = 4.563, binomial p = .0118

Lag	Response	frequency	lag p	Z	exact binomial p
0	2 20 3				
1	3 78 2	17	.0335	5.43	.0000
2	2 20 3	12	.2857	3.35	.0027
3	3 78 2	17	.0335	5.43	.0000

[further output omitted]

Summary

To do a SEARCH, you MUST:

- Specify a criterion actor-behaviour-target event (Analysis Menu - Sequential Analyses - Set Criterion Event)
- Specify one or more data files to analyse (page 54).
- Start the analysis (Analysis Menu - Sequential Analyses - Begin Search for Sequences)

You MAY:

- Set the level of significance required for individual lag analyses (page 73)).
- Set maximum sequence length, in lags (page 73)
- Combine behaviour or actor-target codes (page 42)
- Specify that identical, adjacent events be analysed as separate events (page 56)
- Restrict the analysis to one or more specified contexts (page 56)
- Request that 'stop' codes be ignored (treated as if they were deleted from the data; page 64)
- Request that your new results replace or be appended to earlier results (Options Menu).

COUNT: Patterns of behaviour in FOCAL data

Overview

When comparing sequences of contingent behaviour across groups, it is necessary to know how often a sequence that was statistically significant in one group occurs in the other. For example, we might have two groups of children, one designated as aggressive and the other as non-aggressive (see, e.g., Pepler, Craig, & Roberts, 1994). We would like to examine the question of whether children in these two groups respond in statistically consistent ways following a given criterion behaviour, and whether these sequential patterns of behaviour differ across groups. SEARCH will answer the first question; COUNT is necessary to answer the second.

The COUNT module of LAG can read sequences directly from SEARCH output for one group and count these sequences in the FOCAL data files of a second group. It compares frequency of occurrence in the two groups using χ^2 . COUNT can also look for user-specified sequences. It will use the recoding files used by SEARCH if directed to do so. Single or multiple files can be examined.

Running the program

COUNT is one of four modules in LAG (click on *Start Analysis* in the *Count Sequences* option of the *Analysis* menu). Start LAG by clicking on it or its shortcut, and begin by selecting a data file.

Select one or more data files to Count.

You may select new files or use a list from an earlier analysis. See page 54.

Specify the sequences to be Counted.

You may identify a file containing SEARCH output for another individual or group (or a different context) by choosing Analysis Menu - Count Sequences - Select Search Result File with Sequences. If this SEARCH output file contains sequences for more than one individual, you may use Select Original Search Data File in the Count Sequences Menu to identify the data file whose sequences you want to Count in your new data set. If no original data file is designated, COUNT will use the first set of sequences that it encounters.

Alternatively, you may define sequences of interest to you. Choose Analysis Menu - Count Sequences - Define Sequences to Count. You will be placed in a simple text editor with the usual cut, copy, and paste tools. Any number of sequences, each separated by a single blank line, can be specified. In the example in the box at the right, portions of a SEARCH output file have simply been copied and modified. COUNT ignores any data after the first three elements in a line (actor - behaviour - target); you may use this space for comments, if you wish.

20 98 20				
20 00 20	45	1.0000	7.317	.0000
20 10 20	37	.3190	2.616	.0079
20 00 20	25	.6250	2.253	.0183
20 10 20	37	.3190	2.616	.0079
20 98 20				
20 00 20				
20 05 20				
20 00 20				
20 05 20				

Combining behaviour codes.

Usually you will want to use the same recoding files for both groups – the one that has been Searched and the one you want to Count. Choose View Menu - Behaviour Recoding File *and* Analysis Menu - Do Combine Behaviour Codes. See page 42 for more details.

Pooling data across subjects (Combining actor-target codes)

Again, you will usually want to duplicate the conditions of the original Search. Choose View Menu - Actor-Target Recoding File *and* Analysis Menu - Do Combine Actor-Target Codes. See page 42 for more details.

Should identical adjacent codes be combined? (Analysis Menu)

After recoding, adjacent events may be identical and should usually be combined (see page 56). Again, you will usually want to replicate the conditions of the original Search.

Restricting the Analysis to Certain Contexts.

If the original Search was restricted to certain contexts, you may want to apply the same conditions to your Count. Alternatively, you could Count sequences found in one context in another context. Select Contexts in the View Menu *and* Analysis Menu - Do Use Contexts. See page 56.

Removing 'stop' codes.

If you have identified stop codes (page 57), you should chose the same option for Count that was used with Search – either to Delete Stop Codes (Analysis Menu - Sequential Analyses Menu), or not. If stop codes are ignored, they are treated as if they were deleted from the data file(s).

Sample output for Count

File 1102.dat was Searched and the results reported in "SEQUENCES REPORT FOR EVENT 10 13 40.TXT". We now count the occurrences of this sequence in two other data files, 1101.dat and 9127.dat.

Data read from file 1101.dat
 Original sequences read from file SEQUENCES REPORT FOR EVENT 10 13 40.TXT for data file 1102.dat
 Tuesday, July 24, 2001 at 9:23 a.m.

Test sequence 1 of 1:
 Lag Event
 --- -
 0 10 13 40
 1 40 13 10

The first line reports the data file being Counted. The next line reports the file from which sequences were read, and the data file in which these sequences were found.

The sequence being "tested" is the sequence that Search found in "SEQUENCES REPORT FOR EVENT 10 13 40.TXT".

This sequence occurred 5 of 10 times (.500) in file 1101.dat
 In the original analysis, this sequence occurred 9 of 25 times (.360)
 Chi-square(1) = .583, p = .445, V = .129
 With Yate's correction for continuity, p = .703

Yate's Correction yields a *p*-value much closer to the exact probability than the usual asymptotic estimate, which is also reported.

 Data read from file 9127.dat

The second data file requested.

Original sequences read from file SEQUENCES REPORT FOR EVENT 10 13 40.TXT for data file 1102.dat
 Tuesday, July 24, 2001 at 9:23 a.m.

Test sequence 1 of 1:
 Lag Event
 --- -
 0 10 13 40
 1 40 13 10

No occurrences of the criterion event were found in file 9127.dat

Summary

To do a COUNT, you MUST:

- specify the SEARCH output to read OR define the sequences to be counted (page 78)
- Specify one or more data files to analyse (page 54).
- Start the analysis (Analysis Menu - Count Sequences - Start Analysis)

You MAY:

- Combine behaviour or actor-target codes (page 42)
- Specify that identical, adjacent events be analysed as separate events (page 56)
- Restrict the analysis to one or more specified contexts (page 56)
- Request that 'stop' codes be ignored (treated as if they were deleted from the data; page 64)
- Request that your new results replace or be appended to earlier results (Options Menu).

INTERVAL ANALYSIS: Identifying episodes of behaviour

Overview

We are sometimes interested in examining episodes of behaviour – episodes of conflict, episodes of talking (conversations), or, as in the example below, episodes of emotional distress during family interactions . How often do they occur? How long do they last? What conditions initiate them, and how are they terminated? Before we can answer such basic questions, we need to specify the criteria for identifying them.

Episode onset is well defined – it is marked by the first occurrence of the criterion behaviour of interest (conflict, talking, emotional distress). Often, the end of an episode is also easy to define – there will be a change in activity and often a change in dyadic partner. However, there are times when the criterion behaviour ceases without a change in context or background activity and without a change in dyadic partner. The question then is how long an interval must elapse before we should consider a new occurrence of the criterion behaviour as the beginning of a new episode, rather than a continuation of the previous episode. There are two issues: the length of the interval that marks or separates two episodes, and the reasons for choosing it, given that there are arbitrary or intuitive elements in the choice. This module provides several tools that can provide information relevant to these issues.

Identifying intervals between criterion behaviours

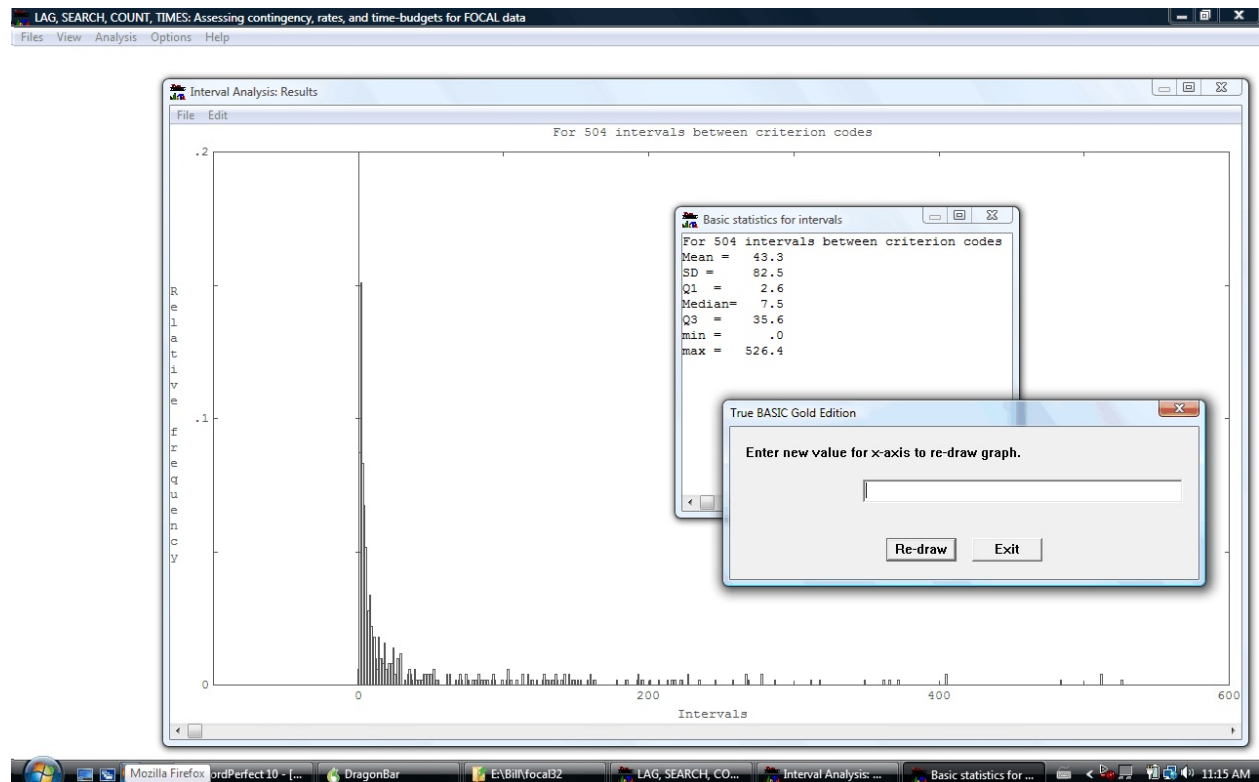
In trying to identify a critical interval length, it is obviously helpful to know what the actual interval lengths are between all the criterion behaviours in the data set, and how they are distributed.

- Start LAG.
- *Select a file or set of files* that you would like to analyze, if these were not loaded when the program started.
- If your behaviour taxonomy contains several relevant criterion codes (for example, codes for different types of conflict, or codes for different types of emotional distress), confirm that these will be combined properly (*View Menu – Behaviour Recoding File*) and that the recoding option is active (there will be a check mark beside the *Do Combine Behaviour Codes* in the *Analysis Menu*).

- *Specify the criterion behaviour.* From the *Analysis Menu*, choose *Interval Analysis* and *Set Criterion Behaviour for Interval Analysis*. In the dialog box that appears, you *must* specify a numeric behaviour code, and you *may* specify codes for either an actor or a target. Thus one could specify only a code for conflict, for example, or codes for conflicts initiated by a particular person, or for conflicts between two particular people.
- Choose *Start Analysis* from the *Interval Analysis* menu.

When the analysis is done, a graph of the distribution of the intervals will be presented.

As shown below, the graph initially displays the entire range of values along the horizontal axis, from zero to the maximum interval found; but the axis scale can be reset and the graph re-drawn. (As expected, most of the intervals are very short: codes are closely spaced within episodes). The critical section is where the curve inflects, flattening out to form the tail. In the example, this occurs between zero and 100 seconds, and we can look at this section more closely by specifying 95 in the dialogue box and clicking on *Re-draw*. A new graph and dialog box will appear, with



values from 100 sec out to the maximum represented by a bar marked “hi”. When you are satisfied with the x-axis value range, exit the dialog box.

The descriptive statistics in the small window indicate that half the criterion codes occur within 7 seconds of each other, after which the intervals quickly lengthen. Clearly the short intervals in the peak represent codes within episodes, while the long intervals in the tail indicate gaps between episodes. Thus the proper cut point is somewhere in the tail – but where? Visually, there is a break at 30 sec, close to the 3rd Quartile, and another, perhaps clearer break, at 60 secs. With these two potential criteria in mind, we can examine the results file for additional information.

Detailed information on intervals.

The results file can be viewed either from the graph window (*File – View Interval Data*) or from the main window (*View – Results*). Following the usual summary information (data files analyzed, recoding files, if used) and descriptive statistics from the graph window, every interval is presented in ascending order, along with its case number, its percentile, and its location in the data files.

The portion of the results file that is of interest to us at the moment – the intervals between 30 and 60 seconds – is presented in the box on the next page. We can see that a cut point at 30 seconds is at the 74th percentile, whereas a cut point at 60 seconds will include 80% of all intervals; and that the choice of cut point will affect 32 cases (= 418 - 386). These 32 are the critical cases that we need to examine in detail. In this process, we can use models based on random distributions to help us decide which cut point is more appropriate.

Modeling the distribution of episodes

It is reasonable to assume (or to require) that episodes are independent of each other, as the presence of some underlying connection (in this case, a persisting mood during those periods when emotional distress was not actively displayed) would be reason to consider that the events in question constituted a single episode. In addition, because these focal samples were collected to be independent of behaviour (Altmann, 1974), it is reasonable to assume that episodes will be randomly distributed across them.

case	percent	interval	ending at	location	
----	-----	-----	-----	-----	
...	[data omitted]				
386	74	29.5	205.2	Family 35, Home Session 1.	24 JAN 1990 at 20:53 Trial 12, Focal person = 1
387	74	31.7	331.3	Family 18, Home Session 4.	04 JUL 1989 at 19:28 Trial 6, Focal person = 2
388	74	33.7	483.2	Family 18, Home Session 3.	20 JUN 1989 at 19:21 Trial 5, Focal person = 3
389	75	34.3	212.1	Family 18, Home Session 2.	12 JUN 1989 at 19:57 Trial 9, Focal person = 3
390	75	34.6	451.4	Family 18, Home Session 4.	04 JUL 1989 at 19:11 Trial 5, Focal person = 3
391	75	35.0	589.3	Family 32, Home Session 3.	09 JAN 1990 at 18:52 Trial 1, Focal person = 3
392	75	35.5	577.4	Family 18, Home Session 1.	31 MAY 1989 at 20:14 Trial 11, Focal person = 3
393	75	35.5	157.9	Family 32, Home Session 3.	09 JAN 1990 at 20:21 Trial 7, Focal person = 3
394	76	35.9	582.4	Family 30, Home Session 2.	04 DEC 1989 at 19:39 Trial 8, Focal person = 1
395	76	36.9	535.9	Family 24, Home Session 4.	03 APR 1990 at 18:34 Trial 2, Focal person = 2
396	76	38.8	403.3	Family 6, Home Session 4.	26 JAN 1989 at 20:10 Trial 9, Focal person = 3
...	[data omitted]				
413	79	51.6	253.6	Family 1, Home Session 3.	15 FEB 1989 at 19:46 Trial 10, Focal person = 2
414	79	51.7	606.2	Family 19, Home Session 1.	08 JUN 1989 at 18:02 Trial 3, Focal person = 3
415	80	51.9	349.0	Family 6, Home Session 4.	26 JAN 1989 at 19:37 Trial 7, Focal person = 3
416	80	53.0	186.4	Family 35, Home Session 4.	21 FEB 1990 at 19:53 Trial 4, Focal person = 1
417	80	53.7	344.8	Family 3, Home Session 2.	09 FEB 1989 at 18:57 Trial 5, Focal person = 3
418	80	55.2	313.9	Family 19, Home Session 2.	06 JUL 1989 at 20:28 Trial 9, Focal person = 3
419	80	60.7	459.5	Family 34, Home Session 1.	27 NOV 1989 at 20:13 Trial 9, Focal person = 2
...	[data omitted]				

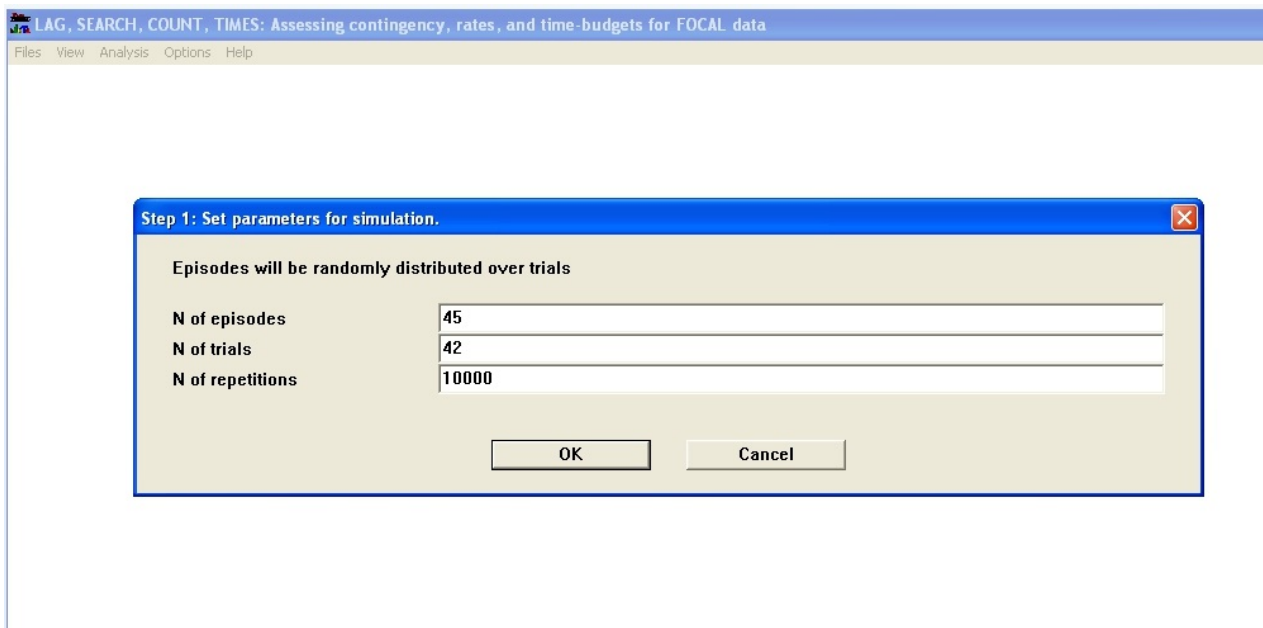
Partial results of interval analysis

For reasons of power, we should use the family with the most cases. Among the 32 critical cases in our example, seven come from Family 18 and seven from Family 19. Output from Scan (see the second chapter of this manual

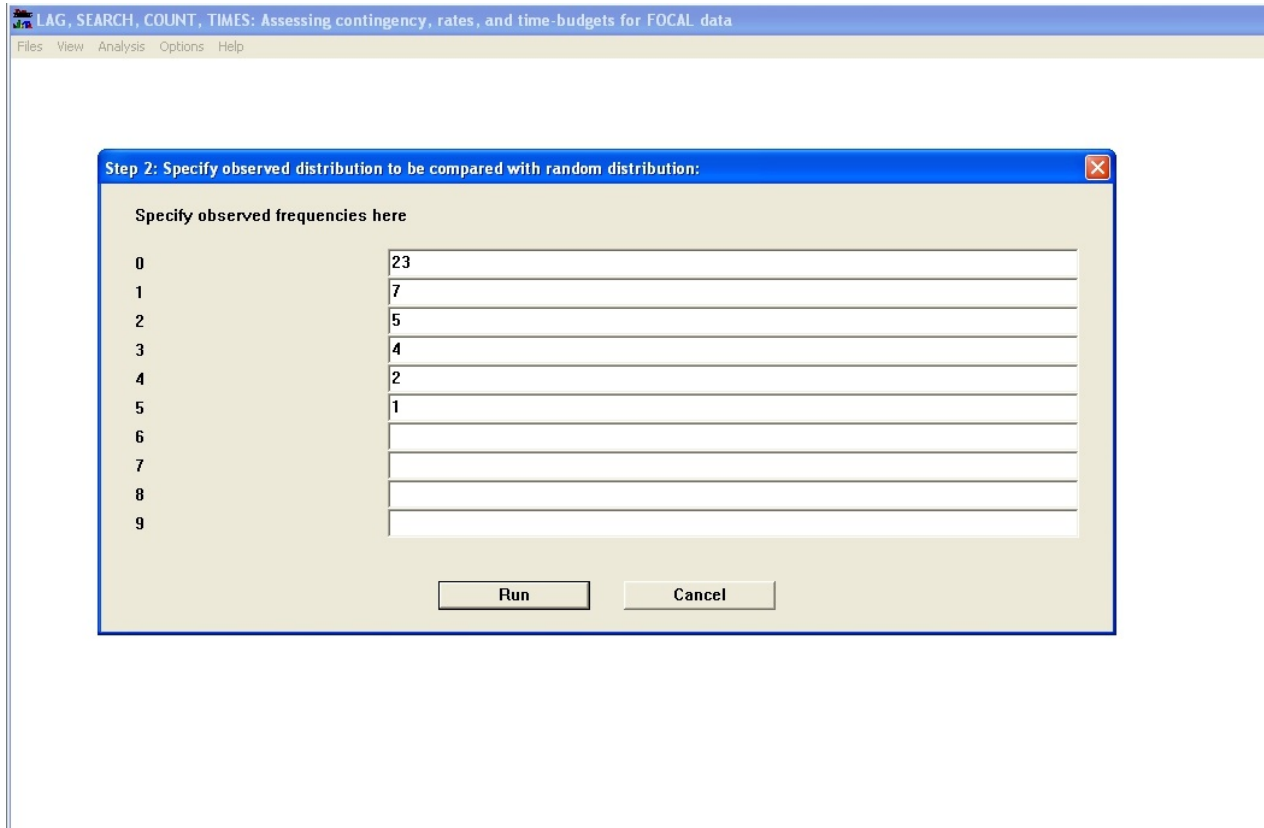
indicates that Family 18 has 42 10-minute focal samples and Family 19, 41. However, output from *Locate All Instances of Criterion Behaviour* (see below) reports 135 criterion codes for Family 18, compared to 106 for Family 19.

Thus we begin by coding all 42 focal samples for Family 18 using a 30-second cut point (plus any other relevant criteria, such as change of dyadic partner). Output from *Locate All Instances of Criterion Behaviour* will ensure that we do not overlook any codes. We then re-code the trials listed among the 32 critical cases, using a 60-second cut point. (The non-critical trials will not need to be re-coded, as their shorter intervals will produce the same results under either decision rule.) We then tally the resulting distributions (that is, the number of focal samples with no episodes of distress, one episode, two episodes, and so forth).

Across the 42 focal samples for Family 18, the 30-second rule generated 45 episodes of emotional distress. Seven focal samples had one episode each, five focal samples had two episodes each, four focal samples had three episodes each, two focal samples had four episodes each, and one focal sample had five episodes. As expected, the 60-second rule generated fewer episodes (34), and fewer focal samples with multiple episodes (values were 10, 4, 4, 1, and 0, respectively, for focal samples with one, two, three, four or five episodes). For both rules, of course, there are the same number of focal samples with no episodes (= 23).



How well do either of these distributions meet the requirement that they are randomly distributed? We begin with the distribution generated by the 30-sec cut point. From the *Interval Analysis – Model Random Distributions* menu, select *Use Simulations to Assess an Observed Distribution*. Parameters for the random model are set in the first dialogue box – see below. The 45 episodes will be randomly distributed



across 42 trials (= focal samples) and the distribution tabulated. This process will be repeated 10,000 times, and this empirical sampling distribution will form the basis for comparison with the observed distribution generated by the 30-sec rule. The number of repetitions is arbitrary, but should be large. 10,000 has been specified here because the observed distribution has extreme values (focal samples with more than three episodes).

The observed distribution is entered into the second dialogue box, above. Click *run*.

Results are shown on the next page. Both the χ^2 goodness-of-fit test and the maximum likelihood test (G) are highly significant, indicating that the observed

LAG, SEARCH, COUNT, TIMES: Assessing contingency, rates, and time-budgets for FOCAL data
 Files View Analysis Options Help

Simulation analysis
 Monday, February 1, 2010 at 6:37 p.m.

Frequency distribution of 45 episodes placed randomly into 42 focal samples, across 10,000 simulations.

N	p	cum p	p(one or more)	observed freq	expected	comp of chi	stand. deviate	F-T deviate	comp. of G	p(obs freq)	
0	.3386	.3386	1.0000	23	14.2	5.42	2.33	2.09	22.11	.0043	upper
1	.3708	.7094	1.0000	7	15.6	4.72	-2.17	-2.48	-11.19	.0034	lower
2	.1983	.9077	.9999	5	8.3	1.33	-1.15	-1.17	-5.10	.1342	lower
3	.0700	.9777	.9526	4	2.9	.38	.62	.66	2.46	.3388	upper
4	.0181	.9959	.5362	2	.8	2.02	1.42	1.13	3.86	.1766	upper
5	.0035	.9994	.1365	1	.1	4.97	2.23	1.15	3.84	.1365	upper
6	.0006	.9999	.0231								
7	.0001	1.0000	.0031								
8	.0000	1.0000	.0006								

Chi square goodness of fit = 18.84, df = 5, p = .0021
 Likelihood ratio test for goodness of fit (G) = 15.98, df = 5, p = .0069

p(one or more) = the probability of observing 1 or more such trials in a set of 42.
 comp. chi = component of chi-square.
 stand. deviate = the standardized deviate from the chi-square analysis.
 F-T deviate = Freeman-Tukey deviates. These are similar to z-scores when the data are from a Poisson distribution.
 comp. of G. = component of G
 p(obs freq) = the binomial probability of finding the observed frequency or more (upper binomial) or the observed frequency or less (lower binomial), in a set of 42 trials.

distribution is very unlikely to have come from a random distribution. The 30-sec cut point is clearly inappropriate and should not be used.

The next step is to assess the distribution generated by the 60-sec cut point. Repeating the steps above for these values yields *p*-values of .06 for the χ^2 goodness-of-fit test, and .10 for G. The fit is poor. These results suggest that a 60-sec cut point is probably not generating a random distribution and that a longer interval (perhaps 90 or 120 seconds) should be considered. Thus the detailed quantitative analyses have served as a useful correction to impressions based on a visual inspection of the data.

Measures of deviation. In addition to information about the simulated distribution (the proportion of focal samples with no episodes, one episode, two episodes, etc.) and the observed distribution, the output reports expected values (= simple probability from the second column x the number of focal samples or trials), the components of χ^2 , the standardized deviates for the χ^2 analysis, Freeman-Tukey deviates (equivalent to z-scores when the data come from a Poisson distribution), and components of χ^2 and G (Dixon, 1992, p. 275).

The final column reports the binomial probabilities for the values in the observed distribution, given the simple probabilities generated by the simulation. In the example above, 23 of 42 trials were found to have no episodes of distress, more than we would expect, given the probability from the random model. The binomial probability indicates that we are highly unlikely to see 23 or more such trials in a set of 42, *p* = .0043. Similarly, the number of trials with one episode, seven, was less than

we would expect by chance, and the binomial probability indicates that we are highly unlikely to see seven or fewer such trials in a set of 42, $p = .0034$.

Given that there are too few trials with one episode, and perhaps too many with four or five episodes, the obvious strategy would be to search and test reasonable criteria that would produce fewer (but longer) episodes – see below.

A note on zero values in the simulated and observed distributions. In the output above, results for the simulation are only reported when frequencies are greater than zero. Thus no simulated trials were found to have nine or more episodes in them, while the proportion of trials with eight episodes is non-zero, but so small that the value is zero to four decimal places. No observed trials had six, seven, or eight episodes in them, but this is consistent with the simulated distribution – not only are the simple probabilities very small, but the binomial probabilities given in the fourth column indicate that we are very unlikely to see such trials even in a set of 42 focal samples.

Goodness of fit tests in this example have been calculated using only those trials with observed frequencies greater than zero. This is a requirement for G , as $\log(0)$ is undefined. The χ^2 test has been similarly restricted, so that it can be directly compared to G . However, if you wish to examine χ^2 results that compare observed values of zero with non-zero values from the simulated distribution, click on this option in the *Interval Analysis–Model Random Distributions* menu. A checkmark will appear. The default value is “off”, since small expected values, which often occur when observed values are zero, render the probability of χ^2 inexact.

Refining the search for a cut point.

In this section, we continue the example above, illustrating how to combine analyses across families or cases.

Having decided to investigate cut points of 90 and 120 seconds, we return to the original output (pages 80-81) to confirm that 90 and 120 are reasonable choices as cut points – that there are reasonable gaps in the distribution at these points. Then, examining the intervals between 60 and 120 seconds in detail, we find 38 intervals in this range (20 from 60 to 90 seconds, and 18 from 90 to 120). As it happens, nearly half of these are contributed by three families, with no other families contributing more than two critical intervals. We therefore begin by focusing on these families.

Using the procedures described earlier, we score the data record for each family

The screenshot shows a spreadsheet with the following data:

	A	B	C	D	E	F	G	H
1								
2	cut point	60-sec		90-sec		120-sec		
3	Family	G	df	G	df	G	df	
4	3			4.43	4	0.58	3	
5	19	5.50	4	3.63	3	3.67	2	
6	34	3.27	3	3.77	3	1.86	3	
7								
8								
9	Sum	8.77	7	11.83	10	6.11	8	
10	P-value	0.27		0.30		0.63		
11								

using cut points of 60, 90, and 120 seconds, tally the distributions at each cut point, and test their fit against random distributions based on the parameters observed in each case. Using procedures described by McDonald (2009), the resulting values of the maximum likelihood goodness of fit test (G) can be combined across families to give summary values for the three cut points.

As illustrated above, after each analysis G and its degrees of freedom were transferred to a spreadsheet. (There is no G for Family 3 at 60-seconds because there were no critical instances for this family in the interval from 60 to 90 seconds.) Because G is additive, we can sum these values (and their degrees of freedom) for each cut point, and use the χ^2 distribution to obtain p -values. (Thus $G = 8.77$ with $df = 7$, yields $p = .27$). Large values of p indicate that the distributions being combined have similar underlying expected values¹².

These results indicate that the distributions derived from a 120-second cut

¹² McDonald (2009) suggests that when p is non-significant, one should pool the distributions being compared. This is not possible in this case, as the expected values, being generated by random processes, are not precisely the same for each distribution. Thus it may be that G is not additive here after all.

point are quite similar to each other, and each is a reasonably good fit to a random distribution based on the parameters (number of episodes and focal samples) observed for those families. Moreover, the fit at 120-sec appears to be substantially better ($p = .63$) than for distributions based on the other cut points ($ps = .27$ and $.30$), suggesting that a 120-second cut point should be used in scoring the entire sample. The final step in this process would involve scoring the remaining families with critical cases in the 90 to 120 range, to confirm that in each case, scoring with a 120-sec cut point yields distributions with acceptable levels of fit.

Generating labeled versions of the data files

Having identified a reasonable cut point (among other criteria), one next needs to assess reliability for identifying episodes in the data files. Because assistants and colleagues would have to learn your numeric codes in order to identify episodes in your original Focal data, Lag can now make copies of these files in which the numeric codes have been replaced by the names (or labels) given in the person and behaviour recoding files (see page 42). Although these “label” files cannot be analyzed by LAG, they make it easy for those unfamiliar with your numeric coding scheme to inspect your data.

From the *Options Menu*, choose *Generate Labeled Data and Save in New Files*. The new files will be named *Labeled [original_file_name]*. If recoding is turned off (uncheck *Analysis – Do Combine Behaviour Codes*), the original, detailed codes will be transferred.

Locating all episodes.

The interval analysis results above include only those trials in which at least two criterion behaviours occurred. You may obtain a listing of all occurrences of the criterion behaviour from the menu *Interval Analysis – Locate All Instances of Criterion Behaviour*. This listing will reflect any recoding that is active.

Summary

To do an interval analysis, you **MUST**:

- Specify a file or set of files.
- Specify a criterion behaviour.

You **MAY**:

- Specify a criterion actor and/or a criterion target
- Combine codes for behaviours or persons
- Generate a list of all occurrences of the criterion behaviour.
- Generate models of the random distribution of n episodes across m trials
- Assess the fit of an observed distribution to a modelled random distribution.
- Designate whether the χ^2 goodness-of-fit test is restricted to intervals with observed frequencies greater than zero (the default) or includes zero values. (This option doesn't affect the calculation of G , as G requires observed frequencies greater than zero – $\log(0)$ is undefined.)
- Generate copies of the files being analysed with the original numeric codes replaced by the text labels specified in the behaviour and persons recoding files (*Options Menu – Save Labeled Data in New Files*)
- Request that new results replace or be appended to earlier results (*Options Menu*).

References

- Altmann, J. (1974). Observational study of behaviour: sampling methods. *Behaviour*, 49, 227-267.
- Bakeman, R., & Gottman, J. (1986). *Observing Interaction: An Introduction to Sequential Analysis*. Cambridge: The Cambridge University Press.
- Bakeman, R., & Gottman, J. (1997). *Observing Interaction: An Introduction to Sequential Analysis*, 2nd edition. Cambridge: The Cambridge University Press.
- Bakeman, R., & Quera, V. (1995). Log-linear approaches to lag-sequential analysis when consecutive codes may and cannot repeat. *Psychological Bulletin*, 118, 272-284.
- Conger, A. (1980). Integration and generalization of kappas for multiple raters. *Psychological Bulletin*, **88**, 322-328.
- DeVore, J. (1982). *Probability and Statistics for Engineering and the Sciences*. Monterey, CA: Brooks/Cole.
- Dixon, W. (1992). Chief Editor, *BMDP Statistical Software Manual*, Volume 1. Los Angeles, CA: University of California Press.
- Hayes, W. (1988). *Statistics*. New York: Holt, Rinehart, & Winston.
- McDonald, J.H. (2009). *Handbook of Biological Statistics* (2nd ed.), pp. 84-87. Sparky House Publishing, Baltimore, Maryland. Available on the web at <http://udel.edu/~mcdonald/statrepctestgof.html>
- Pepler, D., Craig, W., & Roberts, W. (1998). Observations of aggressive and nonaggressive children on the school playground. *Merrill-Palmer Quarterly*, **44**, 55-76.
- Roberts, W. (1993, March). *Programs for the Collection & Analysis of Observational Data*. Paper presented at meetings of the Society for Research in Child Development, New Orleans.
- Sackett, G. (1979). The lag sequential analysis of contingency and cyclicity in behavioural interaction research. In J. Osofsky (Ed.), *Handbook of Infant Development* (pp. 623-649). New York: Wiley.

Appendix

Installation

Observation_programs_nnn_install.exe is a self-extracting compressed (ZIP) file. Click on it in Windows Explorer. Accept or change the default directory *\Program Files\Observation Programs*; it will be created for you and the observation programs unzipped. A program group called *Observation Programs* will be created on your *Start - All Programs* menu. If you wish, you may place the two .DLL files in the *\Windows\System32* folder.

Before you run FOCAL.EXE , you will want to replace the contents of the default files *Default Behaviour Codes.txt* and *Default Codes for Persons.txt* with your own codes. You may use any word or text processor to do this (including Linux or Macintosh applications). Don't insert tabs, and be sure to save the files as text files.

If you transfer FOCAL to a laptop, netbook, or tablet for field work, be sure to copy your behaviour and actor-target code files and the two Dynamic Link Library files that are included – XNMBA458.DLL and XNMTE458.DLL.

When they are run, these programs will store information about user preferences in a small text file, *Observation programs configurations.txt*, in a directory called *C:\Observation Programs*

Uninstalling the programs

You can uninstall the programs by deleting the directory in which they were placed (see above) and the directory containing the configuration file (*C:\Observation Programs*)¹³. If you moved XNMBA458.DLL and XNMTE458.DLL to the *\Windows\System32* folder, you will need to delete them from there.

Program limitations

FOCAL and KAPPA handle up to 100 behaviour codes (0 to 99) and 100 actor-target codes (0 to 99). Data files of any length may be generated and processed.

¹³ Older versions of the programs (2.75 and earlier) placed the configuration file in the root directory, *C:*, but Windows Vista and 7 do not permit this for security reasons. If you used these versions, the configuration file may have been placed in a virtual folder – click on “compatibility files” in Windows Explorer to locate and delete it. For Windows XP, the old configuration file will be in *C:*.

LAG and SEARCH can analyse FOCAL data files of any length. However, the greater the number of unique actor-behaviour-target events, the more RAM is needed for the Table of Comparisons. (Each event forms both a row and column in this table). Before analysis begins, the programs count the number of unique events, thus making the Table of Comparisons no bigger than it needs to be. TIMES reports the number of unique events in your data.

SEARCH will find an unlimited number of contingent sequences, of any specified maximum length.

COUNT will process any number of sequences.

TIMES can handle several thousand unique actor-behaviour-target combinations. FOCAL data files of any size can be processed.

SCAN will process FOCAL data files of any size.

Known issues

When a text editor is first opened (e.g., to view results of an analysis, a data file, or to make session notes), the keyboard may not respond -- the cursor will be stuck at the beginning of the file. A mouse click or touch pad tap anywhere in the text restores keyboard input to normal.

Technical Support

Free technical support is available from the author, Dr. William Roberts, Psychology Department, Thompson Rivers University, 900 McGill Drive, Kamloops, BC, V2C 0C8, Canada. Phone: (250) 828-5374; email: wroberts@tru.ca. Bug reports are welcome.