



RepGrid • WebGrid • RepGrids
RepNet • RepDoc • RepScript

Rep Plus

Conceptual Representation Software

RepGrids Manual

Managing and Analyzing
Collections of Grids

May 2021

Brian R Gaines and Mildred L G Shaw

<http://cpsc.ucalgary.ca/~gaines/repplus/>

Contents

Contents	i
1 RepGrids—Managing and Analyzing Collections of Grids	1
1.1 Methodology of multiple grid analyses—classes, common items	1
1.2 RepGrids tools	2
1.3 RepGrids files	3
1.4 Creating, saving and opening collections of grids	4
1.5 RepGrids window	4
2 Grids Pane	5
2.1 Opening a grid for analysis or editing	6
2.2 Selecting grids for analysis	7
2.3 Setting global parameters for analyses	7
2.3.1 Defining grid identifiers for analyses	9
2.4 Creating and analyzing composite grids	9
3 Socio Pane	13
3.1 Socionets panel	14
3.1.1 Socionets plot	14
3.1.2 Socionets data	17
3.1.3 Socionets outlier detection	18
3.2 Mode grid panel	20
3.2.1 Mode grid generation	21
3.2.2 Mode grid data	23
3.3 Grids with both elements and constructs in common	23
3.3.1 Socionets of grids with both elements and constructs in common	24
3.3.2 Mode grids when both elements and constructs are in common	25
3.4 Batch analysis panel	26
3.5 Class metagrids and selection panel	27
3.5.1 Creating a class metagrid	27
3.5.2 Loading a class metagrid	28
3.5.3 Analysis of a class metagrid	29

4	Histo Pane	31
4.1	Distribution panel: analysis of overall distributions	32
4.1.1	Mean grid generation	32
4.1.2	Mean and standard deviation data	33
4.1.3	Histograms of distributions	35
4.1.4	Histogram parameters panel	37
4.2	Comparing distributions	37
4.2.1	Compare two classes of distributions panel	39
4.2.2	Comparison histograms	39
4.2.3	Comparison data	45
5	Compare Pane	48
5.1	Compare two grids panel	48
5.2	Styles for grids and nets created in analysis panel	49
6	Content Pane	51
6.1	The interplay between content analysis and other grid analysis methods	51
6.2	Populating the content pane	52
6.2.1	Controls for hiding columns and reversing constructs, and popup menu . . .	55
6.2.2	Popup menu commands	55
6.3	Grouping similar items	56
6.3.1	Grouping through the popup menu	58
6.4	Exporting content analysis data	59
6.5	Numerical measures supporting content analysis	61
6.5.1	Honey/Jankowicz indices	62
6.5.2	Socio match and mode values	64
6.5.3	Aligning through the popup menu	68
6.6	Textual search	70
6.7	Classifying the groupings	72
7	Scripts Pane	75
7.1	Complexity script	76
7.2	Get Socio Data script	78
7.3	Grid Data script	82
7.4	Grid Overviews script	83
7.5	Bulk Editing Grids	84

8	RepGrids File Organization	86
8.1	RepGrids file pointers	86
8.2	RepGrids scripts and default grids and nets—GridsScripts directories	87
8.2.1	Structure of default grids and nets	88
9	Appendix 1: Data Sets Used in the RepGrids Manual	90
9.1	Study of a research group specializing in geographic mapping techniques	90
9.2	Study of a ballroom dance community	90
9.3	Study of product development in a manufacturing company	91
9.4	Study of knowledge engineering for expert systems	91
10	Bibliography	92

1 RepGrids—Managing and Analyzing Collections of Grids

Most studies based on the elicitation of conceptual grids result in a collection of data that includes many files: grids; analyses; research notes; and so on, that record the progress of the study and requires management. The RepGrids tool was designed to manage the grids elicited during the study and make them readily accessible to various analytic tools, both as individual grids and as stratified collections of grids.

RepGrids is a generic tool for managing arbitrary collections of grids, providing ease of access to analyses of particular grids (individually or in in batch mode), comparisons of pairs of grids, and analysis of sub-collections of grids that are related in some way. It also provides scriptable access to the grid data for purposes such as batch editing, custom analyses and export to other programs. It is the primary tool for analyses of combinations of grid with elements, constructs or both in common, and for content analysis of grid relevant to a common domain.

1.1 Methodology of multiple grid analyses—classes, common items

The methodological framework for grid comparison discussed in the RepGrid manual and its discussion of common elements and constructs applies to most of the multiple grid analyses in RepGrids, many of which may be seen as extension of the pairwise comparison methods to multiple grids that have some commonality. In particular the diagrams used in that manual may be extended to apply to multiple grid analysis methods.

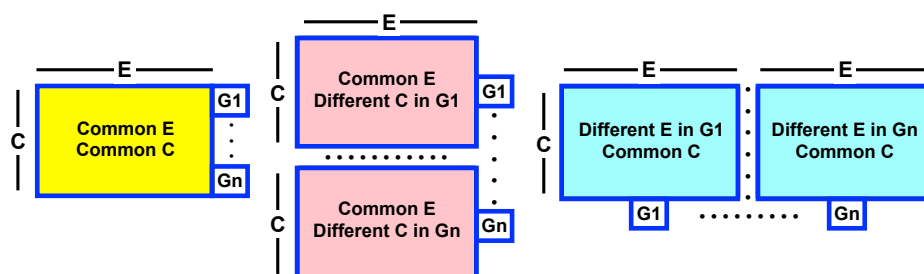


Figure 1: Comparison of multiple grids with some or no common elements and/or constructs

The underlying methodological principles remain the same: that there may be elements in some subset of the grids with lexically identical names, constructs having the same name and pole names, and it may be hypothesized that the community under study intends items with the same names to be coreferential.

RepGrids identifies commonality in the selected grids dynamically showing the number of common elements and constructs as soon as a set of grids is selected, and making the same information available in all analysis panes. The lexical basis of the identification is not problematic if grids have been generated with common items using RepGrid's capabilities to copy a grid in part with only elements, only constructs, or both but without ratings. However, if grid data from a study has been entered manually then errors in the entry of element and construct identifiers may cause the expected identification to fail, but this is readily seen and traced through the dynamic counts.

An additional type of association significant for larger sets of grids is that between those from whom the grids have been elicited, that they may be classified along different dimensions such as job role, gender or education, and that the analyses should be stratified along these dimensions. RepGrids allows an arbitrary subset of grids to be selected for analysis, and this can be controlled by *class metagrids* representing stratifications of the community being studied.

Some analyses in RepGrids are possible when grids have no computer-detectable commonality but possible human-detectable relationships, others when they have either elements or constructs lexically in common, and others only when they have both. Examples are provided of all such possibilities.

There may also be no commonality between the grids in a RepGrids collection, just the convenience of being able to view them together as a set of grid data and process any of them individually. However, there is usually some association between the grids in a collection, for example, that they are part of the same study or that they are not only to be analyzed together but also in combination.

1.2 RepGrids tools

The **Grids** tool supports managing the collection, setting global parameters, defining the form of identifier for each grid that will be used in the analyses, and generating composite grids from selected grids with common elements or constructs.

The **Socio** tools provide current versions of Shaw's (1980) *SOCIOGRIDS* analyses which generates **socionets** based on the strength of matches between grids, and **mode grids** based on the constructs (for grids with common elements) or elements (for grids with common constructs only) that are used, or construed, similarly by those in the selected community.

The **Batch analysis** tool enables the main *RepGrid* analyses (*Synopsis*, *Display*, *Focus*, *PrinGrid*) to be run on a selected set of grids.

The **Class metagrids and selection** tool manages one or more **class grids**, metagrids where the elements are the data grids in the collection and the constructs serve to classify those from whom the grid were elicited, in whatever terms the researcher wishes to use to stratify the community being studied. RepGrid *classes* may be used to specify intersects of these constructs that further extend the classification. The analysis tools in RepGrids can use the class metagrids to stratify the various analyses and the presentation of their results.

The **Defaults** tool manages the default grid and default net that are used as templates by the various RepGrids analyses, allowing the styling of the outputs from these analyses to be customized.

The **Histo** tools provide a family of techniques for analyzing the distributions of ratings when a sets of grids have both elements and constructs in common. They allow the distributions to be plotted as histograms, together with their means and standard deviations. When classes have been defined to stratify the community they allow comparative histograms to be plotted, together with statistics estimating the probability that they are drawn from the same distribution.

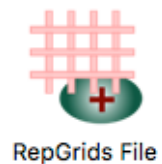
The **Compare** tool allows the **grid comparison tool** from RepGrid to be run on any pair of grids selected from the collection.

The **Content** tool supports the **content analysis** of constructs or elements from selected collections of grids which may have elements, constructs or neither, in common. In particular it supports the human capability to identify items that are similar in meaning even if they are not lexically identical, and to group these items for analysis as if they were a single item. It presents the elements or, more usually, the constructs in a spreadsheet-like format having outliner features, where rows representing similar items may be dragged together and grouped into a single representative item. It facilitates the process by making available measures such as the **centrality indices** specified by Honey (1979) and Jankowicz (2004) and the **mode and match indices** specified by Shaw (1980), if the data is such as to support their calculation.

The **Script** tool provides the capability to run programs written in **RepScript** that have full access to the data in all the grids in a collection. The analyses in RepGrids may be replicated in scripts and extended or new ones developed. This is the appropriate environment in which to develop multi-grid analyses and it is also useful for developing single grid analyses because they may be tested readily on collections of grids.

1.3 RepGrids files

RepGrids files save pointers to the grid files in a collection together with any parameters selected for the various analyses. When reopened the grid data and parameters are reloaded, enabling one to save work in progress and reopen it in the same state. Thus a RepGrids file behaves as if it were a database of grids, except that the grids themselves continue to be stored as separate files.



A particular grid file may be included in several RepGrids collections, and the grids in a collection may be stored anywhere on the computer. In practice, it is usually convenient to manage grid collections using the directory structure provided by the computer operating system, and the grids in a RepGrids collection are often in a single directory or in subdirectories of a single directory.

Such directories may contain other files associated with a project such as text documents, Rep Plus analyses, and so on. Rep Plus recognizes grid files and does not load files that do not represent grids.

1.4 Creating, saving and opening collections of grids

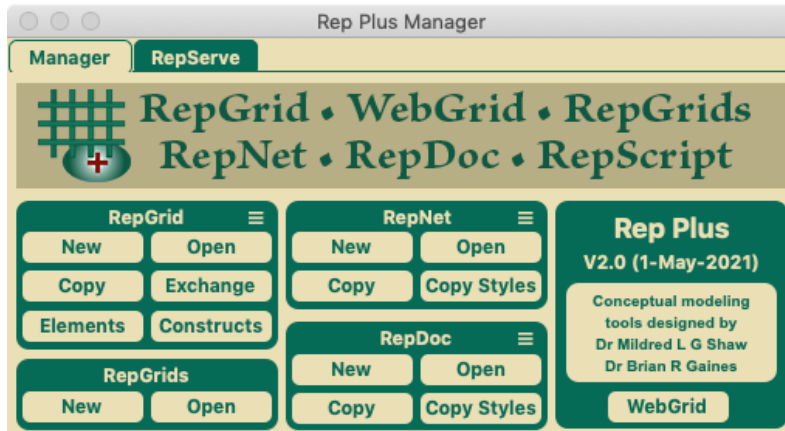


Figure 2: The **RepGrids** panel in the Rep Plus Manager window

A new RepGrids window may be created from the Rep Plus Manager window (Figure 2) by clicking in one of the buttons in the RepGrids panel: **New** to create an empty RepGrids window; or **Open Files** to create a RepGrids window and load any grid files from a specified directory—this can also be achieved by dragging the directory icon to the Manager Window or the Rep Plus application icon.

A RepGrids window may be saved at any time using the usual file menu options **Save** and **Save As...** commands. The window position and size are also saved but moving and resizing the window are not treated as undoable actions so the **Save** options is always active enabling a change in the window geometry to be saved if desired.

A RepGrids file can be opened in the usual ways: by double-clicking its icon; by dragging it to the Manager window or the Rep Plus application icon; by clicking **Open** in the RepGrids pane of the Manager window; by using the File menu commands **Open Recent** or **Open...**; or by keying command-O.

1.5 RepGrids window

Figure 3 shows the initial window when the directory entitled *Geog* containing a collection of files from a study of a small research group specializing geographic mapping techniques (Appendix §9.1) has been opened in RepGrids.

#	File	Name	Note	E	C	Date	Time	Place
1	100	Initial		0	0	1987-06-02	10:55	local
2	110	Peter		11	12	1987-06-02	12:29	local
3	120	Mary		11	15	1987-06-02	13:41	local
4	130	Charlie		8	9	1987-06-03	10:59	local
5	200	Charlie	Peter exchange	11	12	1987-06-03	11:29	local
6	210	Charlie	Mary exchange	11	15	1987-06-03	12:03	local
7	220	Mary	Peter exchange	11	12	1987-06-03	15:20	local
8	230	Mary	Charlie exchange	8	9	1987-06-03	15:47	local
9	240	Peter	Mary exchange	11	15	1987-06-03	12:03	local
10	250	Peter	Charlie exchange	8	9	1987-06-04	12:19	local
11	300	Agreed		12	0	1987-06-04	15:30	local
12	310	Peter	Agreed techniques	12	16	1987-06-04	15:36	local
13	320	Mary	Agreed techniques	12	18	1987-06-04	16:48	local
14	330	Charlie	Agreed techniques	12	14	1987-06-05	11:16	local
15	400	Peter	Agreed techniques	12	11	1987-08-19	14:30	local
16	410	Peter	Peter (Agreed techniques) exchange	12	16	1987-08-19	15:13	local
17	420	Mary	Agreed techniques	12	10	1987-08-20	14:49	local
18	430	Mary	Mary (Agreed techniques) exchange	12	18	1987-08-20	15:58	local
19	440	Charlie	Agreed techniques	12	13	1987-08-21	16:33	local
20	450	Charlie	Charlie (Agreed techniques) exchange	12	14	1987-08-21	16:53	local
21	500	Peter	Agreed techniques	13	14	1990-08-07	13:43	local
22	510	Peter	Peter (Agreed techniques) exchange	12	16	1990-08-07	15:58	local

Count same: 22 Grids 0 Elements 0 Constructs 0 Both Power: 1.0

Identifier: ☐ ID ☒ Name ☐ Note ☒ Date ☐ Time Item: ☐ Note ☐ #

Buttons: Add Grid, Add Grids, Composite

Figure 3: RepGrids window showing the **Grids** pane

The **Title** field at the top left is common to all the panes (except *Scripts*) and is used to identify the set of grids in some of the analyses. If the grids are added from a directory the title is set to the directory name but it can be changed appropriately.

The **Add Grid** and **Add Grids** button at the bottom right allow a single grid, or a directory of grids to added to the collection—dragging the grid or directory icon the RepGrids window is usually the preferred way of doing this. The **Composite** button merges a number of grids into a single composite grid (§2.4).

Along the top left are six tabs corresponding to the different tools described above. Clicking on any them shows a pane providing access to the specific tool or tool set. The features of each pane are described in the following sections.

2 Grids Pane

Figure 3 shows the **Grids Pane** that tabulates all the grids in the collection, controls their selection for some analyses and enables global parameters common to most analyses to be set.

For ease of manipulation and identification, the grids are shown in a tabular format by row number, file name, name and note in grid, number of elements and constructs, and the date, time and place of elicitation or entry.

2.1 Opening a grid for analysis or editing

Double-clicking in any row opens a copy of the grid file referenced in that row for examination and analysis on the assumption the intention is not to edit the original grid file. A note that the grid shown is a copy is placed in the *Grid Annotation field* ahead of any existing annotation (Figure 4).

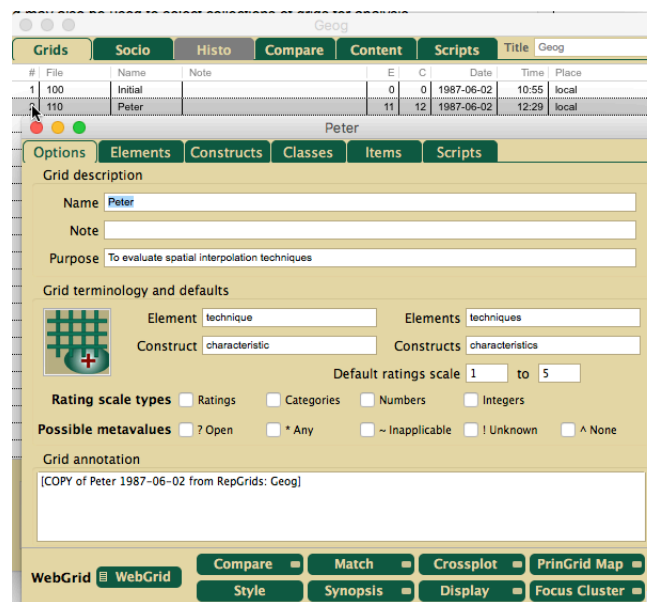


Figure 4: Double-click opens a copy of a grid for inspection and analysis

Right-clicking or CTL-clicking in a row brings up a menu providing an option to open a copy of the grid together with options that support the editing of the original grid (Figure 5):

Open copy Same as double-clicking;

Open actual Opens the grid file on disk for editing rather than a copy;

Refresh data Refreshes the grid data in RepGrids from the edited and saved grid files, however they were opened;

Refresh and renumber Refreshes the grid data and consecutively renumbers the list of grids.

Saving the RepGrids window also refreshes and renumbers the grids and is a simpler way to capture a series of changes after the list of grids, the grids, or both have been edited.

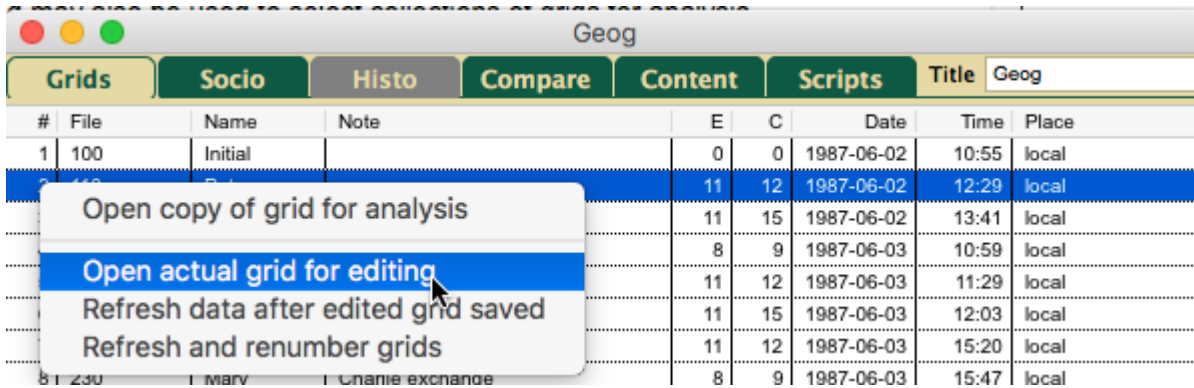


Figure 5: Right-click shows a popup menu supporting grid editing

2.2 Selecting grids for analysis

The various analysis tools in RepGrids generally use as data the collection of grids that have been selected in the Grids Pane, and various means are provided to manage such selection.

Grids may be selected manually by clicking on the rows in the table, and are initially all selected. Multiple rows may be selected by holding down the shift key for contiguous selections and the command key for non-contiguous selections, consistent with the normal conventions of the Mac and Windows operating systems. When a set of grids is added they are all selected automatically.

If metagrids have been used to classify the grid data (§3.5) then the classes defined may also be used to select collections of grids for analysis.

The table of grids may be sorted by the contents of any column by clicking on the title at the head of that column. Clicking when the column is already selected reverses the sort order. Figure 6 shows the grids pane sorted by clicking on the *Note* header to bring the 7 grids with *Agreed techniques* together. Clicking again on this column will reverse the sort order. The row order may also be changed by selecting one or more rows and dragging them to another location.

The row numbers do not change when the row order is changed so that the effect of sorting or changing is apparent and can be reversed by sorting on the row number. However, if the window is saved and reopened from its file, or the *Refresh and renumber* option (§2.1) is selected then the rows numbers will be in numerical order.

2.3 Setting global parameters for analyses

The bottom panel of the grids pane (Figure 7) shows the dynamic count of the number of common elements and constructs in the selected grids and, if there are both, the radio buttons provide a global choice of whether analyses will be based on common elements, common constructs, or both. They are set initially to either the items with the higher count or to *Both* if there is a high proportion of both common elements and common constructs

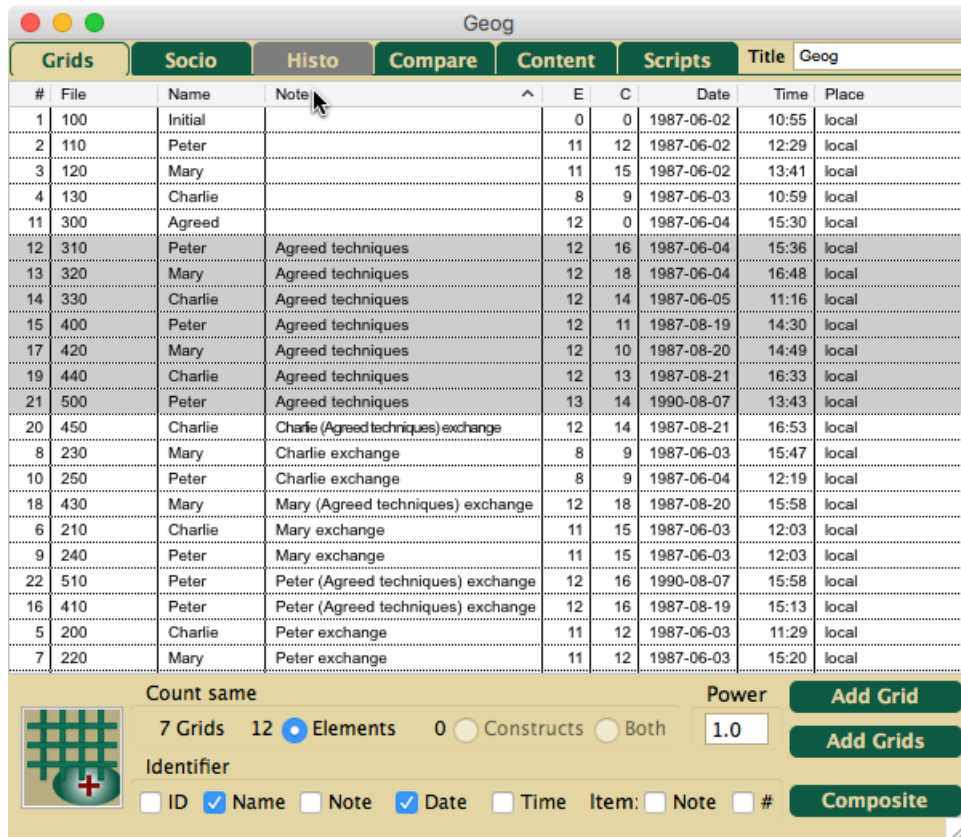


Figure 6: Clicking on a header to sort the grids by the data in a column

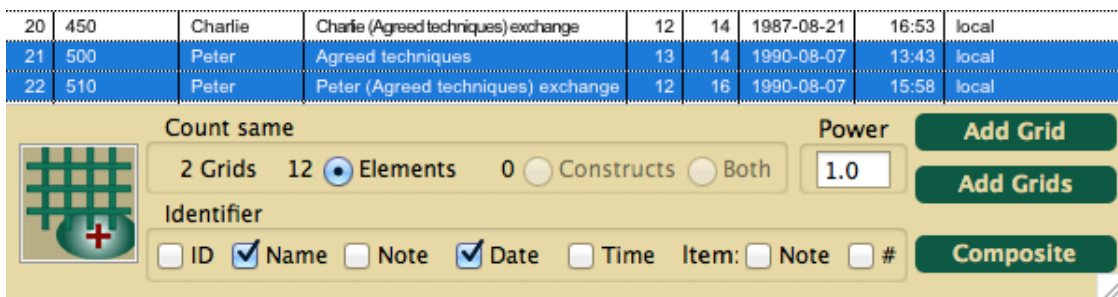


Figure 7: Global parameter setting in the bottom panel of the **Grids Pane**

The **Power** field sets the value of the of the Minkowski metric used in match-based analyses (as discussed in the RepGrid Manual). When the value is other than 1.0 it is shown in the headers of the **Socio** and **Compare** panes as a reminder of a nonstandard value, and it is also included in the titles of any analyses. This value is also transferred to grids created in the course of various analyses so that any further analysis of those grids will use the specified value.

2.3.1 Defining grid identifiers for analyses

In analyses of multiple grids it is often important to identify particular grids and to do so in a way that is significant to those interpreting the analyses, for example, by elicitee's name, an identifying note, or similar information.

The bottom row of the panel specifies the form of **identifier** to be used to identify grids in the various analyses. If no box is checked then **Name** is assumed by default. The same choices of **ID**, **Name**, **Notes**, **Date**, **Time** are provided as for RepGrid but there also options for additional annotation of the items, constructs and elements, with the first line of their note (if there is one), and with their number in the grid. This is particularly useful with some types of elements, such as the names of people, where, for example, a *role* entered in the note field may be more significant than the name.

The **ID** value will be the value of the **ID Item** in the grid if there is one and, if not, the file name. This is particularly useful with manually entered grid data where the name field may not have been used. The other options are useful if there are multiple grids from the same person and further identification through the note, date or time fields is needed.

2.4 Creating and analyzing composite grids

The **Composite** button creates a composite grid, of the common elements and all the constructs of the selected grids, or the common constructs and all the elements of the selected grids. It lists the grid identifiers in the **Grid Annotation** field, and puts the grid and item identification in the note field of the non-common constructs or elements, respectively for use in some analyses of the composite grid.

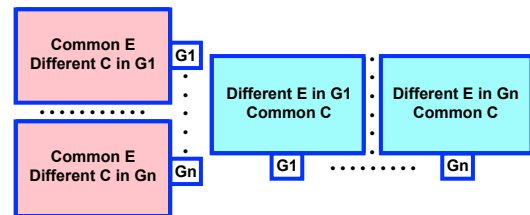


Figure 11 shows 7 grids having common elements selected to create a composite grid. Since some of the grids are from the same person different dates, the checkbox to add the Date to the identifier has also been clicked. Figure 9 shows the grid produced when the **Composite** button is clicked.

12	310	Peter	Agreed techniques	12	16	1987-06-04	15:36	local
13	320	Mary	Agreed techniques	12	18	1987-06-04	16:48	local
14	330	Charlie	Agreed techniques	12	14	1987-06-05	11:16	local
15	400	Peter	Agreed techniques	12	11	1987-08-19	14:30	local
16	410	Peter	Peter (Agreed techniques) exchange	12	16	1987-08-19	15:13	local
17	420	Mary	Agreed techniques	12	10	1987-08-20	14:49	local
18	430	Mary	Mary (Agreed techniques) exchange	12	18	1987-08-20	15:58	local
19	440	Charlie	Agreed techniques	12	13	1987-08-21	16:33	local
20	450	Charlie	Charlie (Agreed techniques) exchange	12	14	1987-08-21	16:53	local
21	500	Peter	Agreed techniques	13	14	1990-08-07	13:43	local
22	510	Peter	Peter (Agreed techniques) exchange	12	16	1990-08-07	15:58	local

Count same				Power		Add Grid
7 Grids	12 Elements	0 Constructs	Both	1.0		Add Grids
Identifier						
<input type="checkbox"/> ID	<input checked="" type="checkbox"/> Name	<input type="checkbox"/> Note	<input checked="" type="checkbox"/> Date	<input type="checkbox"/> Time	Item: <input type="checkbox"/> Note <input type="checkbox"/> #	Composite

Figure 8: Selecting 7 grids to form a composite based on common elements

Figure 9: Composite grid created from grids selected in Figure 3

The analysis tools in RepGrid allow the note fields for elements and constructs to be shown in parentheses in the various analyses, as illustrated in the top part of a Focus analysis of the composite grid shown in Figure 10. RepGrids automatically presets the parameters for showing the note fields for all RepGrid analyses in the grids it creates.

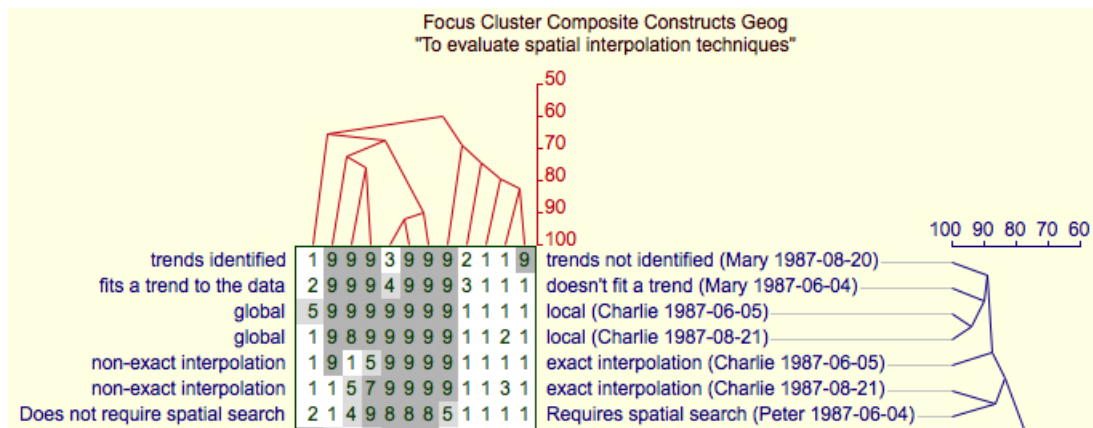


Figure 10: Part of Focus analysis of composite grid illustrating grid identifiers annotating constructs


The weights of items in the original grids are copied into the composite grid and hence may be used in later analysis. The weights for the common items are taken from the first of the selected

grids. If these weights are used in an analysis of the composite grid, it is important to ensure in the grid elicitation process that participants understand and use weight values in a mutually consistent manner.

The use of the **Compare** algorithms to compare grids may be complemented in RepGrids by using the capability to form a composite grid of a collection of grids with both common elements and common constructs and then analyzing those grids with the RepGrid and RepNet tools.

For example, in the *Geog* study there are two exchange grids elicited from Peter some 10 weeks (grid 16) and some 3 years (grid 22) after the original elicitation (grid 12). One can form a composite grid based on the common constructs in the original and later grids and perform a *PrinGrid* analysis. Figure 11 shows these 3 grids selected to form a composite grid.

12	310	Peter	Agreed techniques	12	16	1987-06-04	15:36	local
13	320	Mary	Agreed techniques	12	18	1987-06-04	16:48	local
14	330	Charlie	Agreed techniques	12	14	1987-06-05	11:16	local
15	400	Peter	Agreed techniques	12	11	1987-08-19	14:30	local
16	410	Peter	Peter (Agreed techniques) exchange	12	16	1987-08-19	15:13	local
17	420	Mary	Agreed techniques	12	10	1987-08-20	14:49	local
18	430	Mary	Mary (Agreed techniques) exchange	12	18	1987-08-20	15:58	local
19	440	Charlie	Agreed techniques	12	13	1987-08-21	16:33	local
20	450	Charlie	Charlie (Agreed techniques) exchange	12	14	1987-08-21	16:53	local
21	500	Peter	Agreed techniques	13	14	1990-08-07	13:43	local
22	510	Peter	Peter (Agreed techniques) exchange	12	16	1990-08-07	15:58	local



Count same

3 Grids 12 ☐ Elements 16 ☒ Constructs ☐ Both

Identifier

☐ ID ☒ Name ☐ Note ☒ Date ☐ Time Item: ☐ Note ☐ #

Power 1.0

Add Grid

Add Grids

Composite

Figure 11: Selecting 3 grids to form a composite based on common constructs

In RepNet one can then run the script *PrinGrid Trajectories* which joins up the locations in the plot of the elements having the same names but come from different grids (as indicated in the note field) to produce the plot shown in Figure 12.

PrinGrid uses the *Creator* item stored in the grid to identify it as one that was generated by the RepGrids *Composite* tool and sets the current script to of the net to be *PrinGrid Trajectories* so that a double-click in the net will automatically run the script. It can also be run from the popup menu.

3 Socio Pane

Clicking on the *Socio* tab in the RepGrid window with the 7 grids selected shown in Figure 11 brings up the pane shown below (Figure 13). It has four panels: *Socionets* and *Mode grid* that implement Shaw's (1980) *SOCIOGRIDS* collective analysis of multiple grids with common elements, constructs or both; *Batch analysis* that provides the capability to analyze multiple grids individually; *Class meta-grids* that use the constructs and classes in a grid whose elements are the grids in the RepGrids database to stratify those grids for various RepGrids analyses.

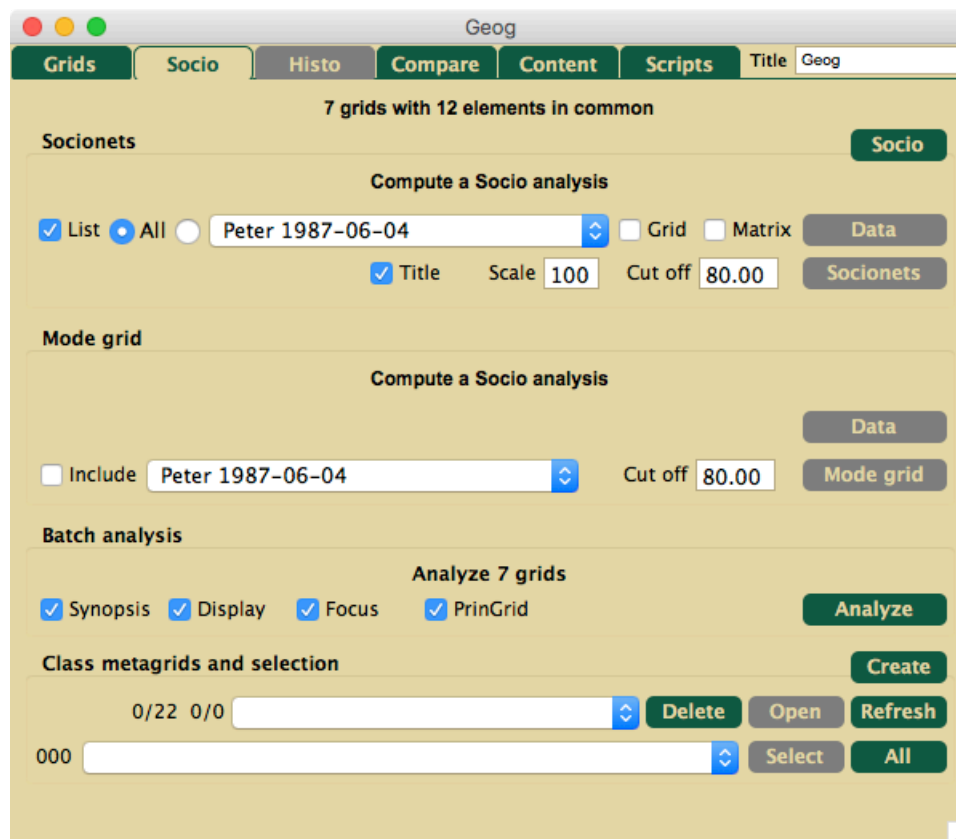


Figure 13: RepGrids window showing the **Socio** pane

The *Socionets* and *Mode grid* buttons are initially disabled until the *Socio* button at the top right has been clicked and a Socio analysis has been computed (and the *Socio* button disappears). If the grids selected or basis of comparison is changed the *Socio* button reappears to re-run the analysis but, otherwise, one can move freely between the panes, for example to change the form of grid identifier.

When the *Socio Analysis* button is clicked, RepGrids displays a progress bar whilst computing a Socio analysis and displays the results as shown in the panels below. Depending on the setting of the radio buttons and the *Power* value at the bottom of the *Grids* pane, similarities between the constructs, elements, or both in the selected grids are computed using the algorithms described in (Shaw, 1980), and these are used to determine the socionets and the mode constructs or elements, or

average grid, associated with the selected grids. The settings of the *Identifier* panel (§2.3.1) determine what annotation will be used to identify grids or items in the analyses.

3.1 Socionets panel

The **Socionets** panel (Figure 14) allows a network of grids to be plotted in RepNet that use the grid comparison match data to show the extent to which the items in one grid may be used to anticipate the items in another grid, the *comprehensibility* of another person's conceptual system, and the match data itself to be exported as text or used to create a class metagrid for outlier management.

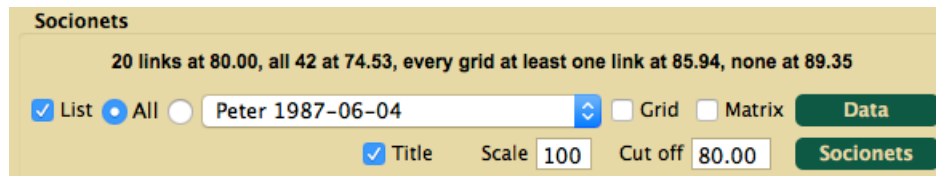
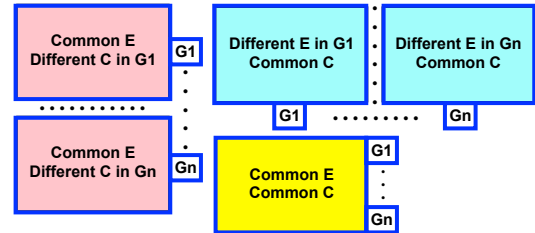


Figure 14: Socionets panel after a Socio analysis

The text at the top of the panel shows: the number of grids with links at the current cut off value; the cut off value at which all grids are linked; the maximum cut off value for every grid to have at least one link; and the cut off value at which no grids are linked.

3.1.1 Socionets plot

The bottom line of controls is used to manage the plot. The value of the cut off determines how great a similarity is needed before a link is shown initially, but the plot is dynamic and one can adjust the cut off in RepNet. The *Title* check box determines whether the plot is titled. The *Scale* value determines the size of the initial plot. The *Cut off* value determines the initial level of the link value below which links will not be made visible in the plot.

Clicking on the *Socionets* button generates a plot showing one of the family of possible socionets with the links visible whose values are not below the specified cut off (Figure 15).

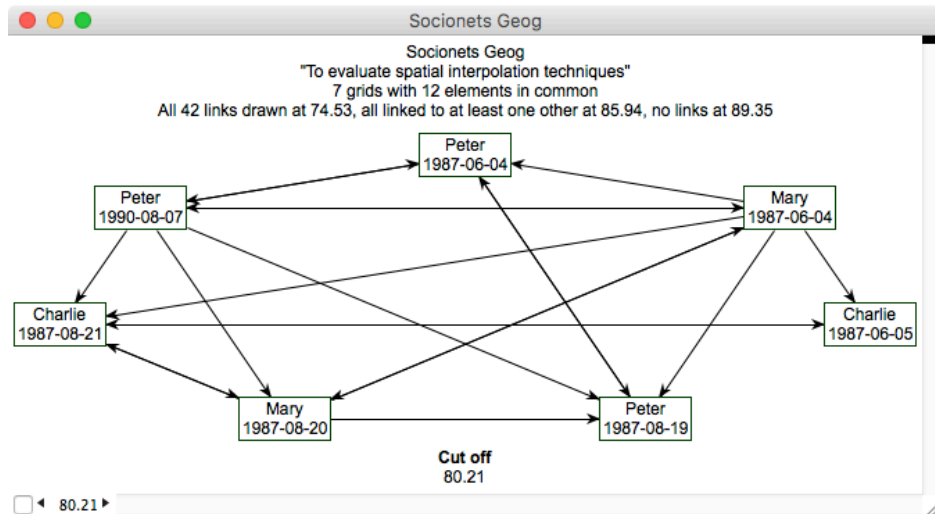


Figure 15: Generation of and interaction with a collection of socionets

The arrows from *Mary 1987-06-04* to *Peter 1987-06-04* indicates that the average over all the constructs in Peter's grid of the matches of the best-matched constructs in Mary's grid is greater than or equal to the cut off, indicating that Mary should be capable of understanding Peter's constructions. Clicking in the check box at the bottom left displays the actual match values as labels on the arrows with the value nearest an arrow head being that of the link to the grid at that arrow head (Figure 16).

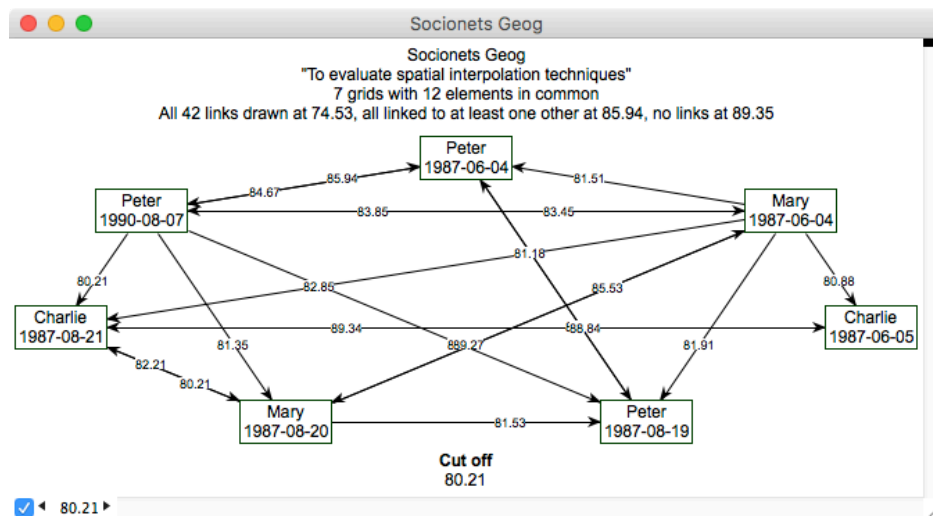


Figure 16: Socionets link values showing

The cut off is shown as part of the socionets plot the bottom centre and also in the text box at the bottom left. It may be adjusted by typing a different value in that text box, or by clicking on the triangular markers on either side of it to change to the next value of cut off that changes the number of links. The links come and go as the cutoff is changed This allows the socionets structure to be explored dynamically.

The plot is a net in RepNet and can be edited, saved in a file and opened again. One can drag the nodes around to make the plot more understandable. One can also copy and paste into a single net socionets generated with different cut off values (Figure 17).

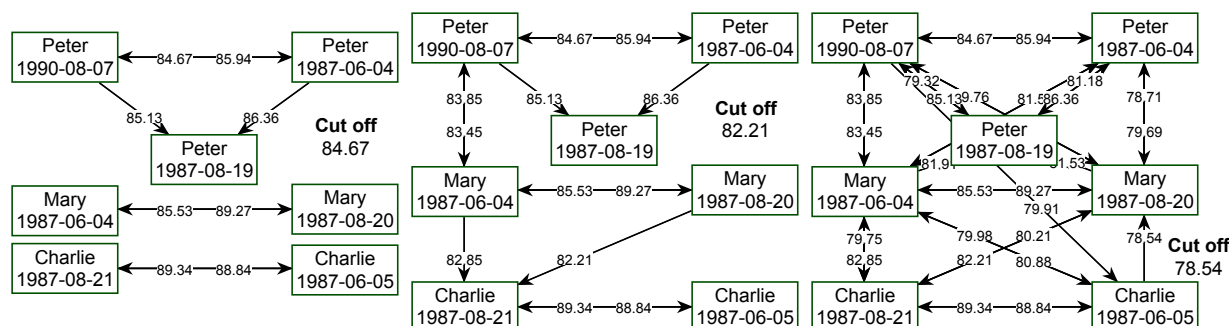


Figure 17: Three socionets at different cut offs

The centre plot in this diagram is the active one, and the ones to its left and right were created by copying and pasting the centre one at different cut off values. When pasted the plot becomes static and not subject to changes when the cut off is changed. The plot on the left was generated at the lowest cut off at which there are only links between the same individual at different times. It can be seen that, at this level, Peter in June 1987 is able to match his constructions in August 1987 and August 1990, and in August 1990 shows a similar pattern, but not in 1987, which corresponds to the reversals already noted in the PrinGrid trajectories.

The plot on the right was generated at the lowest cut off at which Peter and Charlie in 1987 are not connected, and shows the role of Mary as an intermediary between them at that time, with Peter in 1990 able to adopt a similar role to Mary.

The plot in the centre was generated at the lowest cut off at which Mary in 1987 could match Charlie's constructions in August, and also shows at that level a close relationship between her constructions in June 1987 and Peter's in August 1990.

Such an analysis of a set of grids generated by members of a group over a period of time on a topic central to the group's collaboration can be used to facilitate the group's discussion of its dynamics, and to provide insights into issues of communication and the different roles being played. It can also be used to model conceptual relationships and changes within the group.

The style of the socionets graphics is that of the default net specified for the RepGrids default net, and can be changed by choosing an alternative default net. It is apparent that every aspect of the presentation may be changed, for example, dashed lines, a different arrow head for the lower valued end of a symmetric link, the term "Cut off" changed to "Threshold", and so on.

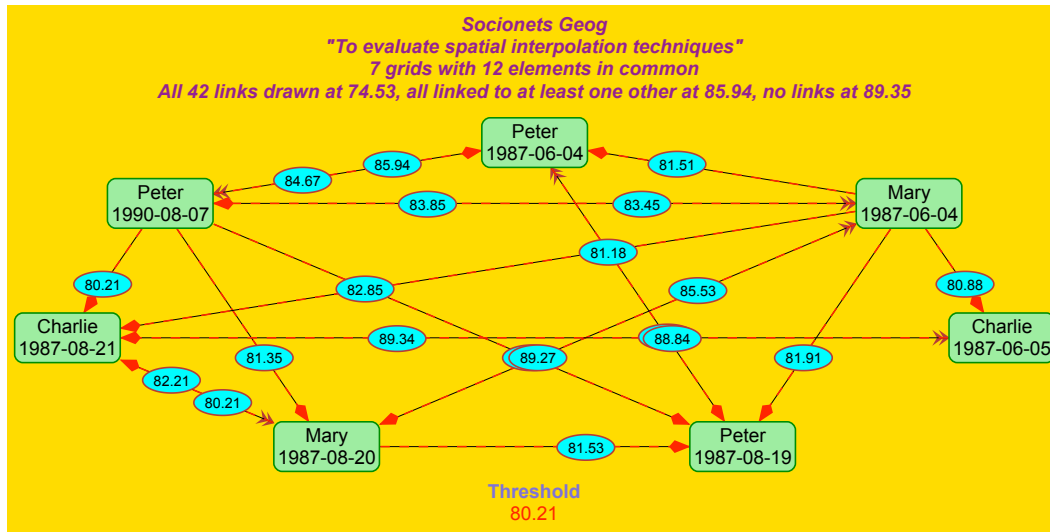


Figure 18: Socionets with a different default style chosen

A good technique for dragging the nodes to positions that make the graph structure more apparent is to start with a high cut off value where few nodes are linked, drag those nodes to be close together, and the lower the cutoff a few steps at a time dragging the newly linked nodes to appropriate positions and adjusting the previous nodes as necessary. Cliques of mutually connected grids become apparent as do lightly connected or disconnected grids.

3.1.2 Socionets data

The top line of controls in the *Socionets* panel (Figure 19) is used to manage textual output of the grid comparison data being used in the socionets plots. On the left the grid matches may be output as a sorted list, either for all the grids or just those for a selected grid. On the right the full matrix of match data may be output.



Figure 19: Socionets data panel

Figure 20 shows the matrix and the initial section of the sorted list.

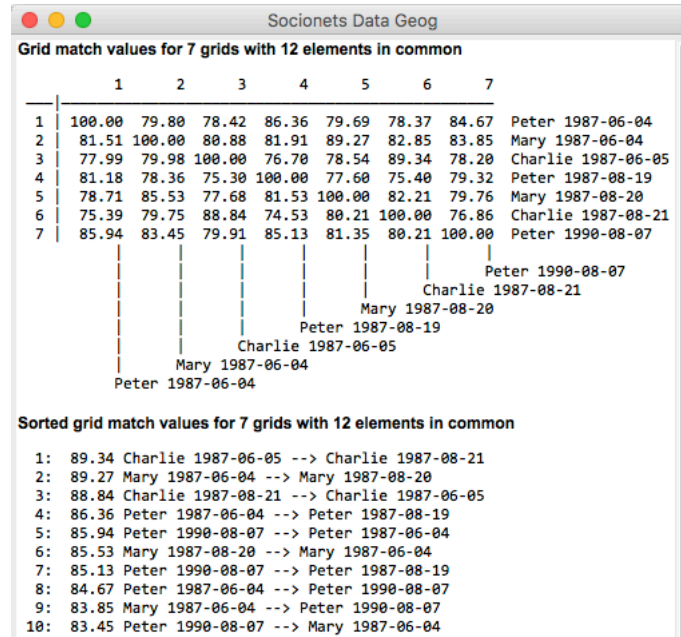


Figure 20: Socionets match matrix and links sorted by declining match

3.1.3 Socionets outlier detection

Outlier detection has long been treated as a major issue in data quality control from Tukey's (1962) *vacuum cleaner* techniques for removing residuals prior to analysis to modern developments where the detection and control of the effects of outliers are fully incorporated in the foundations of statistical methods (Aggarwal, 2013).

Conceptual grid studies are no exception, and one important use of the socionets match data is the detection of *outliers* in a collection of grids intended to be representative of some *community of practice* (Wenger, 1998). Large-scale community studies that collect tens or hundreds of grids to compare conceptual structures of sub-communities stratified in various ways need to ensure that all participants are representative of the communities, have understood what has been requested and are supplying data in good faith.

The Socio grid comparison data may be used to model the conceptual structures of a community in terms of the degree of mutual understanding of the domain under investigation and identify *cliques* of closely related construing as well as *outliers* whose interpretation of constructs expected or intended to be *common* is very different from other members of the community. However, diversity can a significant strength of a community or organization so that any statistical indication that a grid is an *outlier* needs to be investigated further to determine whether that is an indication of poor data or, for example, of an individual providing a different perspective significant for the community dynamics.

In particular, when comparing sub-communities it may be that an entire community is coherent in itself but an outlier to the community as a whole, for example one having particular expertise not generally shared, and it may be appropriate to stratify the outlier analysis by the same criteria used

to stratify the community. RepGrids provides the tools to do this but their use needs to be part of the management of the study by those with an understanding of its purpose and the dynamics of the community being studied.

One outlier detection technique is provide by the capability to output the matches of all the grids with a particular grid that is regarded as highly representative of the community, perhaps generated specifically to represent the expected or required norms of the community, or those of a highly regarded representative, or a *mode grid* (§3.2) representing the central tendencies of the community, or a particular sub-community.

The menu in the centre of the data panel (Figure 19) enables a reference grid to be selected, in this example the initial grid of the senior member of the group. If the radiobuttons are used to select this menu rather than *All* then the *List* checkbox outputs the matches of the remaining grids to that selected in decreasing order of match, and the *Grid* checkbox outputs a class metagrid (§3.5) with a single construct whose ratings are the matches to the reference grid. The values are initially categorized as *low* if they are below the cut off specified and *high* otherwise, but these categories may be changed and others added by editing the grid.

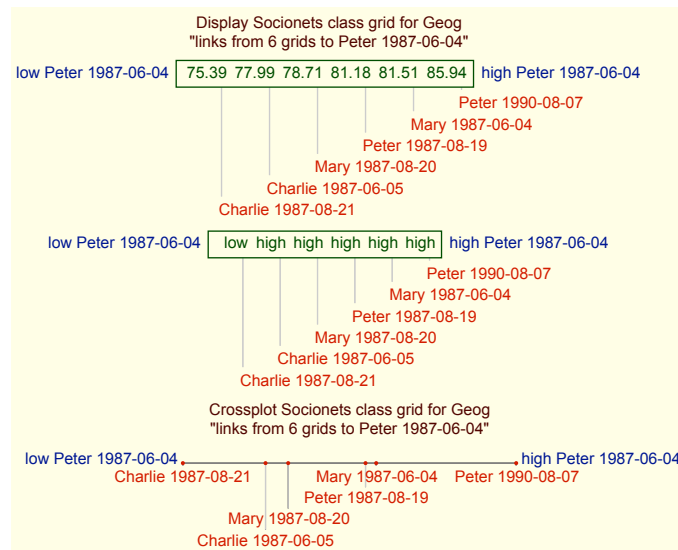


Figure 21: Display and crossplot of a Socio class metagrid

Figure 21 shows the numeric and categoric displays in RepGrid of a small class metagrid together with a crossplot graphic display of the data. The three presentations show the relation between the match data of Figure 20, the categories at a cut off of 76, and the outlier detection where *Charlie 1987-08-21* is clearly separated from the other grids.

This example, involving only 6 grids probably does not require outlier detection. Figure 22 shows similar analysis of a larger community where the technique may be useful. The grid A10 appears as a definite outlier requiring further investigation as might the cluster from A29 to A32 with matches under 75%.

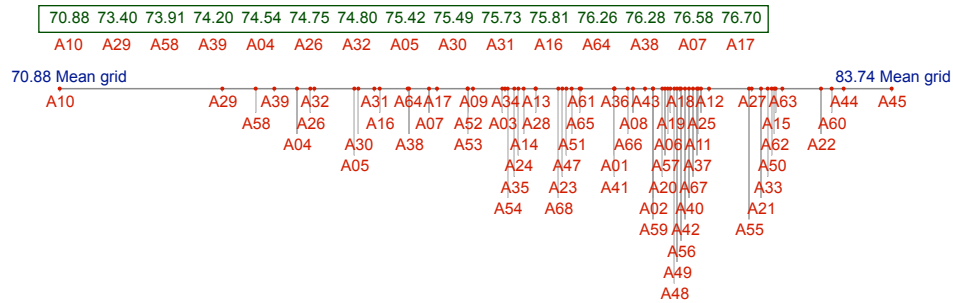


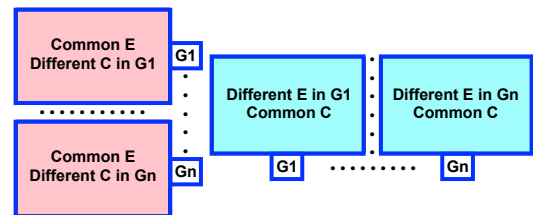
Figure 22: Display and crossplot of a larger Socio class metagrid

The metagrid may be added to the dataset and used to control the election of the grids being analyzed, enabling the sensitivity of results to the inclusion of outliers to be investigated. If a class grid stratifying the community is being used then a composite class grid may be generated and analyzed in RepGrid to investigate how the outlier construct clusters with the stratification constructs.

Class grids may be generated for different reference grids, for example, representing different sub-communities, and used to detect potential outliers from those specific sub-communities. These may also be merged for analysis with RepGrid tools and clustering with stratification class grids, and on on. They are important tools to support “drilling down” into a dataset to investigate hypotheses and increase understanding of the community being investigated.

3.2 Mode grid panel

The *Mode Grid* panel allows a composite *Mode Grid* (Shaw, 1980) to be created from the grids based on the constructs or elements most commonly used in all of them. The top line of controls is used to manage the creation of a mode grid. The number of mode items which will be included depends on the value set in the *Cut off* text box, and the message at the top of the panel indicates the number of items that will be included at the value entered together with the number for values either side of it. This helps the analyst determine an appropriate value for the cut off.



The mode score of an item is computed as the mean across grids of the highest match of that item to all items in every other grid, and measures the degree of consensuality of the item.

Figure 23: Mode grid panel after a Socio analysis

The *Include* check box on the left allows the grid selected on the popup menu to the right of it to be fully incorporated in the Mode Grid if required. The *Identifier* panel on the *Grids* pane provides the capability to annotate the items in the Mode Grid as already described.

3.2.1 Mode grid generation

Clicking on the *Mode Grid* button outputs the mode grid in RepGrid for analysis (Figure 24).

The screenshot shows the 'Mode Constructs Geog' dialog box. The 'Options' tab is selected. The 'Grid description' section includes fields for Name (Mode Constructs Geog), Note, and Purpose (To evaluate spatial interpolation techniques). The 'Grid terminology and defaults' section includes a grid icon, Element (technique), Elements (techniques), Construct (characteristic), and Constructs (characteristics). The 'Rating scale types' section has checkboxes for Ratings, Categories, Numbers, and Integers. The 'Possible metavalues' section has checkboxes for ? Open, * Any, ~ Inapplicable, ! Unknown, and ^ None. The 'Grid annotation' field contains the text: Peter 1987-06-04, Mary 1987-06-04, Charlie 1987-06-05, Peter 1987-08-19, Mary 1987-08-20, Charlie 1987-08-21, Peter 1990-08-07. The bottom of the dialog features a row of buttons: WebGrid, Compare, Match, Crossplot, PrinGrid Map, Style, Synopsis, Display, and Focus Cluster.

Figure 24: Generation of a *mode grid*

As discussed for composite grids, the grid identifiers are entered in the **Grid Annotation** field, and the grid and item identification are entered in the note field of the constructs or elements, respectively for use in some analyses of the mode grid. The weights of items in the original grids are copied into the mode grid.

The generation of a mode grid is an important technique for extracting commonality in conceptual representations from a number of people, typically construing the same elements. Using PrinGrid to produce a principal components analysis of the mode grid in RepGrid provides the same type of analysis as a generalized *Procrustes analysis* (Gower and Dijksterhuis, 2004) of the grid data, but has the major advantage that the mode grid being analyzed is a composite of individual grids with no introduction of artificial ratings. A Focus analysis (Figure 25) of the mode grid can clarify how the constructs from particular individuals relate to one another and contribute to the commonality, and the identifiers appended to different items enable the sources of commonality and difference to be identified.

Three major clusters are apparent in the constructs and five in the elements. One problem with a basic mode grid analysis that is apparent is that very similar constructs may duplicate one another.

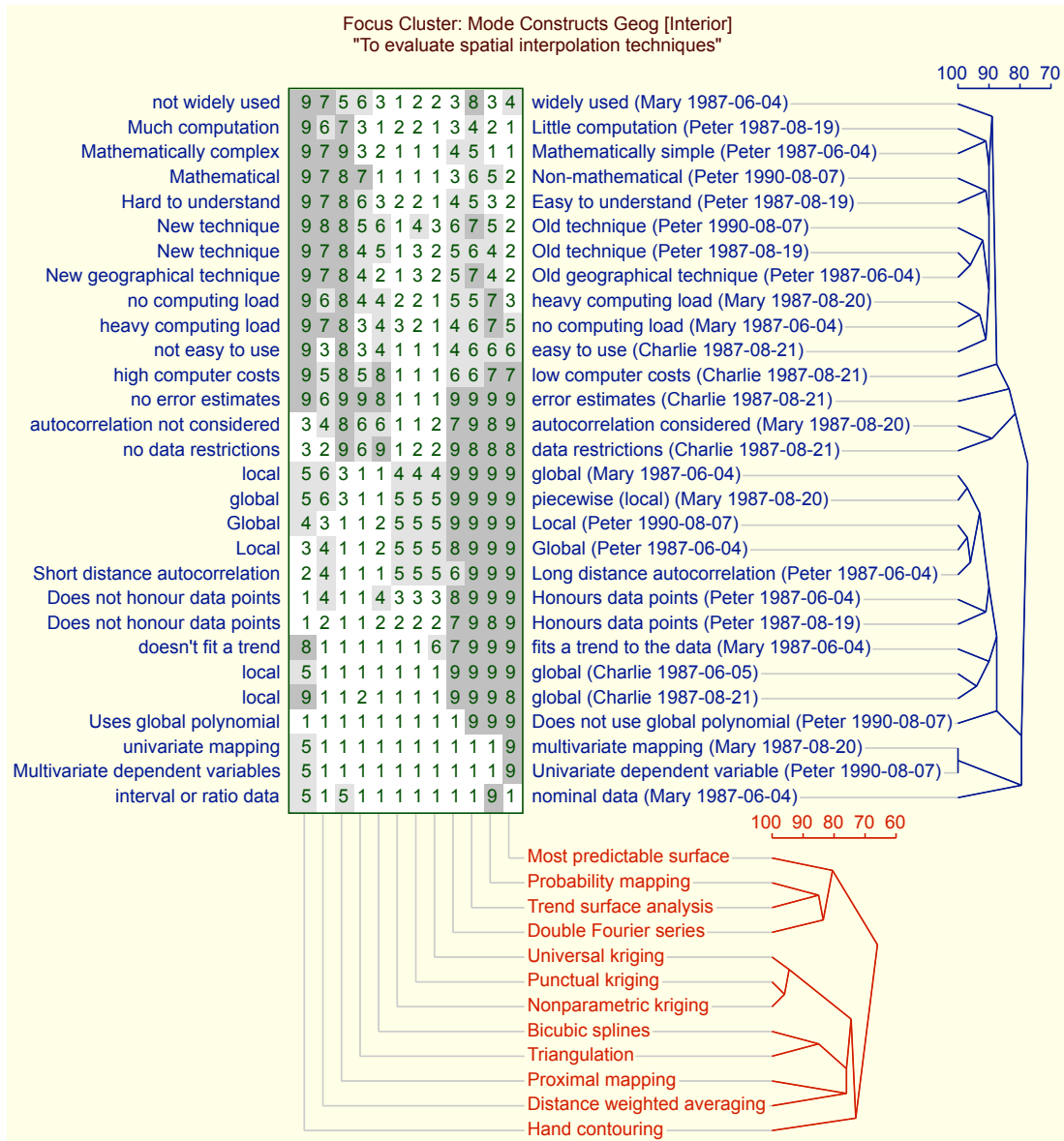


Figure 25: Focus analysis of a *mode grid*

In particular this is a problem in cluttering a principal components analysis to the extent it is difficult to comprehend. The *Content* tool (§6) may be used with mode grid data to group items that are similar textually and generate more perspicuous analyses. When, for example, constructs are grouped into a compound construct with averaged ratings, the construct weight corresponds to the number of items in the group, and a weighted analysis can be used to take into account the significance of that consensual construct across the community.

3.2.2 Mode grid data

The second line of controls in the “Mode Grid” panel is used to manage textual output of the mode items as list sorted by their average matches across all selected grids which provides a measure of their commonality. The list of mode constructs sorted by match values is useful in deciding where to set the cut off when creating a mode grid since it is possible to see the effect of various cut off values, and where natural gaps occur in the list. It also makes it apparent what grids are contributing to the mode and best reflect the consensus of the group. The sorted mode constructs are shown in part below.

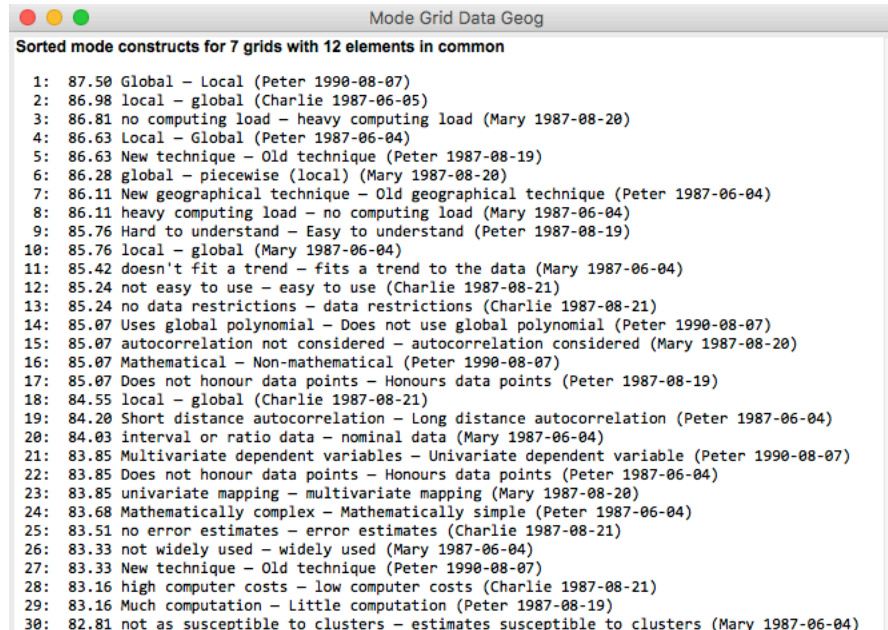


Figure 26: Mode grid constructs sorted by average match across selected grids

3.3 Grids with both elements and constructs in common

When grids have both elements and constructs in common it is possible to compare them as *exchange grids* based on the differences in the ratings on the common items. Socionets can be generated as before based on the overall matches between grids, but the natural measure of consensus across the group is now an average grid rather than a mode grid.

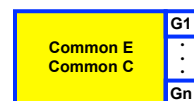


Figure 27 shows the *Grids* pane for data from *Project67* (Shaw and Gaines, 1991), a study in which members of a product development team developing microwave ovens used repertory grids in a pre-web distributed conceptual modelling tool in order to study their group processes.

The first grid represents the chief executive's assessment of a range of existing and potential products, and the four members of the product development team entered their own ratings of the elements on the constructs in that grid, adding extra items as they saw fit. The ninth grid is an average of the first eight grids created as described below and added to the set of grids so as to appear

Project67									
Grids		Socio	Histo	Compare	Content	Scripts	Title Project67		
#	File	Name	Note	E	C	Date	Time	Place	
1	Project67	CEO		8	10	1990-05-02	14:07:11	pd.microsim.com	
2	Marketing	Marketing		8	10	1990-05-10	14:06:44	sales.microsim.com	
3	Sales	Sales		8	10	1990-05-07	09:26:49	pd.microsim.com	
4	Technical	Technical		9	11	1990-05-04	10:07:04	pd.microsim.com	
5	Production	Production		8	10	1990-05-11	15:30:42	factory.micosim.com	
6	Starstream-PLM	Starstream PLM		8	10	1990-05-07	09:26:49	pd.microsim.com	
7	Starlight-PLM	Starlight PLM		8	10	1990-05-07	09:26:49	pd.microsim.com	
8	Heatfast-PM	Heatfast PLM		8	10	1990-05-07	09:26:49	pd.microsim.com	
9	Consultant	Consultant		17	16	1990-05-07	23:00:55	s2347.telesim.com	
10	Mean Project67 sta...	Mean Project67 staf...	staff is internal 8 grids	8	10	2017-09-09	12:35:29	192.168.0.100	

Count same

10 Grids

8 Elements

10 Constructs

Both

Power 1.0

Add Grid

Add Grids

Identifier

☐ ID
☒ Name
☐ Note
☐ Date
☐ Time

Item: ☐ Note ☐ #

Composite

Figure 27: Project67 grids with common elements and constructs

in the socionets. The tenth grid is from an external consultant who both rated the chief executive's exchange grid and also greatly extended it with additional elements and constructs.

The **Count Same** panel shows the grids as having eight elements and ten constructs in common, and offers analyses based on the common elements, common constructs, or both, with **Both** initially selected as the most likely option.

3.3.1 Socionets of grids with both elements and constructs in common

The **Socionets** panel operates in the same way for grids with both elements and constructs in common. However, exchange grid matches are intrinsically symmetric so the plots have undirected lines rather than directed arrows and there is a single match value (Figure 28).

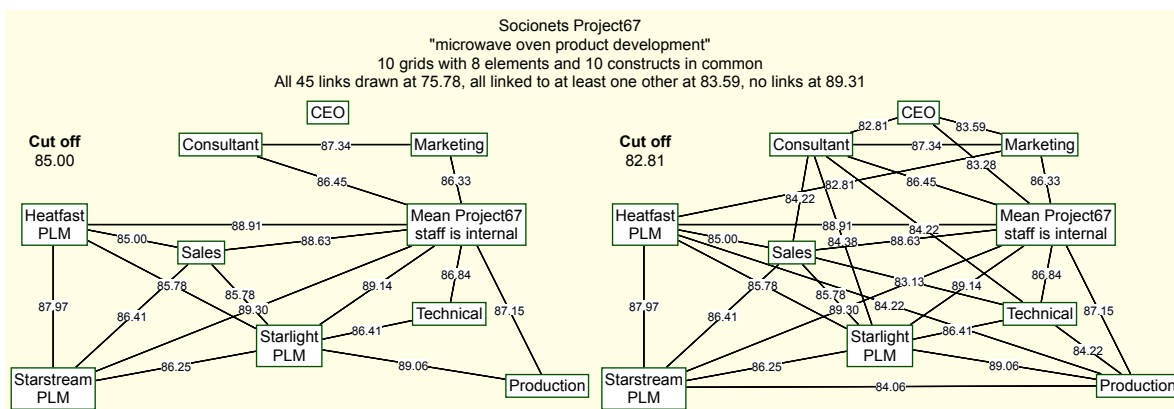


Figure 28: Socionets of Project67 grids with common elements and constructs

The nodes have been dragged to reduce line crossings and make the relationships more apparent. The socionet on the left shows the CEO disconnected, the consultant and marketing director form-

ing a clique with the average, the three product line managers and the sales director forming a clique with the average, and the technical and production directors each forming a clique with the Starlight product line manager and the average. The socionet on the right at a lower cut off value shows the CEO forming a clique with the consultant, marketing director and average. The new CEO who commissioned the study was appointed because the company was losing market share, and brought in a marketing consultant so the socionets reflecting commonality in construing are not surprising.

3.3.2 Mode grids when both elements and constructs are in common

For grids being analyzed with both elements and constructs in common the mode grid computation only uses matches between constructs having the same name. It also provides the capability to *fold* lexically identical constructs into a single construct having the mean of their ratings. It puts the number of constructs averaged in the *weight* field of the mean construct enabling the number of constructs represented to be used to weight analyses as if the constructs had been included rather than folded.

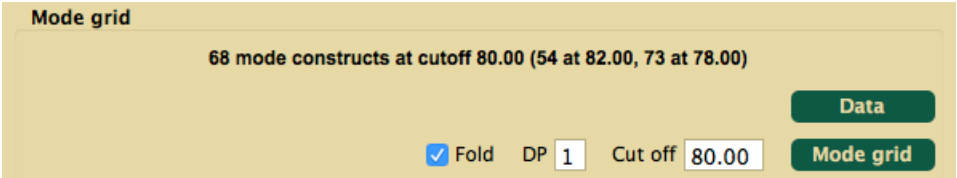
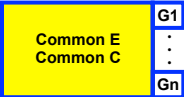


Figure 29: Mode grid panel for grids with both elements and constructs in common

The mode grid panel provides additional controls enabling folding to be selected and the number of decimal places in the mean ratings to be specified (Figure 29). If the number of decimal places is set to be zero, the mean values are rounded and the construct type is *ratings*, otherwise it is *numbers*. Figure 30 left shows the mode grid with DP set to 0, and right with it set to 1 and the weights shown.

Display Mode Constructs Project67 "microwave oven product development"										Display Mode Constructs Project67 [Weights] "microwave oven product development"									
stolid	4	2	7	6	7	9	8	8	fashion trend	stolid	3.9	1.9	7.2	5.6	6.8	8.8	8.0	8.1	fashion trend [8]
limited	6	2	7	7	8	8	9	9	versatile	limited	6.0	1.7	7.4	6.9	8.2	8.4	8.9	8.9	versatile [9]
high cost	5	9	2	4	7	6	6	8	low cost	high cost	5.0	8.9	1.9	4.0	6.6	6.3	5.9	7.9	low cost [7]
not so good value	7	8	2	8	9	8	8	9	good value	not so good value	6.6	8.2	2.1	7.8	8.6	8.0	8.4	9.0	good value [8]
weakly marketed	4	3	3	8	9	9	9	9	heavily marketed	weakly marketed	4.2	2.6	3.4	8.4	8.6	8.6	8.8	9.0	heavily marketed [5]
minimal functions	6	2	9	6	7	7	8	8	wide range of functions	minimal functions	5.6	1.9	8.6	6.0	7.4	7.4	8.2	8.1	wide range of functions [8]
lower inventory costs	4	2	8	5	5	7	5	1	higher inventory costs	lower inventory costs	4.4	2.3	7.6	5.4	4.6	7.2	5.3	1.0	higher inventory costs [9]
standard features	4	1	7	6	6	8	7	9	innovative features	standard features	3.5	1.4	6.8	5.8	6.2	7.8	7.2	8.6	innovative features [8]
complex production	7	9	2	5	6	5	5	9	simple production	complex production	6.5	8.7	1.5	5.3	6.3	5.3	5.2	8.7	simple production [6]
									ideal new product										ideal new product
									new Starlight + rotator										new Starlight + rotator
									new Starlight + colors										new Starlight + colors
									new Starlight										new Starlight
									market lead - Raymark										market lead - Raymark
									existing Starstream										existing Starstream
									existing Heatfast										existing Heatfast
									existing Starlight										existing Starlight

Figure 30: Mode grid panel for grids with both elements and constructs in common

If the cut off is set to zero then all the constructs of the selected grids will be included and the folded mode grid will be a mean grid representing the average of the ratings over all the grids selected. For grids having only elements or only constructs in common, lexically-based folding cannot be used but the *grouping* capability of the content analysis tool (§6) may be used to achieve folding manually based on human assessment that two constructs have very similar meaning.

3.4 Batch analysis panel

It is useful to be able to perform the standard analyses available in RepGrid with a collection of grids in RepGrids. The *Batch analysis* (Figure 31) panel enables the specified analyses to be output for the selected grids. This facility is particularly useful for quality control when a new set of grids is to be analyzed since the *Synopsis* tool provides a rapid appraisal of each grid. Figure 32 shows the four analyses produced of the mean grid from Project67 when the *Batch analysis* button is clicked.

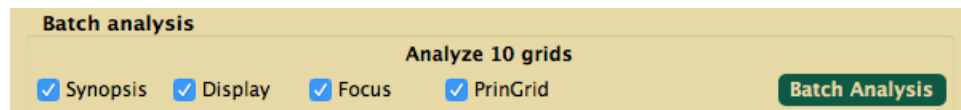


Figure 31: Batch analysis panel

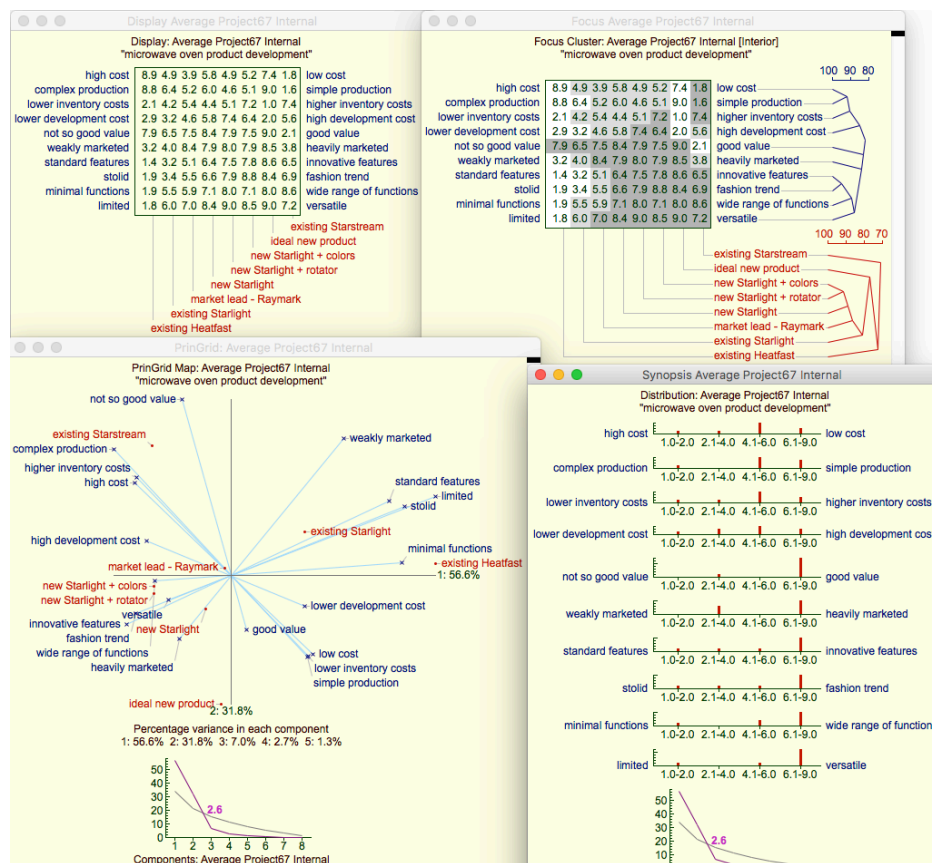


Figure 32: Batch analysis of mean grid for internal staff of Project67

3.5 Class metagrids and selection panel

It is useful with a collection of grids to be able to stratify the collection into various sub-collections by categorizing the grids/elicees along multiple dimensions. The stratification may reflect administrative issues of what grids were collected for what purpose, or demographic issues characterizing the population from whom the grids were elicited. It may be used to manage the selection of grids for analysis or be an integral part of the analysis itself, for example, in comparing different demographics, strata or other facets within the population.

The natural way to encode such classifications within the framework of Rep Plus is to create *class metagrids* whose elements are the grids within the collection being analyzed, and to enter the classification schema as a set of constructs for the categorization of these grids. The RepGrids tool recognizes such *metagrids* and makes the classifications they represent available to manage the selection of grids for analysis and the stratification of the grids within the analyses. The *Classes* tool in RepGrid enables secondary classifications to be developed from the primary ones though logical connectives.

The *Class metagrids and selection* panel (Figure 33) manages the creation and use of class metagrids applicable to the collection of grids in RepGrids.

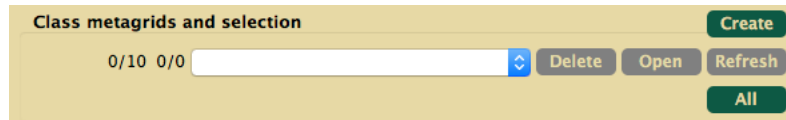


Figure 33: Class metagrids panel when no class grid has been loaded

3.5.1 Creating a class metagrid

If no class grid has been loaded then the only active controls are the *Create* button at the top right and the select *All* button at the lower right. Clicking on *Create* generates a skeleton metagrid having the IDs of the selected grids as elements (Figure 34).

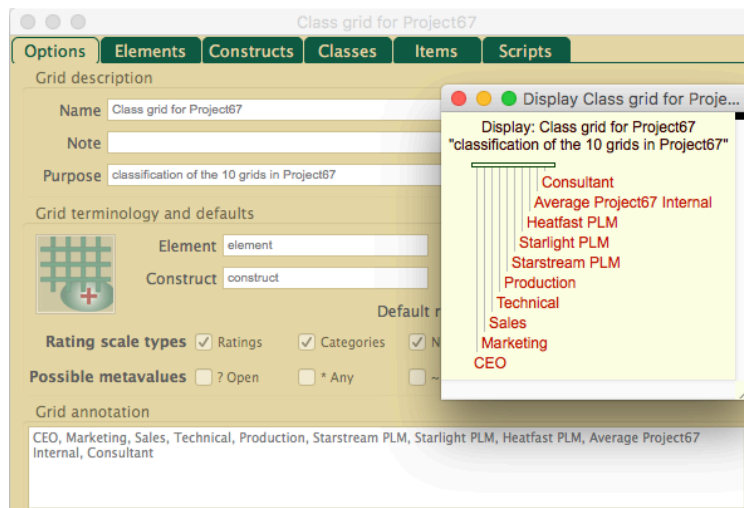


Figure 34: Skeleton class grid with grid IDs as elements

Figure 35 shows the simple class grid initially created to manage the Project67 data. The first construct recognizes that three of those participating come from a marketing background whereas the others primarily focus on the products being manufactured. The second distinguishes the internal staff from the external consultant. In addition the category *all* was added to the *staff* construct as the disjunction of *internal* and *external* to enable all the grids from staff to be selected whilst excluding computed grids such as mean or mode grids. The first construct is used primarily to manage analyses comparing subclasses of the community, the second to manage the different types of grid data.

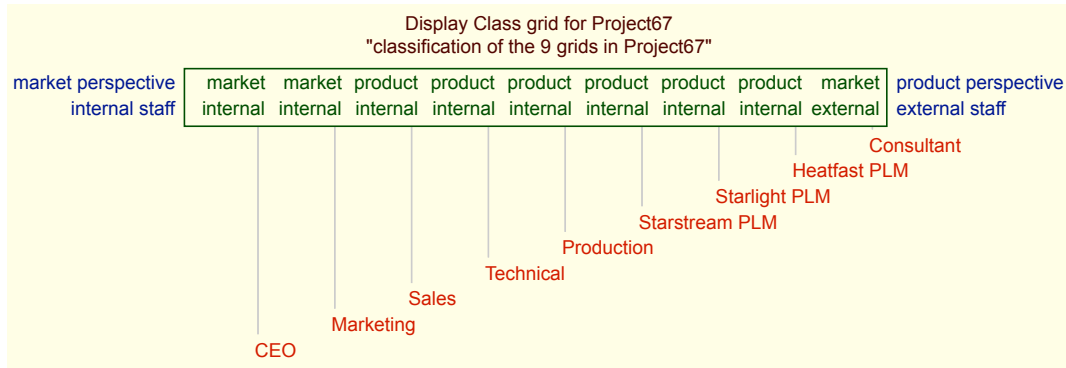


Figure 35: Class grid for Project67

The *Items* pane of the grid includes the item **ClassFlags** whose value specifies the checkbox settings in *Identifier* panel of the *Grids* pane. This is used to ensure the element names in the metagrid can always be linked to the grid names in the RepGrids collection. The existence of this item is also used by RepGrids to identify the grid as a metagrid and treat it differently from the data grids.

Class grid for Project67		
Options Elements Constructs Classes Items Scripts		
#	Name	Value
1	UID	003EE18F308DD9EA56EE0
2	Date	2017-08-07
3	Time	14:49:28
4	Place	192.168.0.100
5	ClassFlags	2

Figure 36: ClassFlags item identifying a metagrid type

Any type of construct may be used in a classes grid and the categories defined for these constructs are used to classify the grids in RepGrids. Often the “categories” type is most convenient but sometimes it is useful to enter a numeric value of a characteristic, such as age or years of experience, and stratify it by defining appropriate categories.

3.5.2 Loading a class metagrid

A class grid can be loaded in RepGrids in the same way as a data grid, for example, by dragging it to the RepGrids window. It becomes part of the grids database but appears in the *Class metagrids*

popup menu rather than the *Grids* pane (Figure 37). Several metagrids may be created, loaded and the active one selected in the popup menu.

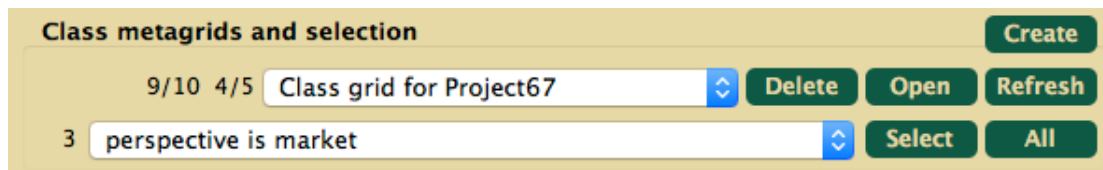


Figure 37: Class metagrids panel with a class metagrid loaded

The numbers on the left of the metagrids menu show the number of grids in class metagrid out of the total in the database followed by number of those in the metagrid that are selected out of the total selected in the *Grids* pane. This makes it apparent whether the coverage of the class metagrid is complete for all the grids, and whether it is complete for those selected. In this example the mean grid is not included in the class grid which is indicated in the first pair of numbers, and it is one of those selected which is indicated in the second pair.

All the buttons are now active: *Delete* removes the class grid shown in the menu from the dataset; *Open* opens it in RepGrid for analysis and editing; *Refresh* reloads it after it has been edited and saved.

The bottom menu shows a class and the number of grids that fall under it. The menu shows both any compound classes and the construct/category basic classes from all the metagrids that have been loaded.

The grids falling under the class shown may be selected in the *Grids* pane by clicking on the *Select button*, and all the grids may be selected by clicking on the *All button*. Selection dynamically updates all the RepGrids analysis panes so that, if the class grid supports the required selection, there is no need to return to the *Grids* pane to make a manual selection. The description of the class selected is stored and is appended to the title of the grids database to identify the class used in an analysis.

3.5.3 Analysis of a class metagrid

One advantage of representing the stratification of a community in a grid is that the usual grid analysis tools may be used to study the relations between the dimensions of the stratification system. If, for example, two apparently independent dimensions are highly correlated in the sample of the community selected for study it may be very difficult to distinguish between them, and it is important to know this if the results of the study are not to be potentially misleading.

Figure 38 shows the PrinGrid plot of the first two components of a principal components analysis of the class grid used to stratify a community of property appraisers in a major study (Bellman, 2012). There are 19 constructs in the class metagrid and the Frontier estimates on the scree plot suggest some 9 underlying significant dimensions with 3 major ones. This indicates that the dimensions of stratification are reasonably independent but some are correlated with others and this needs to be taken account in the interpretation of the results.

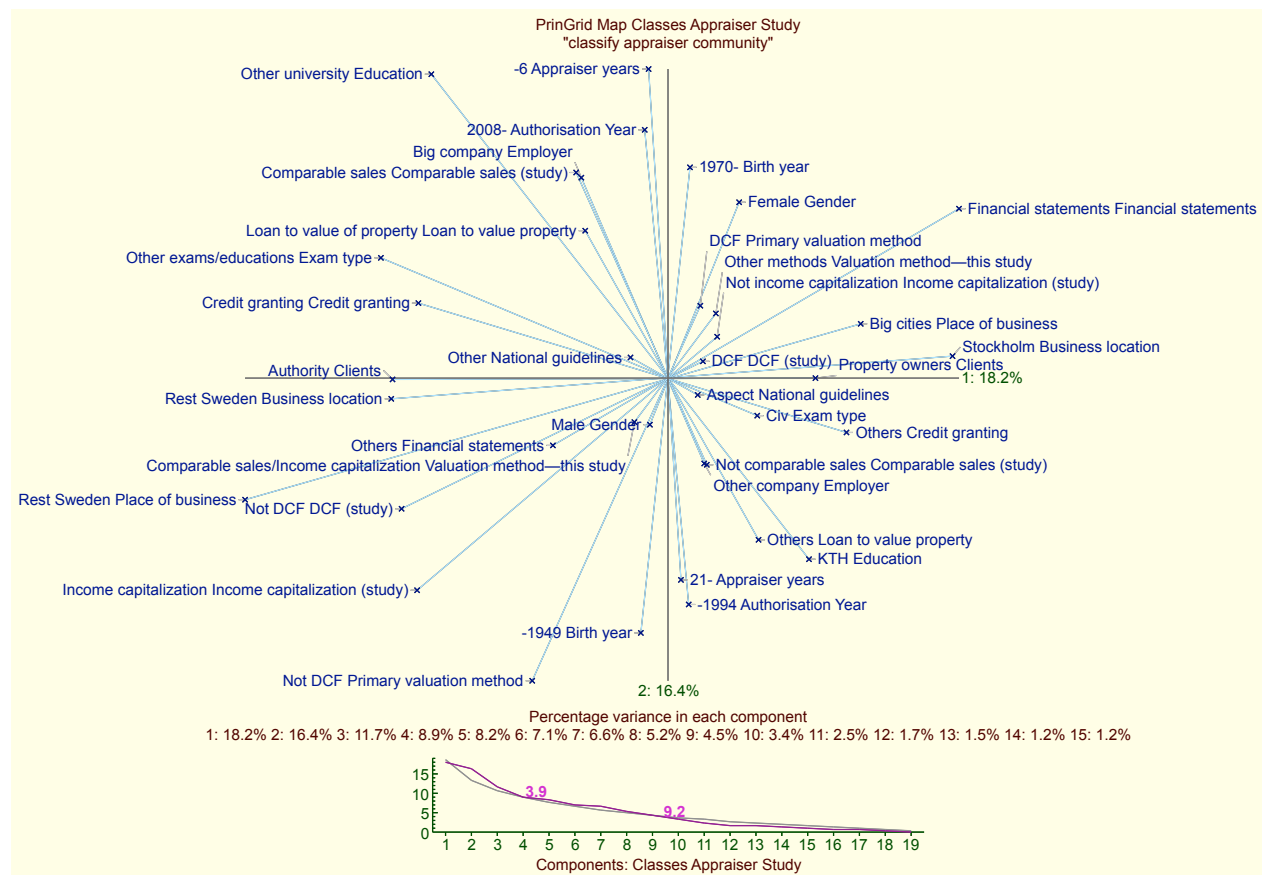


Figure 38: PrinGrid analysis of a class grid from a major study

4 Histo Pane

The *Histo* pane (Figure 39) supports analyses of collections of grids have both elements and constructs in common. The *Histo* tab is only active if *Both* is selected in the *Grids* pane. The top line of text shows the number of grids selected and the numbers of their common elements and constructs.

Common E	G1
Common C	⋮
	Gn

Project67

Grids Socio **Histo** Compare Content Scripts Title Project67

9 grids with 8 elements and 10 constructs in common

Distributions

80 distributions σ 0.00 to 3.18 DP 1 Mean grid

☒ Range σ 0.00 to 3.18 ☐ σ (SD) ☐ μ (Mean) ☒ σ μ sorted Data

By ☐ existing Starlight ☒ Sort Histograms

Histogram parameters

☒ Title ☐ Number μ σ ☒ Graph ☒ Values DP 2 Scale ☒ H ☒ V Scales 3

Compare two classes of distributions

☒ Probability $p \leq$ 2.5 % ☒ 2-sided ☒ Classes ☒ p ☒ d DP 2 Histograms

By ☐ existing Starlight ☒ Sort

3 perspective is market ☒ Link

6 perspective is product ☒ Link

☐ Classes ☐ p ☐ d ☒ d p sorted ☐ Grids $p \leq$ 10.0 % Data

☐ σ (SD) ☐ μ (Mean) ☒ σ μ sorted

Class selection

3 perspective is market ☒ Select All

Figure 39: RepGrids window showing the **Histo** pane

The pane has two panels: **Distributions** which provides various analyses of the distribution of ratings from different grids of an element on a construct construct; **Class selection** which, if one or more class grids have been loaded, allows the collection of grids being analyzed to be selected. The latter duplicates the same capability on the *Socio* pane to avoid changing panes whilst running a sequence of analyses from the *Histo* pane.

4.1 Distribution panel: analysis of overall distributions

The upper region of the distribution panel manages the analysis of the overall distribution of an element on a construct for all the selected grids. The top line of text shows the number of distributions under consideration and the range of their standard deviations (a full population SD rather than a sample population SD estimate—i.e. division by n rather than $n-1$).

4.1.1 Mean grid generation

At the right, the *Mean grid* button provides the capability to generate a mean grid from the selected grids. It is essentially a short-cut restricted duplication of the *Mode grid* functionality of the *Socio* pane with cut off set to zero. It is provided on the *Histo* pane because mean grids are a very common way of generating mode grids summarizing collections with both common elements and constructs.

The mean grid that is shown as the last item of the Project67 dataset (Figure 28) was generated from the grids of the 8 internal staff only. It was created by selecting the grids for the internal staff as shown at the bottom of Figure 40 where the *staff* item in the hierarchical menu allows the selection of *internal* (employees excluding the consultant), *all* (the grids elicited from people excluding the computed mean grid to which the *staff* construct is inapplicable) or *external* (the consultant).

8 grids with 8 elements and 10 constructs in common

Distributions

80 distributions σ 0.00 to 3.00 DP 1 **Mean grid**

☒ σ (SD) ☒ μ (Mean) ☒ $\sigma \mu$ sorted **Data**

Range σ 0.00 to 3.00 **Histograms**

By ☐ existing Starlight ☒ Sort

Histogram parameters

☒ Title ☐ Number $\mu \sigma$ ☒ Graph ☒ Values DP 2 Scale ☒ H ☒ V Scales 3

Compare two classes of distributions

☒ Probability $p \leq 2.5$ % ☒ 2-sided ☒ Classes ☒ p ☒ d DP 2 **Histograms**

By ☐ existing Starlight ☒ Sort

2 perspective is market ☒ Link

6 perspective is product

☐ Classes ☐ p ☐ d ☒ d p sorted ☐ Grids $p \leq 10.0$ % **Data**

☐ σ (SD) ☐ μ (Mean) ☒ $\sigma \mu$ sorted

Class selection

8 staff is internal **Select** **All**

perspective
staff
internal
all
external

Figure 40: Selecting a class to analyze

Clicking on the *Mean grid* button generates a mean grid as shown in Figure 41. The selection criterion *staff is internal* is shown in the title, name and note. The number of grids is also shown in the note, and their identifiers are listed in the annotation.

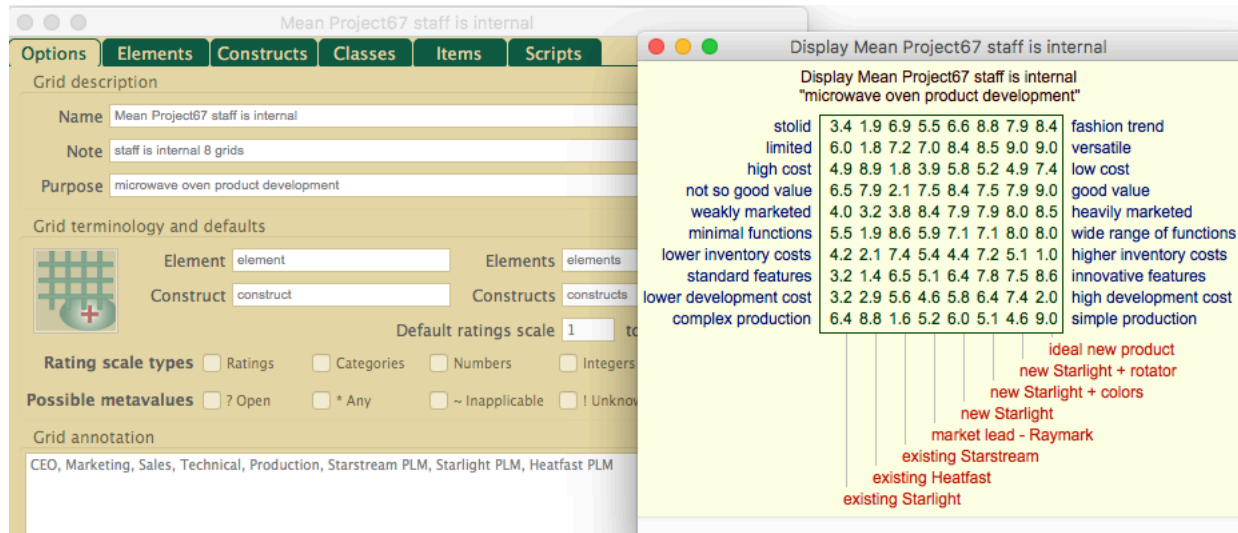


Figure 41: Mean grid for Project67 internal staff

Thus, mean grids for different communities are readily generated and identified. The full *Mode grid* functionality of the *Socio* pane remains available to investigate the sensitivity of such a conceptual summary grid to considerations of the representativeness of the grids on which it is based. In general the effect may be expected to be low because the averaging of ratings reduces the impact of idiosyncrasies but, if the analysis of the mode/mean grid is central to the conclusions of a study it may be appropriate to check that this is so for the data analyzed.

4.1.2 Mean and standard deviation data

On the second row, the *Data* button, depending on the check-box settings, generates textual output of the matrices of standard deviations and means for the grid ratings and a list of standard deviations and means sorted by standard deviation (Figure 42). This last makes the construct-element pairs where there is most and least consensus readily apparent.

Note that the *standard deviation* (SD) is provided as a descriptive statistic in RepGrids to produce comparable estimates of the degree of consensus in the rating of particular elements on particular constructs. It is the maximum likelihood estimator (Hedges and Olkin, 1985) of the actual population SD (divide by n) appropriate to a descriptive statistic rather than Gauss/Bessel unbiased estimator (Kenney and Keeping, 1951) of the SD of the population sampled (divide by n-1) appropriate to t-tests assuming a normal distribution. Grid rating distributions are generally highly skewed, in part because of the limited range of the scale, and far from normal making estimates of the significance of differences based on t-tests highly misleading (Wilcox, 1998).

Mean ratings Project67 staff is internal 8 grids with 8 elements and 10 constructs in common

	1	2	3	4	5	6	7	8	
1	3.38	1.88	6.88	5.50	6.62	8.75	7.88	8.38	stolid – fashion trend
2	6.00	1.75	7.25	7.00	8.38	8.50	9.00	9.00	limited – versatile
3	4.88	8.88	1.75	3.88	5.75	5.25	4.88	7.38	high cost – low cost
4	6.50	7.88	2.12	7.50	8.38	7.50	7.88	9.00	not so good value – good value
5	4.00	3.25	3.75	8.38	7.88	7.88	8.00	8.50	weakly marketed – heavily marketed
6	5.50	1.88	8.62	5.88	7.12	7.12	8.00	8.00	minimal functions – wide range of functions
7	4.25	2.12	7.38	5.38	4.38	7.25	5.12	1.00	lower inventory costs – higher inventory costs
8	3.25	1.38	6.50	5.12	6.38	7.75	7.50	8.62	standard features – innovative features
9	3.25	2.88	5.62	4.62	5.75	6.38	7.38	2.00	lower development cost – high development cost
10	6.38	8.75	1.62	5.25	6.00	5.12	4.62	9.00	complex production – simple production
									ideal new product
									new Starlight + rotator
									new Starlight + colors
									new Starlight
									market lead - Raymark
									existing Starstream
									existing Heatfast
									existing Starlight

Standard deviations Project67 staff is internal 8 grids with 8 elements and 10 constructs in common

	1	2	3	4	5	6	7	8	
1	1.22	0.60	1.69	1.50	1.11	0.43	1.36	1.32	stolid – fashion trend
2	1.00	0.97	1.79	1.32	0.48	0.50	0.00	0.00	limited – versatile
3	0.78	0.33	0.83	1.27	1.48	1.85	1.83	1.87	high cost – low cost
4	0.50	1.05	1.96	1.22	0.86	1.22	1.27	0.00	not so good value – good value
5	2.12	2.49	2.38	0.99	1.17	1.17	1.12	0.87	weakly marketed – heavily marketed
6	0.87	0.78	0.99	0.93	0.93	0.93	1.00	1.12	minimal functions – wide range of functions
7	1.39	0.78	1.49	1.11	0.99	1.48	1.05	0.00	lower inventory costs – higher inventory costs
8	1.39	0.99	0.50	1.96	1.22	0.97	1.58	0.70	standard features – innovative features
9	1.64	1.45	3.00	2.23	0.97	0.99	1.41	1.73	lower development cost – high development cost
10	1.80	0.66	0.86	2.05	1.58	1.69	1.87	0.00	complex production – simple production
									ideal new product
									new Starlight + rotator
									new Starlight + colors
									new Starlight
									market lead - Raymark
									existing Starstream
									existing Heatfast
									existing Starlight

Sorted standard deviations Project67 staff is internal 8 grids with 8 elements and 10 constructs in common

	σ	μ	Construct: Element
1:	0.00	9.00	complex production – simple production: ideal new product
2:	0.00	9.00	limited – versatile: ideal new product
3:	0.00	9.00	limited – versatile: new Starlight + rotator
4:	0.00	1.00	lower inventory costs – higher inventory costs: ideal new product
5:	0.00	9.00	not so good value – good value: ideal new product
6:	0.33	8.88	high cost – low cost: existing Heatfast
7:	0.43	8.75	stolid – fashion trend: new Starlight + colors
8:	0.48	8.38	limited – versatile: new Starlight
9:	0.50	8.50	limited – versatile: new Starlight + colors
10:	0.50	6.50	standard features – innovative features: existing Starstream
11:	0.50	6.50	not so good value – good value: existing Starlight
12:	0.60	1.88	stolid – fashion trend: existing Heatfast
13:	0.66	8.75	complex production – simple production: existing Heatfast
14:	0.70	8.62	standard features – innovative features: ideal new product
15:	0.78	1.88	minimal functions – wide range of functions: existing Heatfast
16:	0.78	2.12	lower inventory costs – higher inventory costs: existing Heatfast
17:	0.78	4.88	high cost – low cost: existing Starlight
18:	0.83	1.75	high cost – low cost: existing Starstream
19:	0.86	8.38	not so good value – good value: new Starlight
20:	0.86	1.62	complex production – simple production: existing Starstream
21:	0.87	5.50	minimal functions – wide range of functions: existing Starlight
22:	0.87	8.50	weakly marketed – heavily marketed: ideal new product
23:	0.93	5.88	minimal functions – wide range of functions: market lead - Raymark
24:	0.93	7.12	minimal functions – wide range of functions: new Starlight
25:	0.93	7.12	minimal functions – wide range of functions: new Starlight + colors
26:	0.97	1.75	limited – versatile: existing Heatfast
27:	0.97	5.75	lower development cost – high development cost: new Starlight
28:	0.97	7.75	standard features – innovative features: new Starlight + colors
29:	0.00	8.38	weakly marketed – heavily marketed: market lead - Raymark

Figure 42: Mean and standard deviation data (truncated) for Project67 internal staff

4.1.3 Histograms of distributions

On the next line, the **Histogram** button generates histograms of a set of distributions selected according to the criteria specified on the right: those whose standard deviations are in the range specified sorted by standard deviation (Figure 43); or those for a particular element or construct, sorted by standard deviation if “Sort” is checked but otherwise in the same order as in the grid (Figure 44).

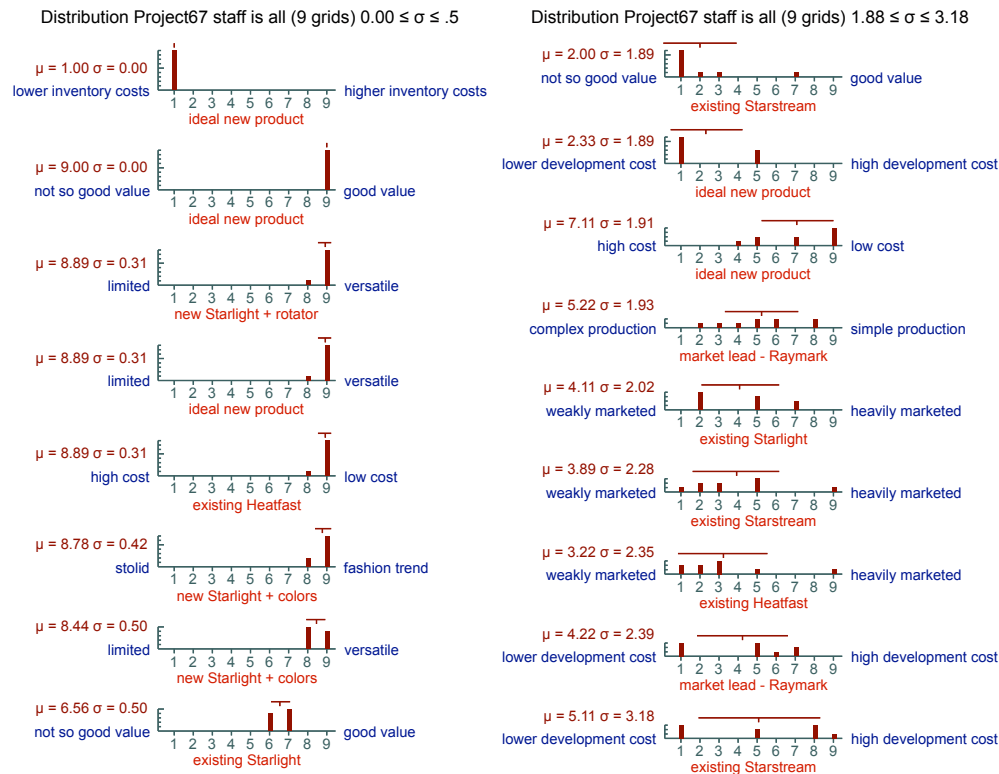


Figure 43: Lowest and highest distributions of Project67 staff ratings sorted by standard deviation

Histograms selected by standard deviation are useful to determine what ratings of any element on any construct have the greatest and least consensus. Those for a particular element enable the degree of consensus of the ratings of that element on every construct to be investigated, and those for a particular constructs enable the degree of consensus of the ratings of every element on that construct to be investigated.

The standard deviation a rough indicator of the degree of consensus is useful for filtering and sorting the results, but the primary value of the histograms is to be able to investigate the structure of the consensus. In a coherent community it will be expected that there will be a high degree of consensus on many items. If some histograms indicate that this expectation is not confirm, it may indicate a significant issue for the community appropriate for further investigation.

When the histograms are produced the RepNet net representing them is dynamically linked back to the dataset that produced them and the *HistoGrids* script is automatically run to provide an interactive interface to the histograms allowing the grids represented by each bar to be shown, and opened if required (Figure 45).

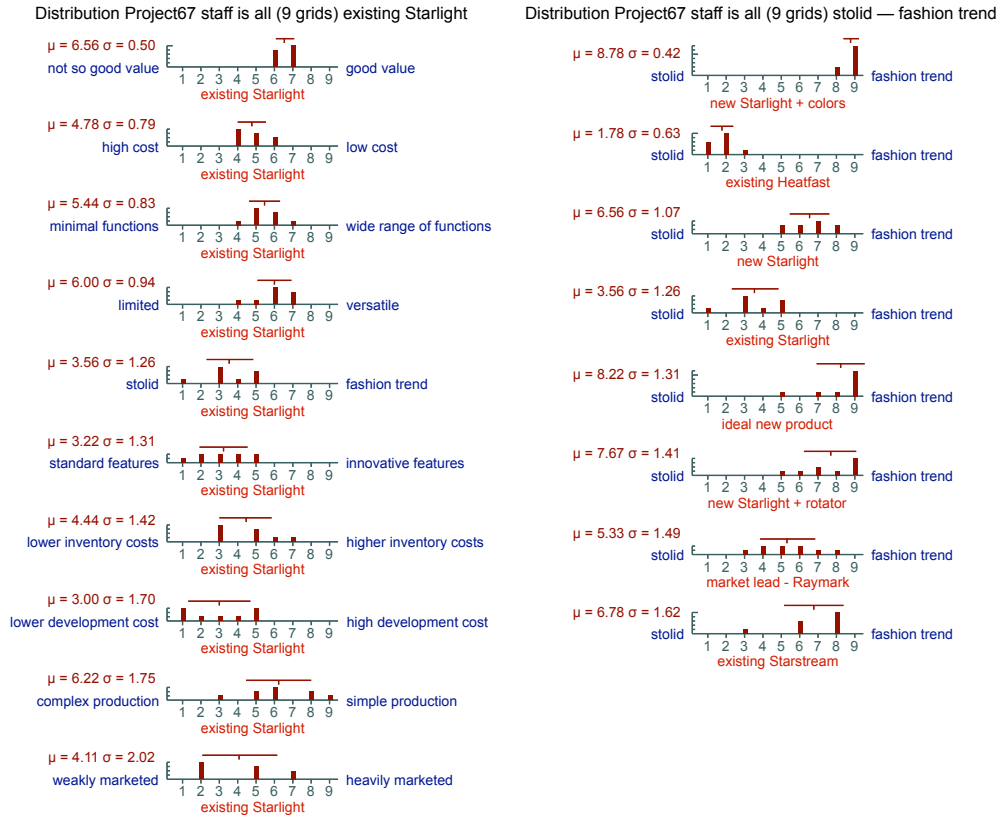


Figure 44: Distributions of Project67 staff ratings for an element and a construct

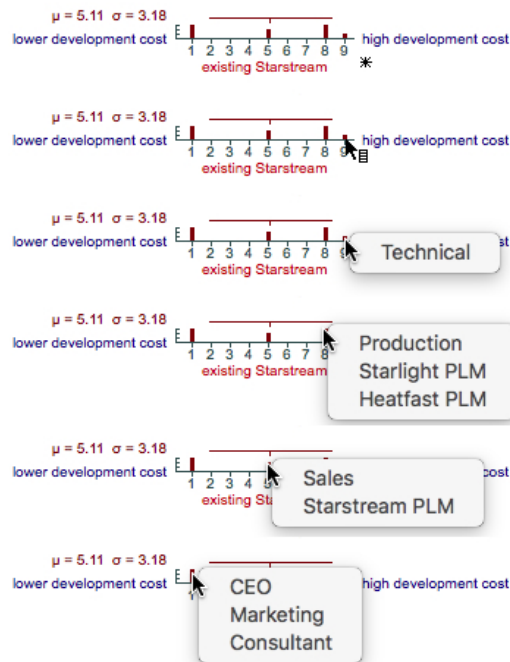


Figure 45: *Histogrids* script providing an interactive interface to identify the underlying grids

Figure 45 illustrates the interactive exploration of the histogram having the least consensus (bottom right of Figure 44). At the top, as the cursor is moved over the histograms it is shown as a star until, as shown in the histogram below, when it is over a histogram bar it changes to an arrow with a menu symbol to indicate that a click will activate a popup menu.

The effect of clicking on each bar in turn is shown in the last four histograms. The popup menu shows the name(s) of the grid(s) contributing to the bar, and selecting a name will open the grid for more detailed investigation. Usually the names in themselves are sufficient in identifying the sources of outliers and consensual subgroups, allowing those involved to determine if the pattern is one reasonably expected or an interesting anomaly meriting further investigation.

The *HistoGrids* script can be terminated by double-clicking when the menu is a star.

4.1.4 Histogram parameters panel

The *Histogram parameters* panel (Figure 46) is used to manage the graphical display of the histograms.

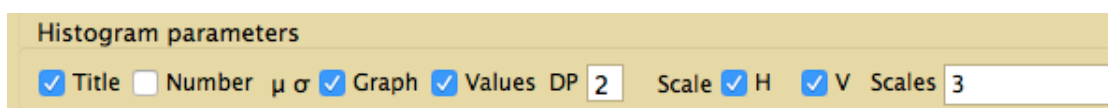


Figure 46: Histogram parameters panel

The first checkbox on the left specifies whether the plot is titled, the next whether element and construct numbers are shown, the next two whether the mean and standard deviation are graphed (location of mean and one SD either side of it) and/or shown as numbers and, if so, to how many decimal places. The next two specify whether the ratings are shown on the horizontal scale and whether a vertical scale is plotted.

The text field on the right enables the geometry of the histograms to be adjusted. It can contain up to 5 numbers separated by commas: the number of pixels for each vertical increment in the histogram bars; their widths; horizontal space between bars; vertical space between histograms, and horizontal space at the end of each plot. If fewer than 5 values are specified the remainder are filled from the appropriate position in the default values 3, 3, 8, 4, 4.

The default values are usually appropriate for studying the results but the options are useful in developing presentations to achieve clarity and simplicity for a wider audience.

4.2 Comparing distributions

If class metagrids have been used to stratify the population then it becomes possible to compare the distributions of the ratings of two sub-communities by showing them separately on a pair of histograms on the same plot. This also makes it possible to assess statistically the null hypothesis that both distributions are drawn from some common distribution, that is that the two sub-communities are not differentiated in the way they are rating a particular element on a particular construct.

If there is a statistically significant difference between the populations it could arise for many different reasons, for example, that one sub-community is more coherent than the other in the way

that it makes the distinctions underlying the constructs, or that there are major conceptual differences between the sub-communities in their interpretation of a construct or element. The reason is usually perceptible in the relation between the paired histograms, whether one is more tightly clustered, whether both are largely clustered but one has a larger number of outliers, whether one is predominantly towards one pole of the rating scale, and the other to the other, and so on. The statistical anomaly indicates a potential social or conceptual difference and the histograms suggest the nature of that difference.

An appropriate measure of whether two distributions over an ordinal scale may be reasonably supposed to samples of the same distribution is the probability that values in one distribution are predominantly greater than those in the other, or two-sidedly that they are predominantly either greater or less, a statistic first developed by Wilcoxon (1945), generalized by Mann and Whitney (1947) in their “U-test,” and given a probabilistic interpretation by Birnbaum (1956).

The statistic requires no assumptions about the underlying distributions of the populations being compared, and has become widely recognized as a preferred alternative to Student’s “t-test” when assumptions of normalcy are inappropriate. For a normal distribution its power its asymptotic power is 95% ($3/\pi$) (Mood, 1954) and for many non-normal distributions it is more powerful than the t-test (Cliff, 2014). This makes it particularly well-suited to conceptual grid rating scale data comparisons where the distributions are generally skewed, discrete, differing in variance and best-interpreted as ordinal data.

There is another statistic, d , applicable to the comparisons which estimates δ the probability that one distribution is higher less the probability that the other is higher (Agresti, 2010). Cliff (1993) recommends this as a simply interpretable measure for reporting the *effect size* (Cumming, 2012; Lakens, 2013). The value of d is simply derivable from the Mann-Whitney U value and the cardinalities of the two distributions, and for a collection of comparisons of different distributions of the same two classes p and d are monotonically related so that both will sort the distributions in the same way.

RepGrids makes both d and p available reporting them as percentages of their maximum possible values with p range from 0 to 100% and d from -100% to +100% where the sign indicates which distribution has ratings generally higher than those of the other. It is generally useful to facilitate meta-analyses to report both statistics.

Note that, as usual, great care needs to be taken in interpreting the significance levels reported. The values calculated are appropriate to confirmatory testing of specific hypotheses that have been advanced prior to data collection. If, however, the analyses are used in an exploratory mode (Tukey, 1977, 1980) to search for likely anomalies then any interpretation must take into account the number of distributions explored, usually the number of elements times the number of constructs.

If, for example, there are 10 element and 10 constructs then 100 cases will be explored. If the probability threshold is set at 5% then one would expect 5 cases to be significant by chance. If one finds 8 cases then the probability of the outcome of the exploration is that of finding 8 cases out of one hundred when one expect 5, and a simple calculation shows the probability of 8 or more to be 12.8% which is the reportable effect size (rather than 5%). 10 cases would have an effect size of 2.81% as the result of an exploratory data analysis. RepGrids reports this overall exploratory statistic also.

4.2.1 Compare two classes of distributions panel

The *Compare two classes of distributions* panel (Figure 47) is used to manage class comparisons.

Compare two classes of distributions

☒ Probability $p \leq 2.5$ % ☒ 2-sided ☒ Classes ☒ p ☒ d DP 2 **Histograms**

By ☐ existing Starlight ☒ Sort

2 perspective is market ☒ Link

6 perspective is product ☒

☐ Classes ☐ p ☐ d ☒ d p sorted ☐ Grids $p \leq 10.0$ % **Data**

☐ σ (SD) ☐ μ (Mean) ☒ $\sigma \mu$ sorted

Figure 47: Compare two classes of distributions panel

4.2.2 Comparison histograms

The two top rows provide similar options to distributions pane of generating histograms either: those satisfying a criterion, in this case that the probability, p , of the distributions being samples from a common distribution is less than that specified for a one or two-side comparison; or those for a particular element or construct, sorted by p if “Sort” is checked but otherwise in the same order as in the grid. This enables either the most significant comparisons to be listed, or those for a particular item of interest. It is often useful if the distributions appear significantly different for a particular element on a particular construct to examine all the distributions for that element and all those for that construct to see if there is general tendency or only a particular possible anomaly.

The additional controls on the first line specify whether the classes and the values of p and/or d will be shown in the plot and, if so, the number of decimal places to be used. This last option is provided because if the significance is exceptionally high then the value of p may be very low and that of d close to 100 so that more decimal places may be appropriate than those specified for the standard deviation and mean in the *Histogram parameters* panel which, otherwise, determines the remaining features of the plot.

The menus on the following two rows are used to select the classes to be compared and show a class and the number of grids falling under it. Each provides access to both the defined compound classes and the construct/category basic classes from all the metagrids that have been loaded. It is common for the second menu selection to be a class related to that in the first menu, and the *Link* button when checked causes the second menu to be set to the class following that in the first menu whenever the latter is changed. The second class can be separately selected if this is not the required comparison.

Figure 48 left shows the 7 histograms with $p \leq 2.5$ produced when the “Histograms” is clicked with the parameter settings shown. The two classes are differentiated by the colours specified in the default net, and listed on the left of each histogram together with the mean and standard deviation

of each distribution. The p and d statistics are shown on the right of each histogram. The probability of finding this many histograms satisfying the specified criterion in exploring all possible histograms is reported at the bottom.

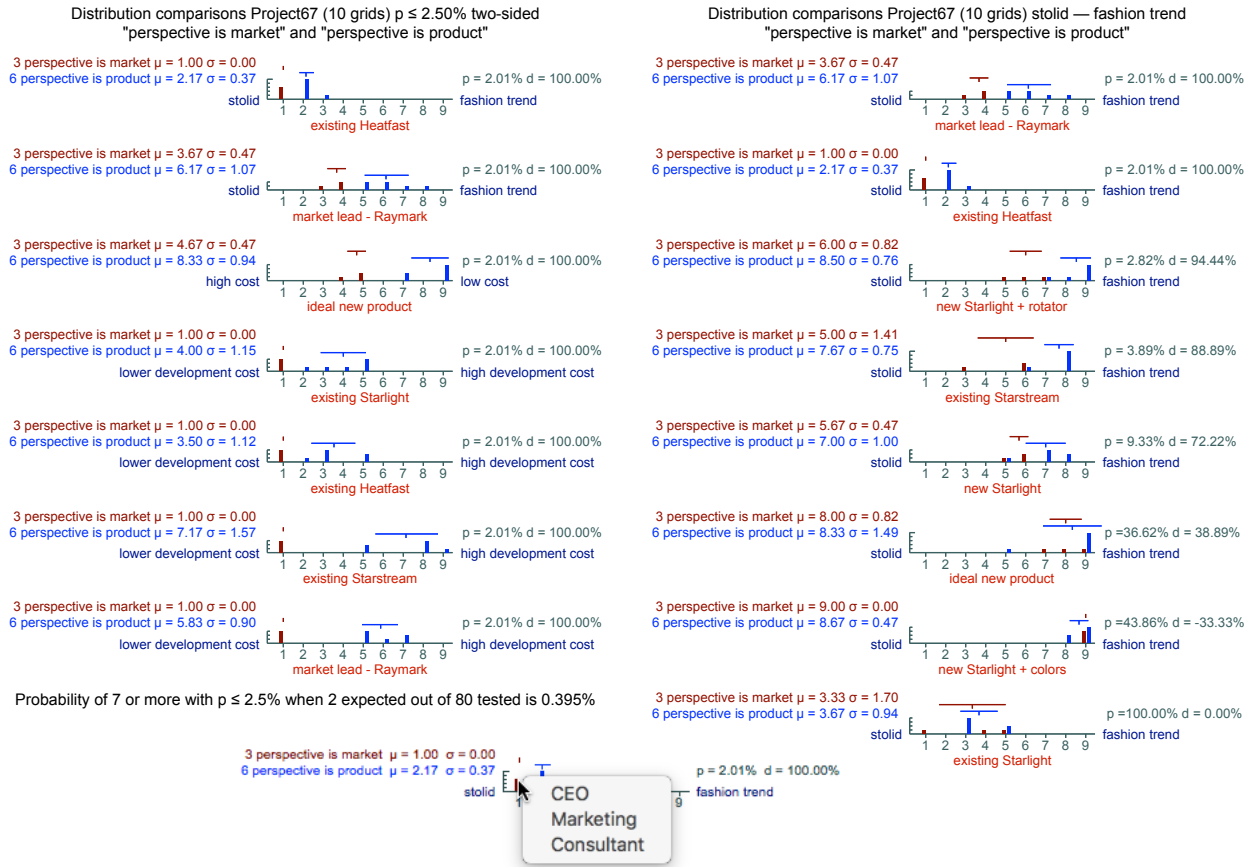


Figure 48: Distributions of Project67 staff ratings for an element and a construct

The first two involve the construct *stolid*—*fashion trend* and it may be of interest to use the option to examine all histograms for this construct as shown on the right where it can be seen that significant differences are not apparent for all elements.

It is also apparent that in the histograms on the left the common reason for a significant difference is that one class is always extremal compared to the other and all the histograms show no overlap which leads to the d statistic being 100% in all cases (it could also be -100% if the relative positions of the two distributions were reversed).

At the bottom of the Figure is shown an examination of the membership of the *perspective is market* class using the *HistoGrids* script. The new CEO and the consultant had a marketing background and both worked closely with the marketing director making it likely that their views were influenced by him and resulted in them forming a coherent group having different views from the remainder of the staff. This might indicate, for example, that the CEO needs to have greater consul-

tation with his other senior managers, and the results of the analysis might make this apparent to all without it having the matter to be raised in discussion.

This potential issue is also apparent in the socionets, and a combination of the socionets and the histograms marked up in RepNet might be used to present it in a useful fashion (Figure 49). This figure was created by copying and pasting three of the histograms into the socionets output and placing two oval regions in the background of the socionets to emphasize the groupings.

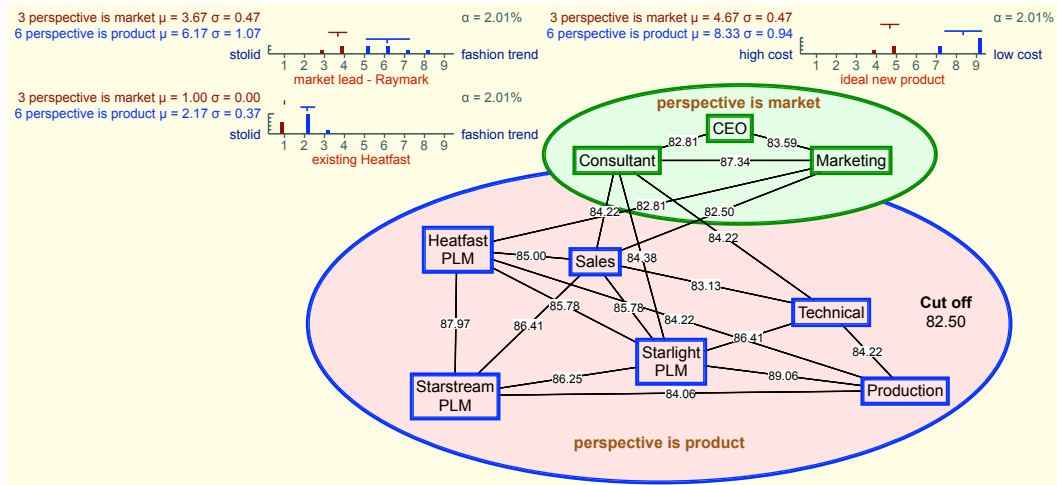
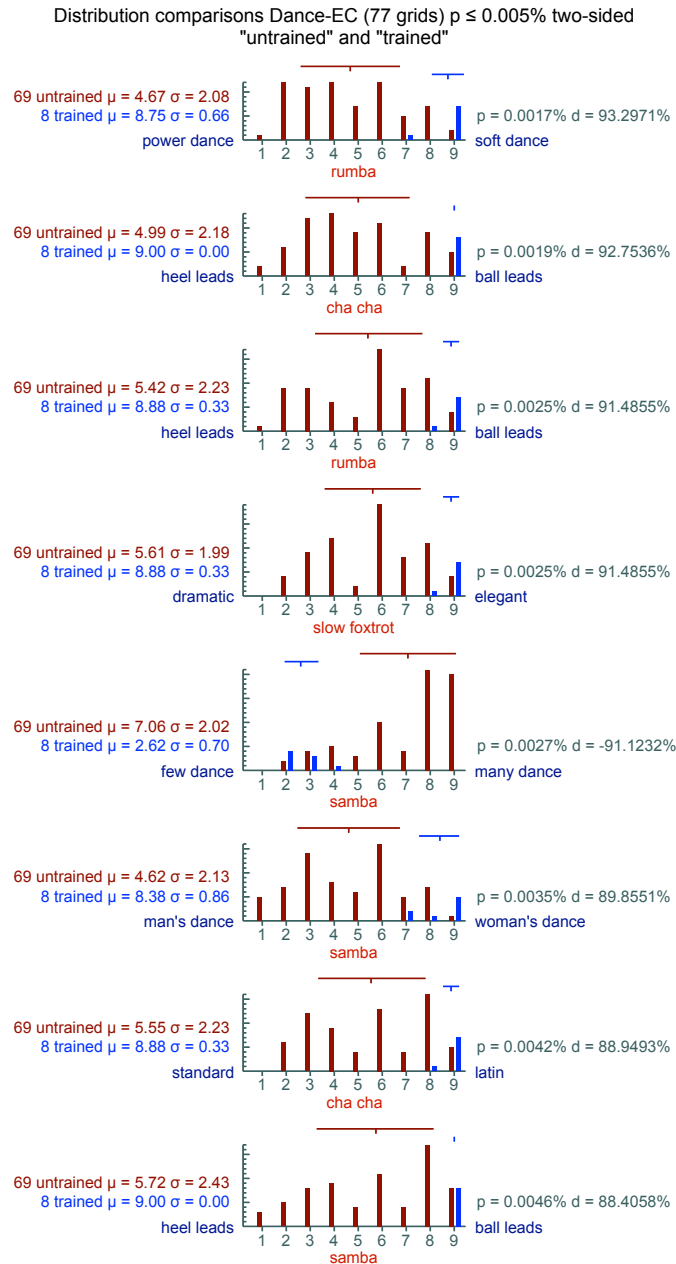


Figure 49: Combining histograms and socionets in RepNet for a presentation

The nine grids *Project67* dataset is small for a distributional analysis. Figure shows a comparison from a project with 77 participants where 73 comparisons were significant at the 2.5% level and the level has been set to 0.05% in order to create a smaller figure. The data is part of the study of the sociocognitive structure of an international ballroom dance community (Gaines and Shaw, 2012) where the stratification shown reflects the dichotomous nature of the community which splits into *social dancers* who dance for recreation and “trained dancers” who compete, take medal tests, are professionally qualified teachers.

The ballroom dance community developed a precisely defined technique and technical vocabulary in the 1920s (Richardson, 1946) in order to facilitate dancers trained by one teacher to dance with those trained by another, and technical terms are widely known but their precise meaning is understood only through intense training. Thus, the main feature of the histograms being compared is that the ratings of trained group cluster tightly whereas those of the untrained group are widely dispersed, but generally overlap those of the trained group since some social dancers have also studied technique.

The size and dichotomous nature of this stratification of the community combine to lead to results whose *statistical significance* corresponds to absolute certainty and would not surprise anyone who knows the community, but serve to illustrate the operation of the software in large-scale studies.



Probability of 8 or more with $p \leq 0.005\%$ when 0.0066 expected out of 132 tested is 7.15E-21%

Figure 50: Distributions of Project67 staff ratings for an element and a construct

Bellman (2012; 2016) conducted a study of Swedish property appraisers to explore possible differences in the conceptual systems of different subgroups stratified along a number of dimensions such as education and location (Figure 38). A stakeholder group developed an initial grid of elements and constructs significant to the purposes of the study (Figure 51, English translation of the Swedish original) and ratings were entered individually by the 67 professional appraisers participating in the study.

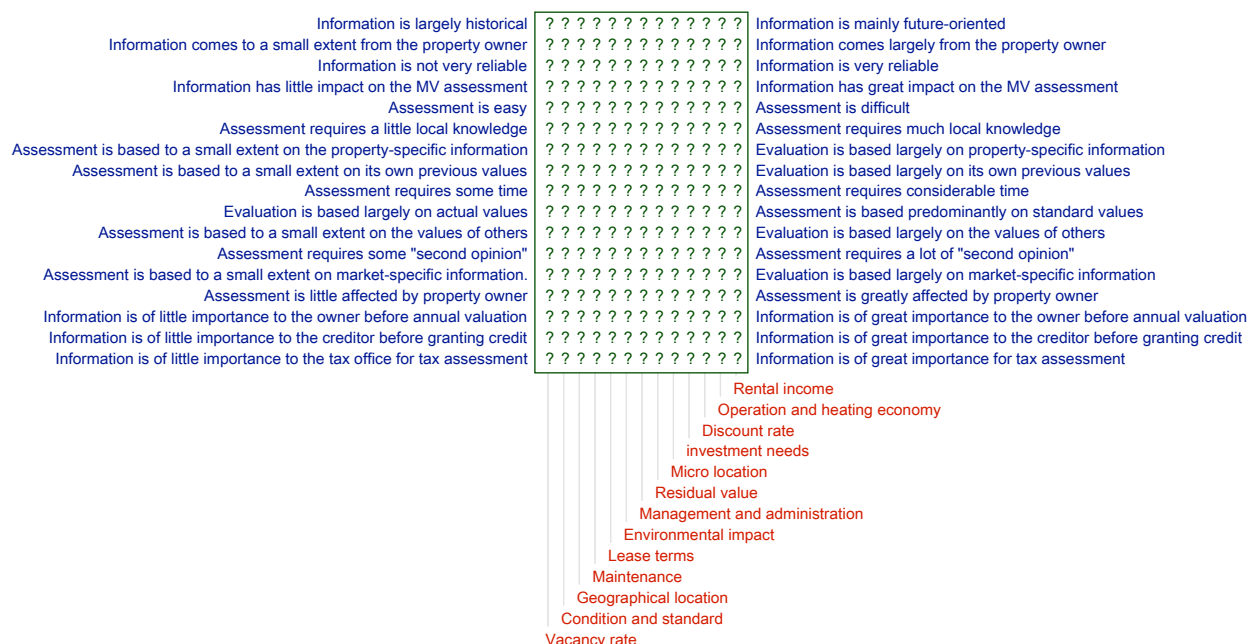


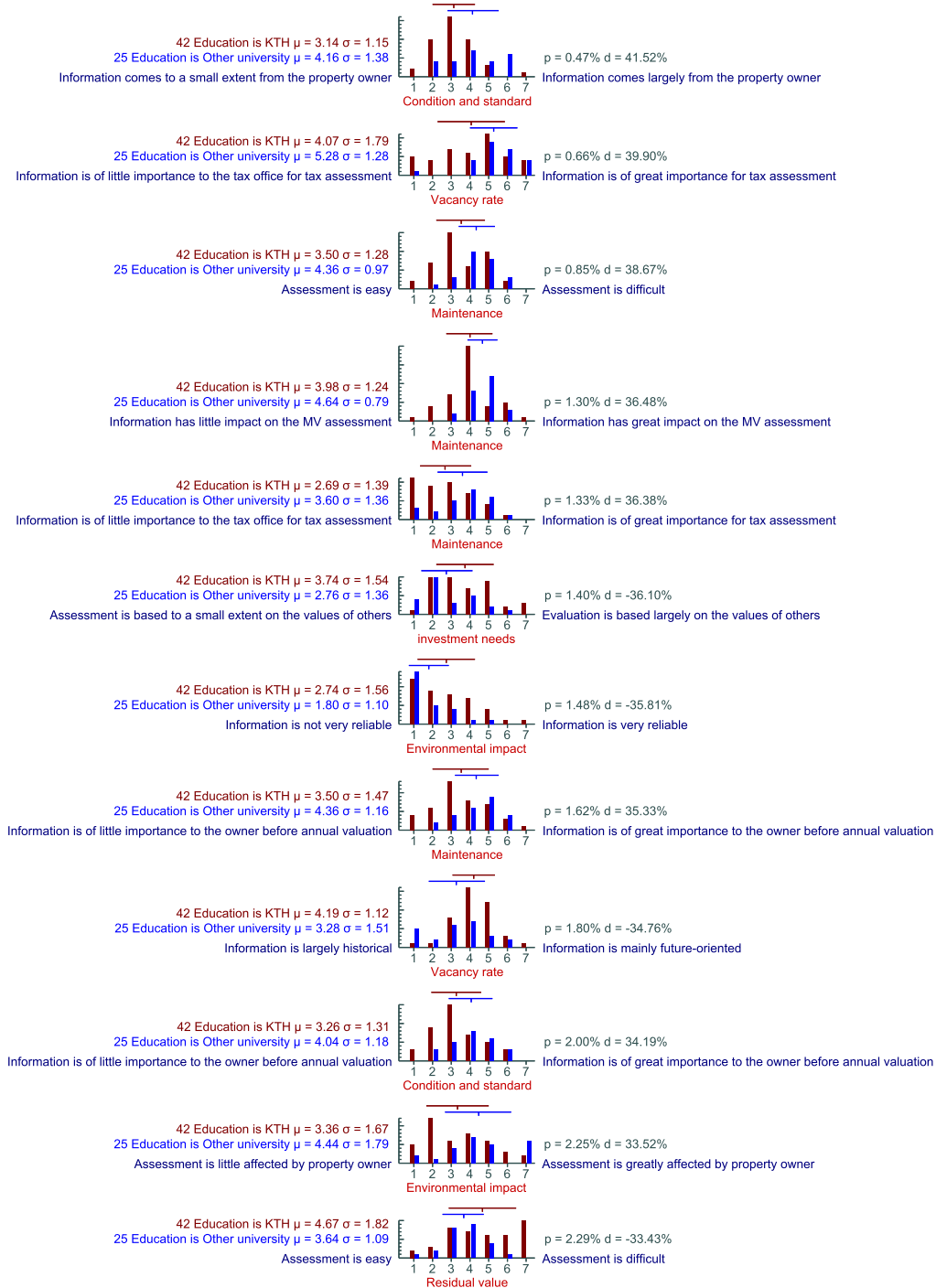
Figure 51: Significant elements and constructs for a property appraiser study (Bellman, 2012)

Figure 52 shows one of the comparisons made in Bellman (2012, p.147) and analyzed in Bellman et al. (2016, Exhibit 6) of two groups distinguished by the institutions where they received their professional education. The authors note that there has been a major attempt by the professional association to standardize the appraisal process based on a common understanding and use of well-defined and accepted criteria, and this is apparent in the histograms where there is a high degree of consensus regardless of the source of the knowledge.

The overall assessment shows that there may be more differences that can be ascribed to chance, and the degree of spread and outliers in some of the distributions suggests topics where there is lack of consensus meriting more specific investigation. This is generally the desired outcome of a successful study of this nature, to increase understanding of the sociocognitive structures involved and highlight possible anomalies for further study.¹

¹We are grateful to Lina Bellman for permission to use her published research to illustrate the techniques described in this section— the dataset itself is confidential.

Distribution comparisons Appraisers (67 grids) $p \leq 2.5\%$ two-sided
 "Education is KTH" and "Education is Other university"

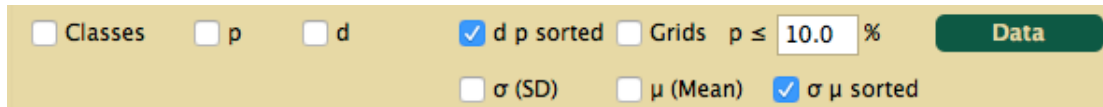


Probability of 12 or more with $p \leq 2.5\%$ when 5.525 expected out of 221 tested is 1.04%

Figure 52: Distributions of property appraisers ratings (Bellman, 2012)

4.2.3 Comparison data

The comparison of the the distributions generates a substantial amount of data that can be accessed through the *Data* section of the *Compare two classes of distributions* panel (Figure 53).



☐ Classes ☐ p ☐ d ☒ d p sorted ☐ Grids p ≤ 10.0 % **Data**

☐ σ (SD) ☐ μ (Mean) ☒ $\sigma \mu$ sorted

Figure 53: Data section of compare two classes of distributions panel

The first line manages the output of the classes and p and d data. Matrices of p and d values may be output and/or a list of construct/element pairs sorted by p together with the values of p, d and the standard deviations and means of each class. if the *Grids* checkbox is set then the grid names will also be included in the list so that the source each bar in the histograms can be identified. The significance threshold for p for the list may be set higher than that for the graphic plots to allow the investigation of the effect of different settings.

The third line specifies whether, for each class, the mean (μ) and standard deviation (σ) will be output, and/or shown as a list sorted by ascending standard deviation (the combined data for the two classes may be output using the upper *Data* button).

Figure 54 show the class lists, p and d matrices and sorted data for the Project67 dataset. Figure 55 show the sorted data with grid names included. This provides an alternative to use of the *HistoGrids* script to determine which grids contribute to which bars in the histograms.

The standard deviation and mean data is similar in presentation to that for the combined classes in Figure 42 except that it is segregated by class.

Group 1: perspective is market (2)

CEO

Marketing

Group 2: perspective is product (6)

Sales

Technical

Production

Starstream PLM

Starlight PLM

Heatfast PLM

Probability % of differences between distributions of ratings for perspective is market and perspective is product (two-sided%)

	1	2	3	4	5	6	7	8	
1	50.50	4.55	6.68	4.55	13.36	50.50	6.68	61.71	stolid – fashion trend
2	18.24	18.24	86.76	24.33	73.89	100.00	100.00	100.00	limited – versatile
3	9.56	73.89	18.24	40.47	4.55	4.55	4.55	4.55	high cost – low cost
4	100.00	73.89	31.73	50.50	61.71	40.47	18.24	100.00	not so good value – good value
5	73.89	4.55	61.71	31.73	18.24	18.24	18.24	50.50	weakly marketed – heavily marketed
6	100.00	86.76	73.89	86.76	73.89	73.89	73.89	61.71	minimal functions – wide range of functions
7	86.76	73.89	61.71	9.56	86.76	9.56	73.89	100.00	lower inventory costs – higher inventory costs
8	40.47	73.89	18.24	40.47	31.73	73.89	31.73	61.71	standard features – innovative features
9	4.55	4.55	4.55	4.55	73.89	40.47	50.50	50.50	lower development cost – high development cost
10	18.24	31.73	31.73	86.76	18.24	18.24	24.33	100.00	complex production – simple production
									ideal new product
									new Starlight + rotator
									new Starlight + colors
									new Starlight
									market lead - Raymark
									existing Starstream
									existing Heatfast
									existing Starlight

Size effect % of differences between distributions of ratings for perspective is market and perspective is product (two-sided%)

	1	2	3	4	5	6	7	8	
1	33.33	100.00	91.67	100.00	75.00	-33.33	91.67	25.00	stolid – fashion trend
2	66.67	66.67	8.33	58.33	-16.67	0.00	0.00	0.00	limited – versatile
3	83.33	-16.67	66.67	-41.67	100.00	100.00	100.00	100.00	high cost – low cost
4	0.00	16.67	50.00	-33.33	25.00	41.67	66.67	0.00	not so good value – good value
5	-16.67	100.00	25.00	-50.00	-66.67	-66.67	-66.67	-33.33	weakly marketed – heavily marketed
6	0.00	-8.33	-16.67	-8.33	16.67	16.67	16.67	25.00	minimal functions – wide range of functions
7	8.33	16.67	25.00	83.33	-8.33	-83.33	-16.67	0.00	lower inventory costs – higher inventory costs
8	-41.67	16.67	-66.67	41.67	50.00	-16.67	50.00	25.00	standard features – innovative features
9	100.00	100.00	100.00	100.00	-16.67	-41.67	-33.33	-33.33	lower development cost – high development cost
10	66.67	50.00	50.00	-8.33	66.67	66.67	58.33	0.00	complex production – simple production
									ideal new product
									new Starlight + rotator
									new Starlight + colors
									new Starlight
									market lead - Raymark
									existing Starstream
									existing Heatfast
									existing Starlight

Sorted significance of differences between distributions of ratings for perspective is market and perspective is product (two-sided ≤10%)

	p	d	o1	μ1	o2	μ2	Construct: Element
1:	4.55	100.00	0.00	1.00	2.45	4.00	weakly marketed – heavily marketed: existing Heatfast
2:	4.55	100.00	0.00	1.00	0.37	2.17	stolid – fashion trend: existing Heatfast
3:	4.55	100.00	0.00	1.00	1.12	3.50	lower development cost – high development cost: existing Heatfast
4:	4.55	100.00	0.50	3.50	1.07	6.17	stolid – fashion trend: market lead - Raymark
5:	4.55	100.00	0.00	1.00	0.90	5.83	lower development cost – high development cost: market lead - Raymark
6:	4.55	100.00	0.50	4.50	0.94	8.33	high cost – low cost: ideal new product
7:	4.55	100.00	0.00	2.00	0.90	5.83	high cost – low cost: new Starlight + rotator
8:	4.55	100.00	0.50	2.50	1.07	6.17	high cost – low cost: new Starlight + colors
9:	4.55	100.00	0.50	3.50	0.76	6.50	high cost – low cost: new Starlight
10:	4.55	100.00	0.00	1.00	1.15	4.00	lower development cost – high development cost: existing Starlight
11:	4.55	100.00	0.00	1.00	1.57	7.17	lower development cost – high development cost: existing Starstream
12:	6.68	91.67	1.00	6.00	0.76	8.50	stolid – fashion trend: new Starlight + rotator
13:	6.68	91.67	1.50	4.50	0.75	7.67	stolid – fashion trend: existing Starstream
14:	9.56	83.33	0.00	4.00	0.69	5.17	high cost – low cost: existing Starlight
15:	9.56	-83.33	0.00	9.00	1.25	6.67	lower inventory costs – higher inventory costs: new Starlight + colors
16:	9.56	83.33	1.00	4.00	0.69	5.83	lower inventory costs – higher inventory costs: market lead - Raymark

Figure 54: Data from comparisons

Sorted significance of differences between distributions of ratings for perspective is market and perspective is product (two-sided $\leq 10\%$)

	p	d	o1	$\mu 1$	$\sigma 2$	$\mu 2$	Construct: Element
1:	4.55	100.00	0.00	1.00	2.45	4.00	weakly marketed – heavily marketed: existing Heatfast
1 G1:							CEO, Marketing
G2:							
2 G1:							
G2:							Sales, Starstream PLM
3 G1:							
G2:							Technical, Starlight PLM
4 G1:							
G2:							
5 G1:							
G2:							Heatfast PLM
6 G1:							
G2:							
7 G1:							
G2:							
8 G1:							
G2:							
9 G1:							
G2:							Production
2:	4.55	100.00	0.00	1.00	0.37	2.17	stolid – fashion trend: existing Heatfast
1 G1:							CEO, Marketing
G2:							
2 G1:							
G2:							Sales, Technical, Production, Starstream PLM, Starlight PLM
3 G1:							
G2:							Heatfast PLM
4 G1:							
G2:							
5 G1:							
G2:							
6 G1:							
G2:							
7 G1:							
G2:							
8 G1:							
G2:							
9 G1:							
G2:							
3:	4.55	100.00	0.00	1.00	1.12	3.50	lower development cost – high development cost: existing Heatfast
1 G1:							CEO, Marketing
G2:							
2 G1:							
G2:							Production
3 G1:							
G2:							Technical, Starlight PLM, Heatfast PLM
4 G1:							
G2:							
5 G1:							
G2:							Sales, Starstream PLM
6 G1:							
G2:							
7 G1:							
G2:							
8 G1:							
G2:							
9 G1:							
G2:							
4:	4.55	100.00	0.50	3.50	1.07	6.17	stolid – fashion trend: market lead - Raymark
1 G1:							
G2:							
2 G1:							
G2:							
3 G1:							CEO
G2:							
4 G1:							Marketing
G2:							
5 G1:							
G2:							Sales, Technical
6 G1:							
G2:							Starlight PLM, Heatfast PLM
7 G1:							
G2:							Starstream PLM
8 G1:							
G2:							Production
9 G1:							
G2:							

Figure 55: Sorted data from comparisons with grid names included

5 Compare Pane

Clicking on the *Compare* tab in the RepGrids window brings up the Compare pane (Figure 56). It comprises two sub panels, one supporting the comparison of any pair of grids from the selected grids, and the other the management of the default styles for the grids and nets created by various analyses.

5.1 Compare two grids panel

The *Compare two grids* panel is identical in functionality to that of the RepGrid *Compare* dialogue. The only variation is that instead of opening a second grid to compare with that open in RepGrid one uses the two popup menus at the top to compare any pair of grid that have been selected in the *Grids* pane. The chosen grids and the number of items they have in common are shown in the top line.

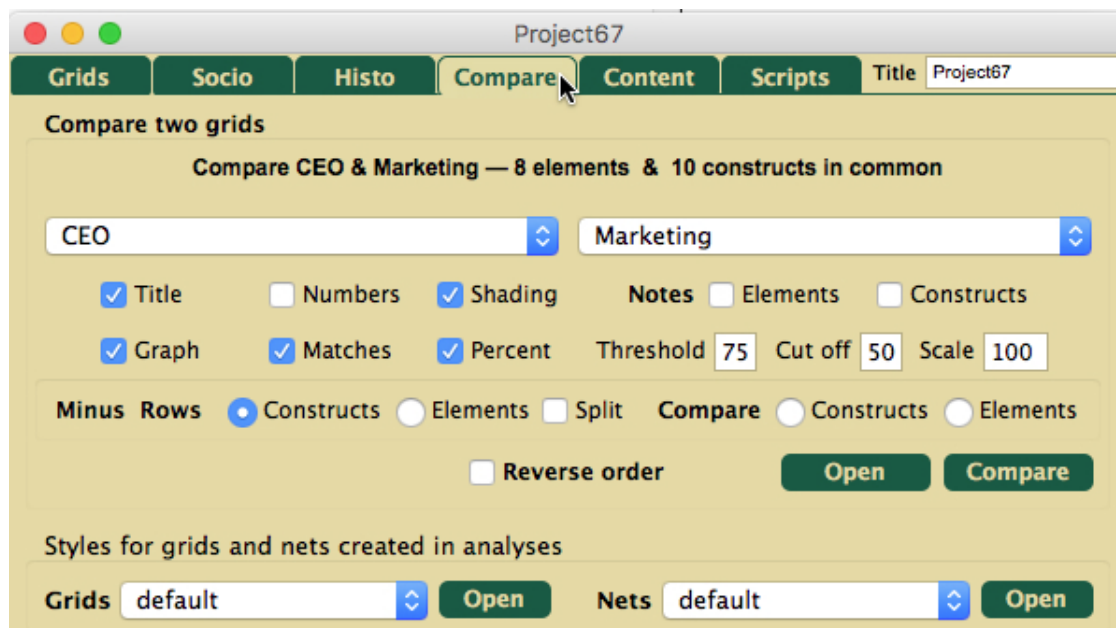


Figure 56: RepGrids window showing the **Compare** pane

This ease of comparison of any pair of grids in the database makes RepGrids a useful tool to manage the comparison of any collection of grids whether or not they are otherwise being held in a RepGrids file. One can create a new RepGrids window in the Rep Plus Manager, drag the collection of grids to it, and use the *Compare* tool to investigate whatever comparisons are of interest.

The first row of check boxes determine whether: the plot is titled; elements and constructs are numbered; high and low differences are shaded; notes attached to elements and constructs are shown.

The second row determines whether matches will be graphed; match values will be shown; the cumulative percentage of matches above or equal to the match value will be shown; and specifies the threshold, cut off, and scale. The *Threshold* value determines where in the plot matches will be

shown as below threshold. The *Cut off* value specifies the lowest match that will be plotted. The *Scale* value determines how much space will be allocated to the graph of matches.

The panel below is only visible if the grids being compared have both common elements and common constructs. The four radio buttons select whether a *Minus* plot of the difference grid is required and, if so, whether the rows should be constructs or elements, or whether a *Compare* plot is required where the best matching constructs or elements are shown. The *Split* check box determines whether a *Minus* plot shows the difference in ratings or the actual ratings on separate lines (*Compare* plots always use separate lines).

In the bottom row, the normal mode of comparison is that the primary grid is compared with the secondary grid, and the *Reverse order* check box reverses this. The *Power* value of the Minkowski metric used to compute matching scores is that specified at the bottom of the *Grids* pane. The *Open* button opens a copy of first grid of the comparison in RepGrid with the *Compare* dialogue shown, the second grid opened as specified, and the parameters set to those specified. This enables other features of RepGrid to be used such as element and construct selection if required. Otherwise, the *Compare* button activates the specified comparison.

5.2 Styles for grids and nets created in analysis panel

The *Defaults* panel at the bottom of the *Compare* manages the selection of default grid and default net that are used in the various RepGrids analyses (Figure 57). It also enables those available to be opened and edited.

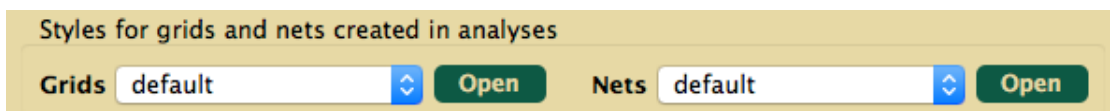


Figure 57: Styles for grids and nets created in analysis panel

The styles to be used in grids and nets generated in various RepGrids analyses are stored in grids and nets located in a *GridsScripts* directory in the network of such directories accessible to the current RepGrids file (§8.2). The *Grids* and *Nets* menus show the current default grid and net, and allow alternative ones to be selected (Figure 58).

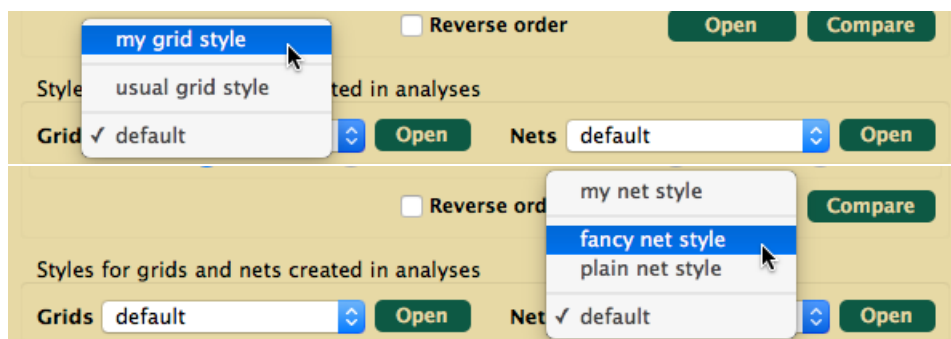


Figure 58: Selecting styles for grids and nets created in analysis

The lowest item in the menu are the default grid and net issued with the Rep Plus software and stored in a GridsScripts directory alongside the application. There is another GridsScripts directory in the Rep Plus directory in the Documents or MyDocuments directory that is intended for generally available style grids and nets developed by users, and these show in the central partition of the menu. There may also be more data specific or private GridsScripts directories in association with the RepGrids database files and these show at the top of the menus.

The *Open* buttons allow the selected style grids and nets to be opened for editing. The selections themselves are stored in the RepGrids file and reselected when that file is opened so that once a user has made a selection it is preserved as the ongoing default for that RepGrids database.

Note that a file at a higher level with the same name as one in a lower level *shadows* the one at a lower level so that it no longer appears in the menu. Thus, files named *default* in an upper level directory will effectively replace the supplied default files without those files themselves having to be deleted.

6 Content Pane

Content analysis of textual material based on the lexicographical similarity, frequency and relationship of the words used, and the *coding* of those words by researchers in terms of their *meanings* was originally developed in literary scholarship and then extended to a wide range of studies of textual materials in many disciplines (Barcus, 1960). It has become a standard method of *qualitative research* (Neuendorf, 2002; Krippendorff, 2013), particularly with the increasing availability of computer tools to support the analyses (West, 2001b,a).

Content analysis is applicable to grid data because the element and construct descriptors entered by the eliciters are well-localized discrete textual phrases that be analyzed lexicographically and coded by researchers not only through their likely linguistic meanings but also through the way they are used in the grids and the relations between them induced by the rating data. Honey (1979) pioneered the content analysis of constructs in multiple grid datasets, and Jankowicz (2004, Ch.7) has extended, applied and explained a range of content analysis methods including Honey's indexing techniques. RepGrids also supports the content analysis of element descriptions but this is unusual and the emphasis and examples in this section will focus on the content analysis of construct descriptions.

6.1 The interplay between content analysis and other grid analysis methods

There are a significant difference between the content analysis of textual material in general and that of the phrases describing items in conceptual grids. The short grid phrases do not provide as much contextual information as available from an embedding in a coherent textual narrative. However, grids do provide rather more explicit evidence of their usage through the ratings showing how they are used to distinguish a set of elements. As Wittgenstein (1953) argued the *meaning* of a word is evidenced and communicated by its use in *language games* rather than by their lexical structure or dictionary definition, and conceptual grid elicitation enables such usage to be made explicit in the context of the relevant domain being investigated.

The analysis of multiple grid analysis methods in terms of construct usage and the terminology applied (Shaw and Gaines, 1989) generates a crossplot in terms of two constructs: *same distinction—different distinction* and *same terminology—different terminology* (Figure 59). Content analysis methods focus on the two left quadrants where different people are making the same distinction, and in particular, on the human resolution of the issues in the lower left quadrant where people are applying differing terms to what may be the same distinction—*correspondence* of usage but not of terminology.

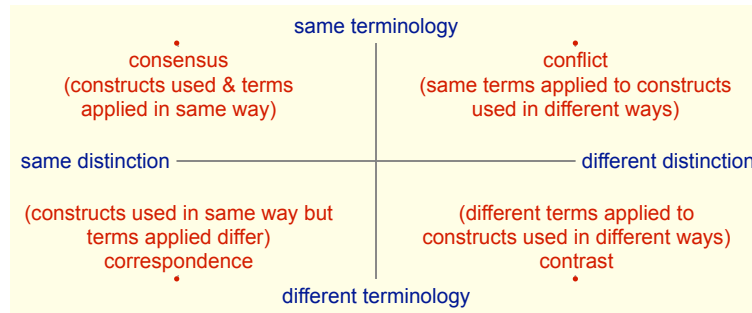


Figure 59: Consensus, conflict, correspondence & contrast in construct usage and terms applied

A problem arises because it is also possible that there is *conflict* of usage in that the same terms are being applied to different distinctions (top left quadrant), and the availability of the rating data may be used to make this apparent. If two grids have two lexically identical, or lexically very similar, constructs, but use them differently in categorizing the elements it suggests that the lexical information is misleading. In addition, if two constructs are used similarly but the lexical descriptions seem to indicate that different distinctions are being made it suggests that the similarity indicates a *correlation* in the domain that the set of elements is inadequate to distinguish rather a *correspondence*.

Thus, content analysis of grids can make use of two distinct sources of information, construct matches indicating the same or similar usage, and lexical similarity/interpretation indicating the same or similar communicative intention. The RepGrid *Contents* pane makes both sources of information available in a hierarchical tabular format that supports grouping of items based on lexical information taking into account matches based on usage.

The matches between items in the grid are dynamically computed and always available, and are categorized as low, medium or high according to Honey's criteria to support his method if a supplied construct has been used. If the grid dataset has common elements (or constructs) then the construct (or element) match data from the SOCIO analyses is made available.

If one or more class metagrids are part of the dataset then they may be used to track how many items in each group are from a particular class.

6.2 Populating the content pane

The use of the content analysis tools will be illustrated using a dataset from an early study of the potential application of grid elicitation and analysis to the development of knowledge-based systems (?). Carlson's (1983) *BIAIT* method of business process analysis uses seven constructs to characterize the nature of a business to facilitate the development of appropriate data collection and accounting practices. Such analyses continue to be used in developing computerized accounting systems today (Leshob et al., 2015), and the 1982 study investigated whether the *BIAIT* model would naturally emerge from the construing of those with some degree of experience of business and accounting practices.

A pilot study showed that six of the *BIAIT* constructs appeared in a grid elicited from an experienced businessman construing a range of different types of business. It raised the question of

whether elicitation should focus on individual experts or would be more robust if a consensus was developed from a group with varying experience—an early interest in what has become to be known as *crowdsourcing* (Howe, 2008). Grids were elicited from sixteen individuals having a wide range of levels of business and accountancy expertise construing an initial set of ten elements.

The only analysis tools available in the early 1980s were those based on Shaw's (1979; 1980) SOCIOGRIDS that offered early versions of the Sociogrids and Mode grid analyses provided in the "Socio" pane. It was possible to generate a mode grid automatically from the construct matches and compare this with a *BIAIT* grid that used the seven *BIAIT* constructs to construe the elements, and determine that each of the *BIAIT* constructs was well-matched in the mode grid. However, the same construct was used to match more than one *BIAIT* construct suggesting that the element set might be too small, allowing naturally associated business processes to lead to correlated constructs. When the textual labels of the poles of the 186 constructs elicited were examined all of the *BIAIT* constructs were found showing that crowdsourcing was promising but the analysis at that time was informal.

The content tool in RepGrids support the humanly-managed process of identifying lexical relations, providing the match data used in the Socio analyses to help ensure that the lexical similarities also correspond to conceptual similarities.

Figure 60 show the grids pane for the *BIAIT* study dataset which also has a class metagrid representing the classification of the business expertise of those participating.

BIAIT CA

Grids Socio Histo Compare Content Scripts Title BIAIT Study

#	File	Name	Note	E	C	Date	Time	Place
1	AH	AH	Eco Major	11	10	2-Nov-1983	21:07	local
2	EM	EM	Visual Arts Major	10	10	9-Nov-1983	11:19	local
3	FALC	FALC	BBA Major	11	10	1-Nov-1983	14:10	local
4	FEP	FEP	Businessman	10	13	8-Nov-1983	10:48	local
5	FJCS	FJCS	Eco/Bus Major	10	10	3-Nov-1983	15:05	local
6	GCB	GCB	Eco Major	11	10	8-Nov-1983	11:57	local
7	HJ	HJ	Eco/Bus Major	10	11	3-Nov-1983	11:07	local
8	LCL	LCL	Eco/Bus Major	10	10	5-Nov-1983	12:03	local
9	LYCL	LYCL	CPSC Major	10	11	6-Nov-1983	10:03	local
10	MEMK	MEMK	Bus Major	10	11	4-Nov-1983	14:06	local
11	MSC	MSC	Major Eco/Bus	11	11	6-Nov-1983	14:01	local
12	PEK	PEK	Businessman	11	9	31-Oct-1983	11:35	local
13	RKK	RKK	Math Major	12	10	4-Nov-1983	17:21	local
14	SFG	SFG	Accounting Major	10	10	2-Nov-1983	17:09	local
15	SMH	SMH	Eco Major	11	10	2-Nov-1983	14:19	local
16	THL	THL	Chemistry Major	13	10	5-Nov-1983	10:10	local

Count same: 16 Grids 10 ☒ Elements 0 ☐ Constructs ☐ Both Power: 1.0

Identifier: ☐ ID ☒ Name ☐ Note ☐ Date ☐ Time Item: ☐ Note ☐ #

Class metagrids and selection

16/16 16/16 Class grid for BIAIT Study

6 business expertise is low

Figure 60: BIAIT study dataset

The initial content pane is empty except for a heading showing the number of grids, total number of constructs and common elements and a *Populate* button.

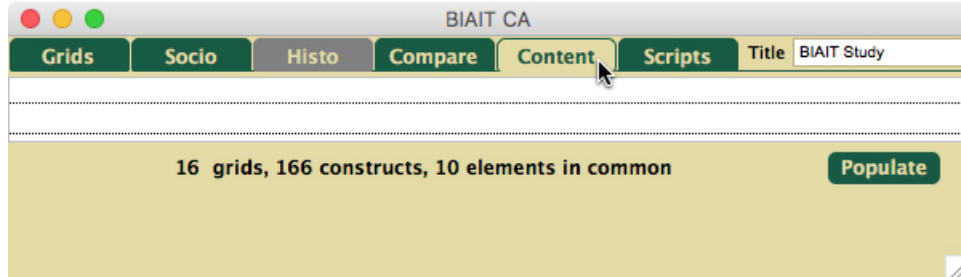


Figure 61: Initial Content Pane (truncated)

When the *Populate* button is clicked a table is generated with columns showing the grid ID and the name, poles and note of each construct (Figure 62).

ID	LHP	RHP	Name	Note
AH	More profits	Less profits		
AH	Higher volume	Lower volume		
AH	More work	Less work		
AH	Rental	Sales		
AH	No continuous capital required	Continuous capital outlay		
AH	Public benefit	Private benefit		
AH	Less necessary	Necessary		
AH	Demand	Low demand		
AH	Supply	Low supply		
AH	Low price	High price		
EM	Sold publically	Sold privately		
EM	Intangibles	Tangible		
EM	Being hired	Being sold		
EM	Service	Sales		
EM	Cheaper	More expensive		
EM	Customer goes to salesman	Agent looks for customers		
EM	Salesman not required	Salesman required		
EM	Training not required	Training required		
EM	Decorative	Useful		
EM	One time sale	Ongoing service		
FALC	Full time	Part time		
FALC	Renting	Sales		
FALC	Complex business	Simple business		

Figure 62: Content pane populated

The construct “name” field is included because it is some used as part of the identification of the poles, and the construct “note” field because it is sometimes use to explain a construct’s meaning in greater detail which may be relevant to a construct analysis.

The table constitutes the content analysis database and, while it originates in the grids selected in the *Grids* pane, it is recorded separately in the RepGrids file so that once it is created other analyses may be performed without interfering with the content analysis. This enables the analysis to be developed over time, saved in the file, and reopened in the same state for continuation as required.

A content analysis can involve much effort and is difficult to reconstruct if it is lost so the system is as protective as possible and it is good practice to make frequent dated backups of the RepGrids file. If, for example, some grids used in the content analysis were deleted from the *Grids* pane, when the file is saved the user will be warned about the deletion and asked whether those grids should be reinstated to support the content analysis.

The “Discard” button clears the content analysis—as it is not undoable the user has to confirm this is the intent. The *Select* button selects in the *Grids* pane those grids used in the content analysis so that they may be analyzed in other tools.

The *Reverse* button appears if the items being analyzed are constructs. It reverses those constructs that are selected in the *Content* pane to enable them to be aligned with similar constructs before they are grouped.

6.2.1 Controls for hiding columns and reversing constructs, and popup menu

The check boxes at the bottom allow the Honey/Jankowicz *Index* (§6.5.1), *Name*, *Note* and group count # % (§6.3) columns to be opened or closed as required. Other columns and associated check-boxes appear for *Match* and *Mode* if a Socio analysis is run (§6.5.2).

Column widths may be changed by dragging the bars in the headers. The state of the checkboxes and the width of the columns are saved in the RepGrids file.

The “Discard” button clears the content analysis—as it is not undoable the user has to confirm this is the intent.

The *Select* button selects in the *Grids* pane those grids used in the content analysis so that they may be analyzed in other tools.

The *Reverse* button reverses the constructs that are selected in the *Content* pane to enable them to be aligned with similar constructs before they are grouped.

6.2.2 Popup menu commands

Right-clicking or CTL-clicking in any row displays a popup menu (without changing which rows are selected) that provides a range of options that vary according to the state of the analysis. Figure 63 shows the options available when the *Content* pane is initially populated with constructs.

ID	LHP	RHP	Name	Note
AH	More profits	Less profits		
AH	Higher volume	Lower volume		
AH	More work	Less work		
AH	Rental	Sales		
AH	No continuous capital required	Continuous capital out		
AH	Public benefit	Private benefit		
AH	Less necessary	Necessary		
AH	Demand	Low demand		
AH	Supply	Low supply		
AH	Low price	High price		
EM	Sold publically	Sold privately		
EM	Intangibles	Tangible		

Figure 63: Popup menu from right-clicking or CTL-clicking in a row

The *Copy cell* command copies the text in the cell that was clicked into the clipboard. It is useful in setting up *search phrases* for textual searches (§6.6) the highlight items with similar terms to those forming a group. It is more specific that the *Copy* command in the *File* menu that copies selected rows into the clipboard in the text export format (as if *Selected* and *Visible* were specified).

The *Reverse this row* command reverses the construct in the row that has been clicked. It is useful when one wishes to reverse a construct without disturbing already selected constructs, The *Reverse selected rows* command provides the same function as the The *Reverse* button.

6.3 Grouping similar items

The content analysis process consists of selecting items that are construed as being similar (by whatever criteria are being applied) and grouping them into a single item that is representative of all those grouped. Figure 64 shows three constructs selected (by clicking on each with the command key held down to achieve discontinuous selection) that all make a similar conceptual distinction between a product being rented or being sold outright, even if the terminology is different. The # % column has been opened to show information about the number of items in groups.

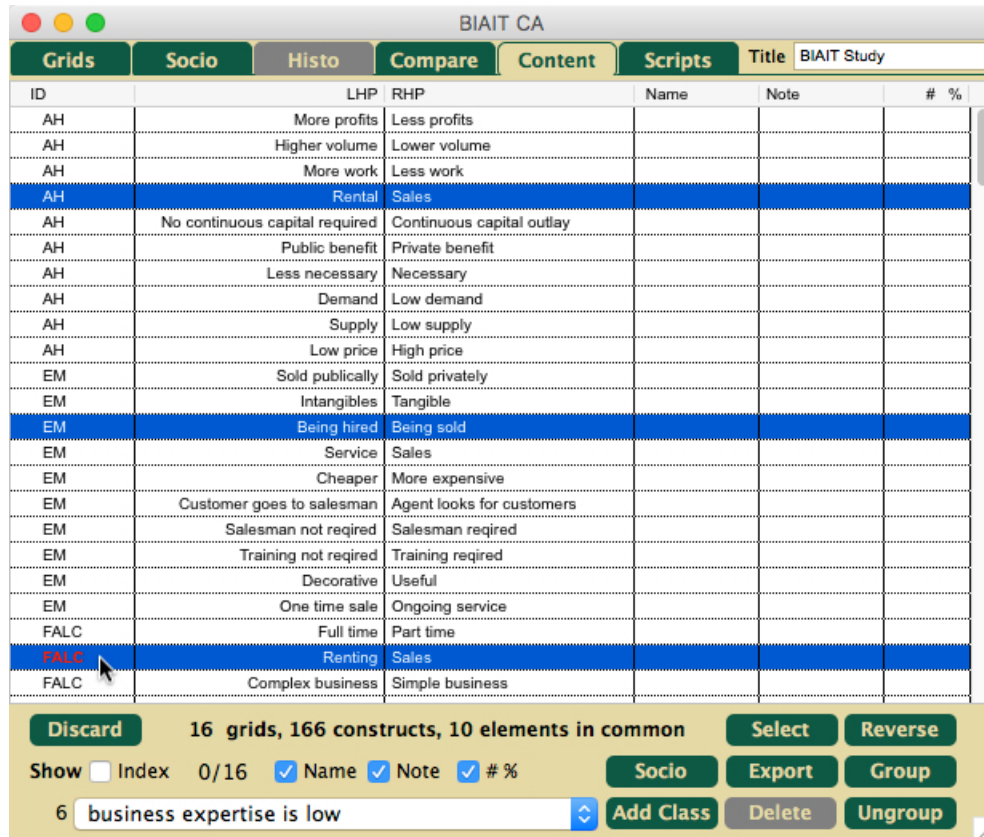


Figure 64: Selecting items for grouping

Clicking on the **Group** button creates a new group that includes those selected (Figure 65). The number of items, and the percentage of all the items, in the group are shown in the # % column on the right.

ID	LHP	RHP	Name	Note	# %
AH	More profits	Less profits			
AH	Higher volume	Lower volume			
AH	More work	Less work			
Group	Rental	Sales			3 1.3
AH	No continuous capital required	Continuous capital outlay			

Figure 65: New group created

The group is identified as such in the **ID** column but otherwise appears as an item of the type grouped, in this case a construct. It also behaves as such and may be sorted or dragged to another location just like any other item. It may also be made a member of another group allowing the creation of hierarchical structures of any depth. Thus the items in the table, and in groups, can include groups as well as constructs (or elements).

The initial terms in the name and pole fields are copied from the first construct grouped to minimize unnecessary typing, but the fields together with the name and note field may be freely edited. The first construct group can itself be specified by dragging it to be at the top of the group before

grouping to take advantage of this feature. If the annotation is very long the note field may be opened in a text editing dialogue by double clicking in the ID cell.

The *ID* field has an arrowhead symbol pointing horizontally to indicate that the group is closed and its members are no longer visible. Clicking in the arrow rotates the arrowhead to point down and opens the group to show its members (Figure 66).

ID	LHP	RHP	Name	Note	#	%
AH	More profits	Less profits				
AH	Higher volume	Lower volume				
AH	More work	Less work				
▼ Group	Rental	Sales			3	1.8
AH	Rental	Sales				
EM	Being hired	Being sold				
FALC	Renting	Sales				
AH	No continuous capital required	Continuous capital outlay				

Figure 66: Group opened to access items grouped

Grouping can be undone and if a discontinuous selection has been grouped then *Undo* will restore the items in the group to their original locations and delete the group itself. The *Ungroup* button simply flattens a group leaving the items grouped located where the group had been and deleting the group itself. Ungrouping can also be undone, effectively restoring the original group.

Note that when a group is selected all its members are automatically also selected. However, items within a group may be selected individually or together with other items just as if they were not within the group (Figure 67). Items within a group may be dragged out of it or to a different location in the group, and items may be dragged into it, making it possible to edit and extend groups once created.

ID	LHP	RHP	Name	Note	#	%
AH	More profits	Less profits				
AH	Higher volume	Lower volume				
AH	More work	Less work				
▼ Group	Rental	Sales			3	1.8
AH	Rental	Sales				
EM	Being hired	Being sold				
FALC	Renting	Sales				
AH	No continuous capital required	Continuous capital outlay				

Figure 67: Item(s) selectable within a group

6.3.1 Grouping through the popup menu

Grouping selected items through the *Group* button is an effective way to start a group and, if all the groups are dragged to one location such as the top of the table, dragging items between groups is an effective way of refining groupings. However, when the content analysis involves a large collection of items and one wishes to move an item remote from the groups to a particular group, it may be tedious to have to drag it through the table and it would be better to have a local command that can move it directly to the group. It may also be problematic to recollect what groups have been created if they are not visible.

Once one or more groups are defined the popup menu offers additional options: to append the row clicked to a specified group (without changing the current selections); to append all selected items to a specified group (Figure 68).

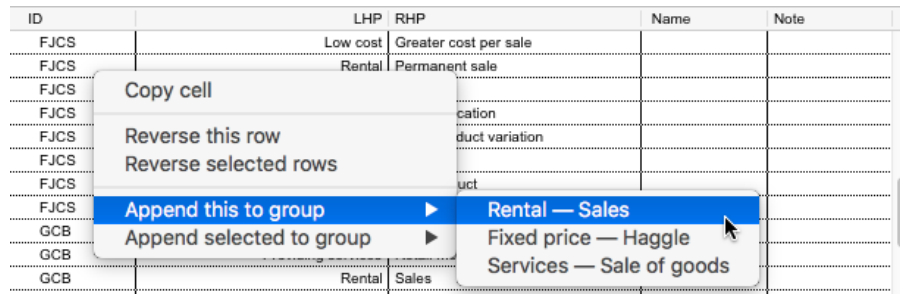


Figure 68: Grouping through the popup menu

The menu lists the groups already defined and the two options provide the flexibility to manage various strategies for grouping. For example, if one is selecting a number of items for grouping and notices an isolated item that belongs in another group, one may group that item without disturbing the existing selections.

6.4 Exporting content analysis data

The content analysis tool itself only groups the grid items making no changes in them and not consolidating in them in any way other than grouping them. The normal presentation of the resulting structures is to display the groups in full (e.g. Jankowicz, 2004, Table 7.9, pp.178-179) or summary form (e.g. Jankowicz, 2004, Table 7.1, p.153). The export function supports copying selected parts of the content analysis data in a TAB-separated format in the clipboard or a file that may be pasted in, or opened by, document processors and spreadsheets to create such presentations.

If the grids have common items then it is also possible to export the content analysis data as a *mode grid* representing a consensual synthesis of the community. This goes beyond that of §3.2 by allowing the *folding* of items (§3.3.2) based not only on lexical identity but also on lexical content such as similarity and common meaning. The ratings of an item are copied directly and of a group are the means of those of its members.

Clicking on the *Export* button brings up the dialogue shown in Figure 69.

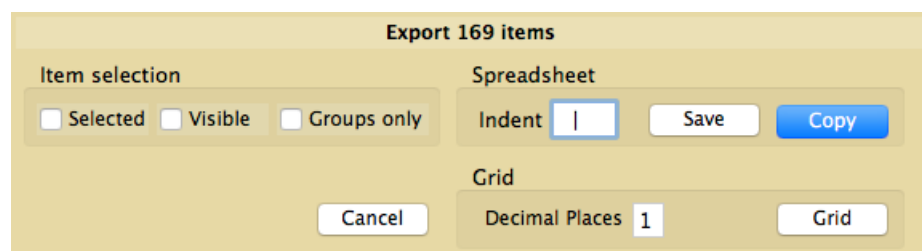


Figure 69: Content analysis export dialogue

The *Item selection* panel on the left specifies what items will be exported:

Selected Export only selected items;

Visible Export only visible items—excludes the contents of groups unless they are open;

Groups only Export groups only.

The text at the top shows the number of items that will be exported and changes according to the eight selection combinations that are possible. The top six items were selected, one of which as a group having three members, and two other groups have been created. The number of items ranges from 1 for selected groups only, through 3 for groups only, 6 for selected visible, 9 for selected (includes members of closed groups), through to 169 as shown for all items without restriction.

The *Spreadsheet* panel on the top right specifies that the items will be exported in a TAB-separated format suited to spreadsheets and document processors, either as a file or copied to the clipboard. The *Indent* field supports hierarchical structures by indenting the members of a group if they are exported (with multiple levels of indents if groups are included in groups).

Figure 70 shows the selected items exported to the clipboard and pasted into a spreadsheet. If visible items only had been exported then the three group members would not appear.

	A	B	C	D	E	F
1	ID	Name	LHP	RHP	Note	# %
2	AH		More profits	Less profits		
3	AH		Higher volume	Lower volume		
4	AH		More work	Less work		
5	Group		Rental	Sales		3 1.8
6	AH		Rental	Sales		
7	EM		Being hired	Being sold		
8	FALC		Renting	Sales		
9	AH		No continuous capital required	Continuous capital outlay		
10	AH		Public benefit	Private benefit		

Figure 70: Content analysis data exported to a spreadsheet

The columns exported are those visible as controlled by the check boxes that open and close the columns. Thus, through the *Item selection* panel and the column check boxes the user has a high degree of control over what parts of the content analysis are exported.

The *Grid* panel on the bottom right specifies that the items will be exported as a mode grid (Figure 71). A group will be exported as a construct having the name and poles specified with ratings for any common elements being the mean of those of the members having the number of decimal places specified, and a weight corresponding to the number of constructs included. If the number of decimal places specified is zero then the construct type will be *Ratings*, otherwise *Numbers*. If there are no common elements the grid will have no elements.

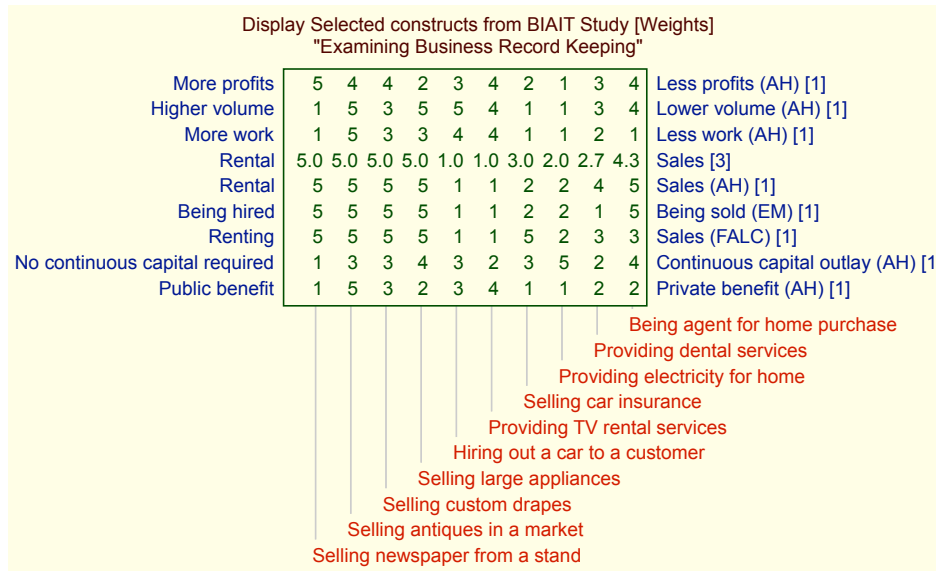


Figure 71: Content analysis data exported as a grid

Usually only the groups created in the content analysis will be exported to a grid, possibly with some of the ungrouped items, but the user has the flexibility to specify whatever might be of interest.

6.5 Numerical measures supporting content analysis

As discussed in §6.1, the content analysis of grid data can be guided not only by the textual material but also by numeric measures derived from the ratings data. Items may be grouped based on some combination of common meaning and common usage.

Honey (1979) developed a technique to support content analysis of grids that involves seeding all the grids being elicited with one or more *given*, or *supplied*, constructs that represent the purpose of the study. Matches to these are used to identify constructs in each grid that are significant to a study, and to align them so that their similarity is more apparent. Jankowicz further developed and exemplified this technique, and presented it in tutorial form (Jankowicz, 2004).

If the grids being analyzed have common elements (or constructs) then Shaw's (1980) *mode* score can be computed (§3.2). This is the mean across grids of the highest match of an item to all items in every other grid, and measures the degree of *consensuality* of the item. In addition the matrix of matches between all items can be made available to check the degree of common usage in a group.

The two techniques provide different information relevant to a content analysis and may sensibly be combined to provide maximal support from analysis of the grid ratings to guide a content analysis based on the conceptual relationships evident in the textual material.

6.5.1 Honey/Jankowicz indices

Honey's technique requires the specification one or more *given* constructs that are created to correspond to the purpose of the study, for example in a study of *mapping techniques* one might use *least effective technique*—*most effective technique* of in a study of *fraud* one might use *least serious fraud*—*most serious fraud*. A given construct in each grid is matched against the other constructs in the grid to provide a measure of how relevant they are to the purpose of the study, and to reverse constructs if necessary so that their poles align with the given construct.

Figure 72 shows *Content* pane from the study of a group studying mapping techniques (Shaw and Gaines, 1989) where three constructs were supplied that were central to the aims of the study, *Not very important*—*Very important*, *Not very effective*—*Very effective*, *Not widely used*—*Widely used*.

ID	LHP	RHP	Index
Peter 1987-06-04	Does not honour data points	Honours data points	
Peter 1987-06-04	Local	Global	
Peter 1987-06-04	Low level data	High level data	
Peter 1987-06-04	Mathematically complex	Mathematically simple	
Peter 1987-06-04	Discontinuous	Continuous	
Peter 1987-06-04	Does not require a priori model	Requires a priori model	
Peter 1987-06-04	Short distance autocorrelation	Long distance autocorrelation	
Peter 1987-06-04	Models the stationarity	Assumes stationarity	
Peter 1987-06-04	New geographical technique	Old geographical technique	
Peter 1987-06-04	Hard to adapt to multivariate	Easy to adapt to multivariate	
Peter 1987-06-04	Few points	Many points	
Peter 1987-06-04	Does not require spatial search	Requires spatial search	
Peter 1987-06-04	Does not use polynomial	Uses polynomial	
Peter 1987-06-04	Not very important	Very important	
Peter 1987-06-04	Not very effective	Very effective	
Peter 1987-06-04	Not widely used	Widely used	
Mary 1987-06-04	qualitative and quantitative	quantitative	
Mary 1987-06-04	local	global	
Mary 1987-06-04	autocorrelation not considered	autocorrelation considered	
Mary 1987-06-04	doesn't honour data points	honours data points	
Mary 1987-06-04	multiple variables considered	usually one variable considered	
Mary 1987-06-04	mathematical curve fitting	doesn't fit a mathematical curve	
Mary 1987-06-04	nonparametric	parametric	

Figure 72: A grid data set with three given constructs representing the purpose of the study

The *Index* check box on the left has been set to show a column where the Honey/Jankowicz indices may be displayed. If one double-clicks in a row in the *Index* column the values of the indices are displayed showing the matches of all constructs in the grid to the construct in that row together with Honey's H, I, L (high, intermediate, low) triage of the matches based on their rank order (Figure 73).

Geog			
Grids Socio Histo Compare Content Scripts Title Geog			
ID	LHP	RHP	Index
Peter 1987-06-04	Does not honour data points	Honours data points	I 66.67
Peter 1987-06-04	Global	Local	I 66.67
Peter 1987-06-04	Low level data	High level data	L 55.21
Peter 1987-06-04	Mathematically simple	Mathematically complex	L 61.46
Peter 1987-06-04	Continuous	Discontinuous	H 71.88
Peter 1987-06-04	Requires a priori model	Does not require a priori model	L 63.54
Peter 1987-06-04	Long distance autocorrelation	Short distance autocorrelation	I 64.58
Peter 1987-06-04	Assumes stationarity	Models the stationarity	L 61.46
Peter 1987-06-04	Old geographical technique	New geographical technique	I 67.71
Peter 1987-06-04	Hard to adapt to multivariate	Easy to adapt to multivariate	I 67.71
Peter 1987-06-04	Few points	Many points	H 68.75
Peter 1987-06-04	Does not require spatial search	Requires spatial search	H 70.83
Peter 1987-06-04	Uses polynomial	Does not use polynomial	L 59.38
Peter 1987-06-04	Not very important	Very important	H 79.17
Peter 1987-06-04	Not very effective	Very effective	****
Peter 1987-06-04	Not widely used	Widely used	H 75.00
Mary 1987-06-04	quantitative	qualitative and quantitative	H 72.92
Mary 1987-06-04	global	local	L 56.25
Mary 1987-06-04	autocorrelation not considered	autocorrelation considered	I 71.88
Mary 1987-06-04	doesn't honour data points	honours data points	I 68.75
Mary 1987-06-04	multiple variables considered	usually one variable considered	L 54.17
Mary 1987-06-04	mathematical curve fitting	doesn't fit a mathematical curve	H 73.96
Mary 1987-06-04	nonparametric	parametric	L 62.50

Discard 3 grids, 48 constructs, 12 elements in common Select Reverse

Show ☒ Index 2/3 ☐ Name ☐ Note ☐ # % Socio Export Group Ungroup

Figure 73: Double-clicking in the *Index column* to compute the Honey/Jankowicz indices

Note that eight of the constructs in Peter's grid have been automatically reversed in the indexing process to align them with construct selected for indexing *Not very effective*—*Very effective*.

Note also that the constructs in Mary's grid have also been indexed. This is because there is a lexically identical construct in her grid and the indexing process automatically propagates to all the grids on the assumption they include the same given construct. If this propagation is not appropriate then double-clicking with the shift key down will index only one grid.

The numbers after *Index* checkbox show that 2 out of 3 grids have been indexed. If we scroll down we can see that Charlie's grid was not indexed because he entered the given construct as *not so effective technique*—*very effective technique* (Figure 73). Double-clicking in the *Index* column for that construct completes the indexing process.

ID	LHP	RHP	Index
Mary 1987-06-04	doesn't fit a trend	fits a trend to the data	L 56.25
Mary 1987-06-04	no computing load	heavy computing load	H 80.21
Mary 1987-06-04	assumes isotropic surface	assumes anisotropic surface	L 64.58
Mary 1987-06-04	estimates susceptible to clusters	not as susceptible to clusters	H 80.21
Mary 1987-06-04	doesn't incorporate geologic model	incorporates geologic model	H 73.96
Mary 1987-06-04	representative	interpretive	I 70.83
Mary 1987-06-04	not very important	very important	H 91.67
Mary 1987-06-04	not very effective	very effective	****
Mary 1987-06-04	widely used	not widely used	I 67.71
Charlie 1987-06-05	local	global	
Charlie 1987-06-05	nominal	interval	
Charlie 1987-06-05	no data restrictions	data restrictions	
Charlie 1987-06-05	non-exact interpolation	exact interpolation	
Charlie 1987-06-05	non-continuous	continuous	
Charlie 1987-06-05	periodicity	non-periodicity	
Charlie 1987-06-05	very smooth	non-smooth data	
Charlie 1987-06-05	low computer cost	high computer cost	
Charlie 1987-06-05	no error estimates	error estimates	
Charlie 1987-06-05	deterministic	stochastic	
Charlie 1987-06-05	non-linear surface	linear surface	
Charlie 1987-06-05	not so important technique	very important technique	
Charlie 1987-06-05	not so effective technique	very effective technique	
Charlie 1987-06-05	not widely used	widely used	

Discard 3 grids, 48 constructs, 12 elements in common Select Reverse
 Show ☒ Index 2/3 ☐ Name ☐ Note ☐ # % Socio Export Group
 Ungroup

Figure 74: Propagation of the indexing to lexically identical constructs

Content analysis through grouping and the export of the resulting structures proceeds as already described except that exporting to a grid is no longer useful if there are no common elements. The analysis procedures are well-documented in the literature (Honey, 1979; Jankowicz, 2004), and the RepGrids content analysis tool supports them by carrying out the index calculations, providing facilities to help manage the process, recording the results, and exporting them in a form suitable for preparing their presentation.

6.5.2 Socio match and mode values

In the BIAIT Study the grids have common elements and a *Socio* button appears in the lower panel (Figure 61) which will run a Socio analysis on the grids selected for the content analysis. This analysis is private to the content analysis and will automatically run again when a saved file is reopened.

Figure 75 shows the *Content* pane after a Socio analysis. Two new columns have been generated: *Match* which can be used to show the matches of all constructs to any chosen construct; *Mode* showing the Socio mode grid comparison matches. The *Socio* button has been replaced by check boxes controlling the visibility of the two new columns.

BIAIT CA

Grids Socio Histo Compare Content Scripts Title BIAIT Study

ID	LHP	RHP	Match	Mode
FALC	Renting	Sales	★★★★	84.67
SFG	Rental	Buy	85.00	84.33
RKK	Rental income	Profit	87.50	84.17
HJ	Selling	Rental	● 92.50	84.00
AH	Rental	Sales	85.00	83.17
SMH	Rental or lease	Sale	95.00	82.83
FEP	Return after use	Ownership transfers	82.50	82.67
PEK	Customer looks after	We look after	● 87.50	82.67
PEK	Expensive	Low-cost	● 62.50	82.33
THL	Sales representative	Property	● 87.50	82.17
MEMK	Higher priced	Lower priced	● 70.00	82.17
PEK	Customer owns	We own	● 82.50	82.00
FJCS	Rental	Permanent sale	82.50	82.00
EM	Service	Sales	82.50	81.83
LYCL	Expensive for customer	Affordable for public	● 72.50	81.83
EM	Being hired	Being sold	82.50	81.67
GCB	Tangible	Intangible	● 87.50	81.33
HJ	Done inside	Done outside	● 62.50	81.00
LCL	Don't gain ownership	Gain ownership	80.00	81.00
RKK	Services	Sale of goods	80.00	80.67
FJCS	Older product	New product	65.00	80.50
FEP	Impersonal interaction	Close personal interaction	65.00	80.33
EM	Intangibles	Tangible	82.50	80.17

Discard 16 grids, 166 constructs, 10 elements in common Select Reverse

Show ☐ Index 0/16 ☐ Name ☐ Note ☐ # % ☒ Match ☒ Mode Export Group

6 business expertise is low Add Class Delete Ungroup

Figure 75: Content pane after running Socio

The *Match* column is initially empty but double-clicking in it for a certain construct populates it as shown with the matches to that construct of all the other constructs. The construct selected is indicated by a row of four bullets. A match value corresponding to the best match involving reversing one of the constructs has a bullet in front of it. The construct itself may be reversed by selecting it and clicking on the *Reverse* button and the bullet will then disappear. The matches are dynamic and the construct double-clicked for matching can be reselected as required.

The *Mode* column is initially unsorted but clicking on the header sorts it as shown with the constructs in the order as they would be selected for incorporation in a mode grid. This enables the generation of a mode grid to be managed more flexibly than in the *Socio* pane, and the mode match values to be used to guide a content analysis that commences with the most consensual constructs.

The top eight constructs seem all to be related to the distinction *rent—purchase* and we may consider grouping them. The fourth one from HJ needs to be reversed before grouping. The third one from RKK addresses an aspect of this distinction that renting leads to a continuing income and purchasing to an immediate one-off profit. If the content analysis is to be presented as clusters of associated conceptual distinctions then this enriches the cluster. If the analysis is to be exported to a grid and presented through grid analyses then it may be appropriate to leave that construct out of the main cluster so that the conceptual correspondence is apparent through the *Focus* clustering or *PrinGrid* map.

Similar consideration apply to for PEK's *Customer looks after*—*We look after*, FEP's *Return after use*—*Ownership transfers*, EM's *Service*—*Sales* and RKK's *Services*—*Sale of goods*. In addition these last two seem be conceptually related and we can consider grouping them and, possibly, including that group in the *rent*—*purchase* purchase group.

Figure 76 shows such a hierarchical grouping including appropriate reversals to align all the constructs. The main group for *rent*—*purchase* has a subgroup of associated notions making corresponding distinctions that are conceptually different but functionally related to *rent*—*purchase*. This in turn has a subgroup for the various forms of the *service*—*sales* distinction.

ID	LHP	RHP	Name	Match	Mode	# %
▼ Group	Rent	Purchase				11 6.6
FALC	Renting	Sales		****	84.67	
SFG	Rental	Buy		85.00	84.33	
HJ	Rental	Selling		92.50	84.00	
AH	Rental	Sales		85.00	83.17	
SMH	Rental or lease	Sale		95.00	82.83	
FJCS	Rental	Permanent sale		82.50	82.00	
▼ Group	Rent	Purchase	Corresponds			5 3.0
RKK	Rental income	Profit		87.50	84.17	
FEP	Return after use	Ownership transfers		82.50	82.67	
PEK	We look after	Customer looks after		87.50	82.67	
▼ Group	Service	Sales				2 1.2
EM	Service	Sales		82.50	81.83	
RKK	Services	Sale of goods		80.00	80.67	
PEK	Expensive	Low-cost		● 62.50	82.33	
THL	Sales representative	Property		● 87.50	82.17	

Figure 76: Hierarchical grouping of constructs corresponding to *rent*—*purchase*

The export capabilities allow many possibilities, such as the top-level group, the three groups, or the entire structure to be output as a basis for presenting the results. Figure 77 shows the entire structure exported to a spreadsheet. The use of indentation to represent the hierarchy is apparent. The structure suggests a presentation where the main distinction is between renting and purchasing and that this has correlated aspects of long term income versus immediate profit, allowing for the return and maintenance of rental items, and the notion of providing a service.

	A	B	C	D	E	F	G
1	ID	LHP	RHP	Name	Match	Mode	# %
2	Group	Rent	Purchase				11 6.6
3	FALC	Renting	Sales		****	84.67	
4	SFG	Rental	Buy		85.00	84.33	
5	HJ	Rental	Selling		92.50	84.00	
6	AH	Rental	Sales		85.00	83.17	
7	SMH	Rental or lease	Sale		95.00	82.83	
8	FJCS	Rental	Permanent sale		82.50	82.00	
9	Group	Rent	Purchase	Corresponds			5 3.0
10	RKK	Rental income	Profit		87.50	84.17	
11	FEP	Return after use	Ownership transfers		82.50	82.67	
12	PEK	We look after	Customer looks after		87.50	82.67	
13	Group	Service	Sales				2 1.2
14	EM	Service	Sales		82.50	81.83	
15	RKK	Services	Sale of goods		80.00	80.67	

Figure 77: Constructs corresponding to *rent*—*purchase* exported to spreadsheet

Similarly, there are several relevant possibilities when exporting the group to a grid. Figure 78 shows the entire contents of the group exported as a grid.

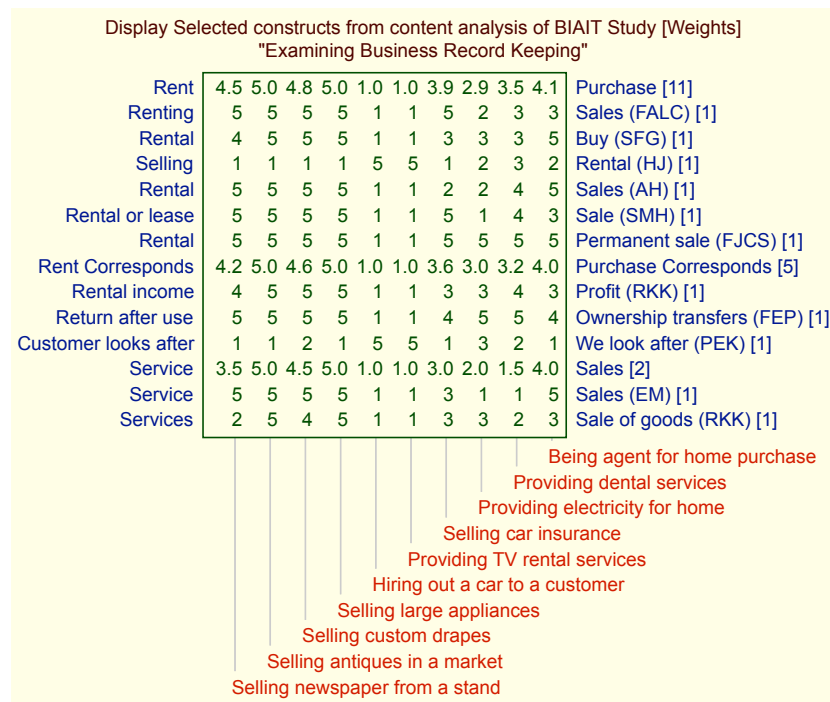


Figure 78: Constructs corresponding to *rent*—*purchase* exported to a grid

Figure 79 shows the top level group exported as a grid with a single construct having the mean ratings of the eleven constructs in the group.

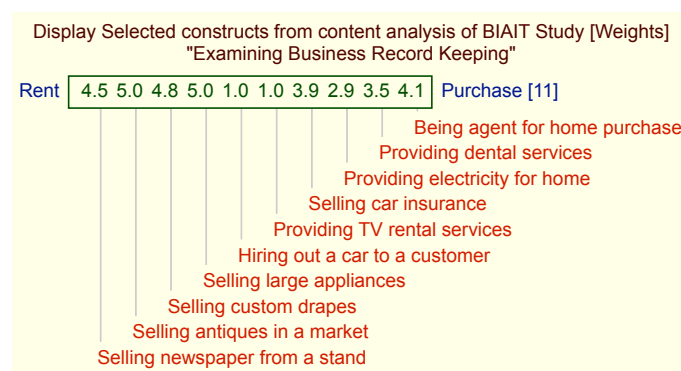


Figure 79: Top level exported to a grid

Figure 80 shows the three groups exported as a grid. The ratings of the first construct are the mean of all eleven constructs and include those of the other two groups, and, similarly, the ratings of the second are the mean of five constructs including those in the third group.

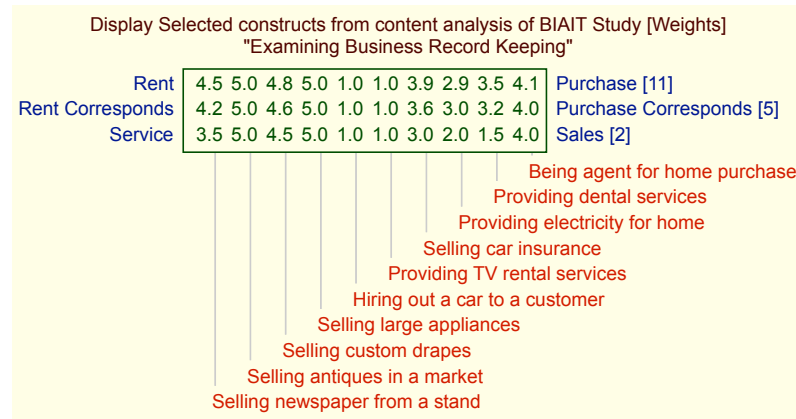


Figure 80: Three groups exported to a grid

Figure 81 shows the three groups exported as a grid after the two subgroups have been dragged outside the main group to be at the same level. The ratings are now only averaged over those of the separate groups.

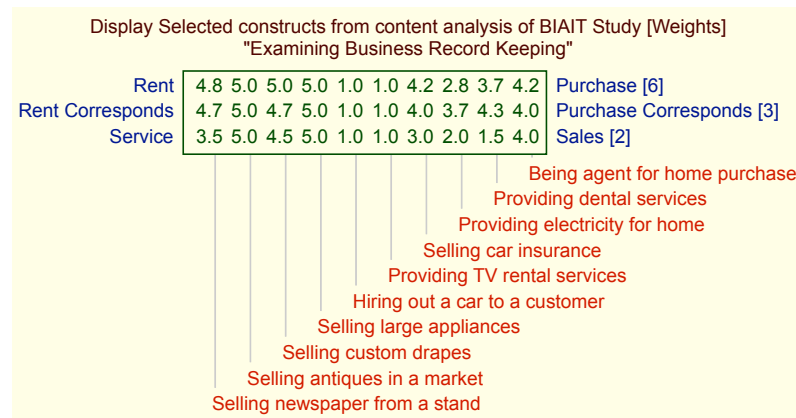


Figure 81: Three groups separated at same level exported to a grid

Thus, the support of hierarchical structures together with the ease of modifying them and the various export options makes it possible to export the data from the content analysis in a wide variety of forms that may be chosen to address the needs of any particular study.

6.5.3 Aligning through the popup menu

In the content analysis of a collection of constructs it is important to make sure the poles of the constructs in a group are aligned, particularly if the groups are to be exported to a grid. This needs to be checked on an individual basis and can be done on an individual basis. However, the ratings provide a basis for alignment that can be used to automate the alignment process in large part.

If the Honey/Jankowicz techniques have been used then setting the given construct in the *Index* column automatically aligns all the other constructs in the same grid and this will usually lead to conceptual alignment when constructs from different grids are grouped.

A similar approach to achieving alignment may be taken when a Socio analysis has been applied and the *Match* and *Mode* values are available. if one clicks on the header of the *Mode* column

to bring the most consensual mode construct to the top, and then clicks in the *Match* column for that construct to show all the matches to that construct, then the constructs that are not aligned to that construct are apparent (Figure 82).

ID	LHP	RHP	Match	Mode^	#	%
FALC	Renting	Sales	★★★★	84.67		
SFG	Rental	Buy	85.00	84.33		
RKK	Rental income	Profit	87.50	84.17		
HJ	Selling	Rental	● 92.50	84.00		
AH	Rental	Sales	85.00	83.17		
SMH	Rental or lease	Sale	95.00	82.83		
FEP	Return after use	Ownership transfers	82.50	82.67		
PEK	Customer looks after	We look after	● 87.50	82.67		
PEK	Expensive	Low-cost	● 62.50	82.33		
THL	Sales representative	Property	● 87.50	82.17		
MEMK	Higher priced	Lower priced	● 70.00	82.17		
PEK	Customer owns	We own	● 82.50	82.00		
FJCS	Rental	Permanent sale	82.50	82.00		

Figure 82: Sorting by mode score and matching other constructs to that with the highest score

If Socio has been run and one right-clicks or CTL-clicks in a row, additional options appear in the popup menu: to align all constructs to that in the row; to align any selected constructs to that in the row (Figure 83). Selecting the first option aligns all the unaligned constructs and leaves those reversed selected to show which ones were changed (Figure 84).

ID	LHP	RHP	Match	Mode^	#	%
FALC	Renting	Sales	★★★★	84.67		
SFG	Rental	Buy	85.00	84.33		
RKK	Rental income	Profit	87.50	84.17		
HJ	Selling	Rental	● 92.50	84.00		
AH	Rental	Sales	85.00	83.17		
SMH	Rental or lease	Sale	95.00	82.83		
FEP	Return after use	Ownership transfers	82.50	82.67		
PEK	Customer looks after	We look after	● 87.50	82.67		
PEK	Expensive	Low-cost	● 62.50	82.33		
THL	Sales representative	Property	● 87.50	82.17		
MEMK	Higher priced	Lower priced	● 70.00	82.17		
PEK	Customer owns	We own	● 82.50	82.00		
FJCS	Rental	Permanent sale	82.50	82.00		

Figure 83: Popup menu alignment options

ID	LHP	RHP	Match	Mode^	#	%
FALC	Renting	Sales	★★★★	84.67		
SFG	Rental	Buy	85.00	84.33		
RKK	Rental income	Profit	87.50	84.17		
HJ	Rental	Selling	92.50	84.00		
AH	Rental	Sales	85.00	83.17		
SMH	Rental or lease	Sale	95.00	82.83		
FEP	Return after use	Ownership transfers	82.50	82.67		
PEK	We look after	Customer looks after	87.50	82.67		
PEK	Low-cost	Expensive	62.50	82.33		
THL	Property	Sales representative	87.50	82.17		
MEMK	Lower priced	Higher priced	70.00	82.17		
PEK	We own	Customer owns	82.50	82.00		
FJCS	Rental	Permanent sale	82.50	82.00		

Figure 84: Constructs aligned to that with the highest mode score

Aligning all the constructs to the most consensual one often leads to all similar constructs being aligned, but will not necessarily do so. If when grouping constructs one notes that many are not aligning then the alignment options may be used to align all ungrouped or selected constructs to a representative construct for the group. Alignment is only applied to visible constructs and not to those in closed groups or to the groups themselves making it easy to manage the process even without selection.

6.6 Textual search

Content analysis involves identifying conceptually related items and grouping them, and it can be a time-consuming and tedious process. Numerical measures derived from the grid data can help identify significant items and commonalities of construing that support the process but there remains an essential element of lexical interpretation which can be supported by text search facilities.

When the content pane has been populated a *Find* command becomes available in the *Edit* menu and can be activated from the menu or by keying command-F. This brings up a search dialogue (Figure 85).

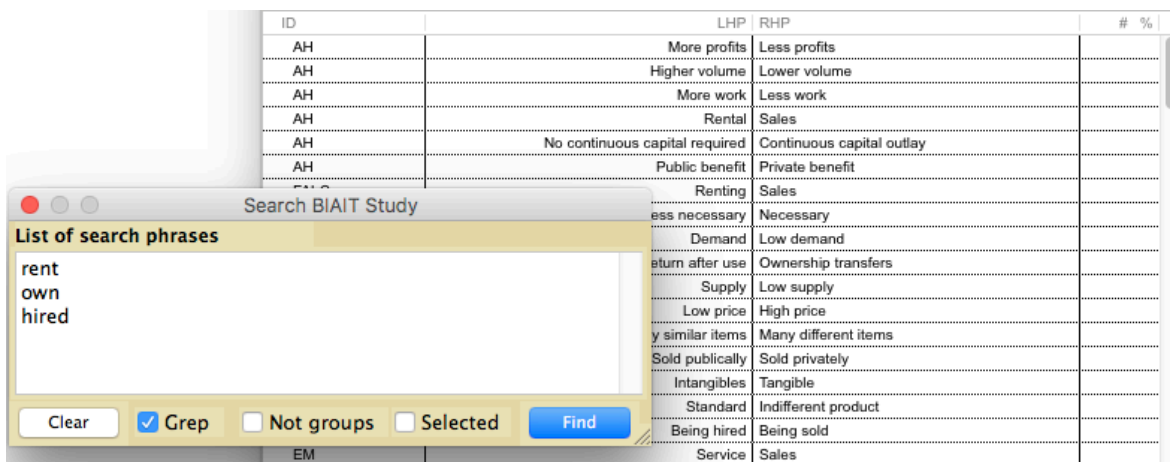
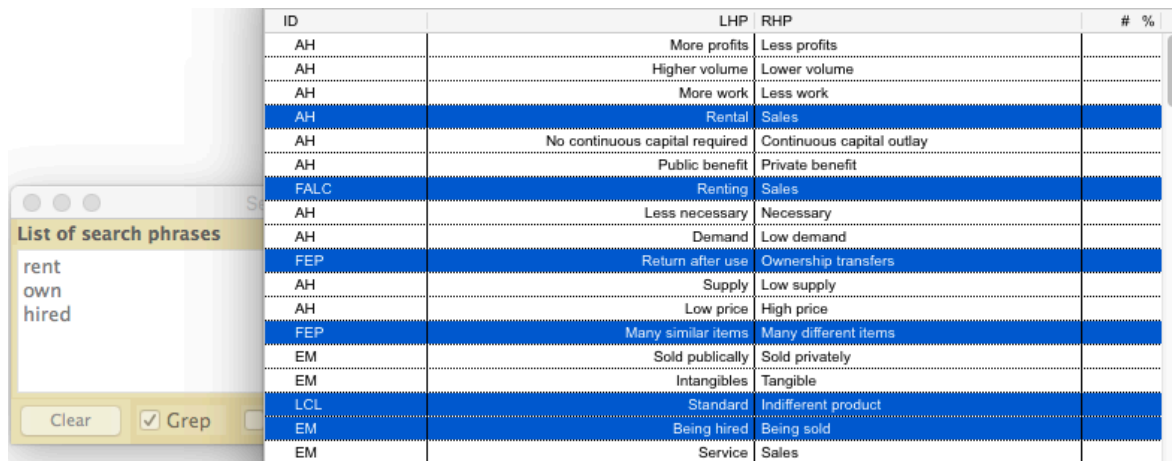


Figure 85: Search dialogue

Any number of words or phrases may be entered, one on each line. Each will be matched against the item identification fields (name, note and poles) that are visible, and the items matching any of the search phrases will be selected (Figure 86), including some unintended (*different* includes *rent*).



The screenshot shows a search results window. On the left, a 'List of search phrases' box contains 'rent', 'own', and 'hired'. Below it are 'Clear' and 'Grep' buttons. The main area is a table with columns: ID, LHP, RHP, and # %.

ID	LHP	RHP	# %
AH	More profits	Less profits	
AH	Higher volume	Lower volume	
AH	More work	Less work	
AH	Rental	Sales	
AH	No continuous capital required	Continuous capital outlay	
AH	Public benefit	Private benefit	
FALC	Renting	Sales	
AH	Less necessary	Necessary	
AH	Demand	Low demand	
FEP	Return after use	Ownership transfers	
AH	Supply	Low supply	
AH	Low price	High price	
FEP	Many similar items	Many different items	
EM	Sold publicly	Sold privately	
EM	Intangibles	Tangible	
LCL	Standard	Indifferent product	
EM	Being hired	Being sold	
EM	Service	Sales	

Figure 86: Search results

The *Content* pane comes to the front to show the results of a search. The search dialogue remains visible because one outcome of a search can be to amend the search terms. It can be hidden by clicking on the red dot at the top left. It is also stored in the RepGrids file. If it is activated again the search terms entered will still be present and the state of the check boxes will be unchanged. Clicking on the *Clear* button clear the search phrases in preparation for a new search.

The three checkboxes at the bottom of the search dialogue control the search process:

Grep Interpret the search phrase as a pattern-matching *regular expression* (Fitzgerald, 2012) such as *(\b)rent*;

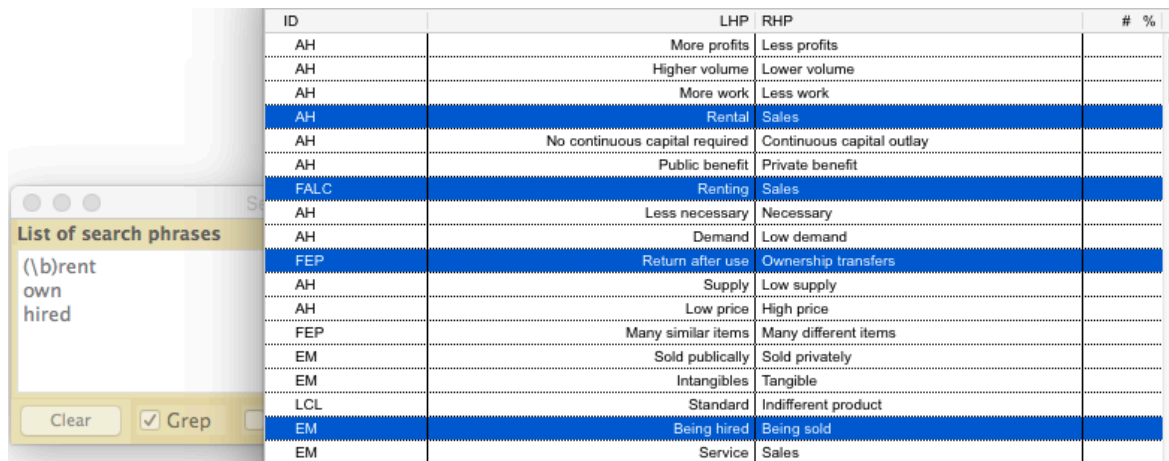
Not groups Do not include groups in the search—they are usually not intended to be further grouped;

Selected Only search selected items—note that any prior selection may be changed as an outcome of the search.

The *Grep* option allows complex pattern-matching search phrases to be defined to make the search more specific. Since an alphanumeric string is itself a regular expression, normal search phrases including only alphanumerics and spaces may be freely mixed in the list with those having a more complex interpretation.

The *Grep* option is most often useful to ensure that a search term is not embedded. For example, *rent* alone will match *rental* or *rented* as intended but it is common suffix and will also match *different*, *apparent*, and so on. The regular expression *(\b)rent* will only match text that begins with *rent*. The *\b* specifies the boundary of a word. The surrounding parentheses are unnecessary but good practice to separate the rather opaque syntax from the rest of the text.

If *\b* is prefixed to the search phrase *rent* then the unintended hits on *different* and *indifferent* no longer appear (Figure 87).



ID	LHP	RHP	#	%
AH	More profits	Less profits		
AH	Higher volume	Lower volume		
AH	More work	Less work		
AH	Rental	Sales		
AH	No continuous capital required	Continuous capital outlay		
AH	Public benefit	Private benefit		
FALC	Renting	Sales		
AH	Less necessary	Necessary		
AH	Demand	Low demand		
FEP	Return after use	Ownership transfers		
AH	Supply	Low supply		
AH	Low price	High price		
FEP	Many similar items	Many different items		
EM	Sold publicly	Sold privately		
EM	Intangibles	Tangible		
LCL	Standard	Indifferent product		
EM	Being hired	Being sold		
EM	Service	Sales		

Figure 87: More selective search results

The results may be scrolled down to check that all should be grouped, any that are inappropriate may be deselected (command-click), and the popup menu may be used to append the selected constructs to the appropriate group.

6.7 Classifying the groupings

If the population from whom the grids have been elicited has been stratified into classes then it is often of interest to know how many grids from each class contributed to a group of items. If one or more class grids representing the stratification are part of the dataset then a menu is shown at the bottom of the *content pane* allowing a class to be selected together with buttons allowing column for that class to be added to, or deleted from, the analysis.

For example, those from whom the grids were elicited in the study of accounting eliciting of business process expertise (§6.2) were stratified in terms of their business expertise and this was represented in a class metagrid added to the dataset (Figure 88).

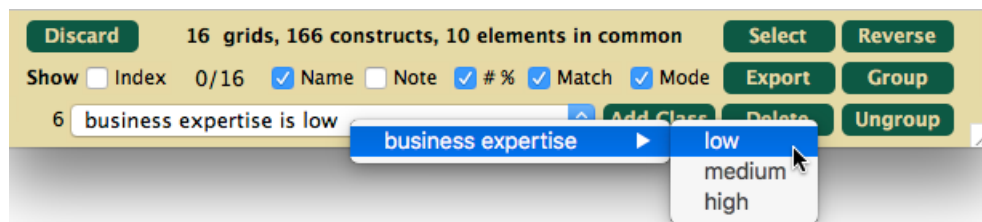


Figure 88: Classes stratifying those from whom grids were elicited

If one selects each class and clicks on *Add Class* then three columns are added which show the number and percentage of the grids in each class that are members of the group (Figure 89).

ID	LHP	RHP	Name	Match	Mode	#	%	low	medium	high
▼ Group	Rent	Purchase	Rent			11	6.6	3 4.8	4 6.6	4 9.3
FALC	Renting	Sales		★★★★	84.67					
SFG	Rental	Buy		85.00	84.33					
HJ	Rental	Selling		92.50	84.00					
AH	Rental	Sales		85.00	83.17					
SMH	Rental or lease	Sale		95.00	82.83					
FJCS	Rental	Permanent sale		82.50	82.00					
▼ Group	Rent	Purchase	Corresponds			5	3.0	2 3.2	1 1.6	2 4.7
RKK	Rental income	Profit		87.50	84.17					
FEP	Return after use	Ownership transfers		82.50	82.67					
PEK	We look after	Customer looks after		87.50	82.67					
▼ Group	Service	Sales				2	1.2	1 1.6	1 1.6	0 0.0
EM	Service	Sales		82.50	81.83					
RKK	Services	Sale of goods		80.00	80.67					
PEK	Expensive	Low-cost		● 62.50	82.33					
THL	Sales representative	Property		● 87.50	82.17					

Figure 89: Class counts and percentages of classes

Note that the percentage in the # % column is that of the total number of items in the *Content* pane, and that in the class columns is that of the total number of items in the specified class.

The additional data is included when the content analysis is exported to a spreadsheet (Figure 90).

	A	B	C	D	E	F	G	H	I	J
1	ID	LHP	RHP	Name	Match	Mode	# %	low	medium	high
2	Group	Rent	Purchase	Rent			11 6.6	3 4.8	4 6.6	4 9.3
3	FALC	Renting	Sales		****	84.67				
4	SFG	Rental	Buy		85.00	84.33				
5	HJ	Rental	Selling		92.50	84.00				
6	AH	Rental	Sales		85.00	83.17				
7	SMH	Rental or lease	Sale		95.00	82.83				
8	FJCS	Rental	Permanent sale		82.50	82.00				
9	Group	Rent	Purchase	Corresponds			5 3.0	2 3.2	1 1.6	2 4.7
10	RKK	Rental income	Profit		87.50	84.17				
11	FEP	Return after use	Ownership transfers		82.50	82.67				
12	PEK	We look after	Customer looks after		87.50	82.67				
13	Group	Service	Sales				2 1.2	1 1.6	1 1.6	0 0.0
14	EM	Service	Sales		82.50	81.83				
15	RKK	Services	Sale of goods		80.00	80.67				
16	PEK	Expensive	Low-cost		* 62.50	82.33				
17	THL	Sales representative	Property		* 87.50	82.17				

Figure 90: Class counts and percentages of classes in exported content analysis

7 Scripts Pane

Clicking on the *Scripts* tab in the RepGrids window brings up the Scripts pane (Figure 91). The *Scripts* tab at the top right has been replaced by a scripts menu allowing the selection of a script from those available in the network of *GridsScripts* directories in the network of such directories accessible to the current RepGrids file (§8.2).

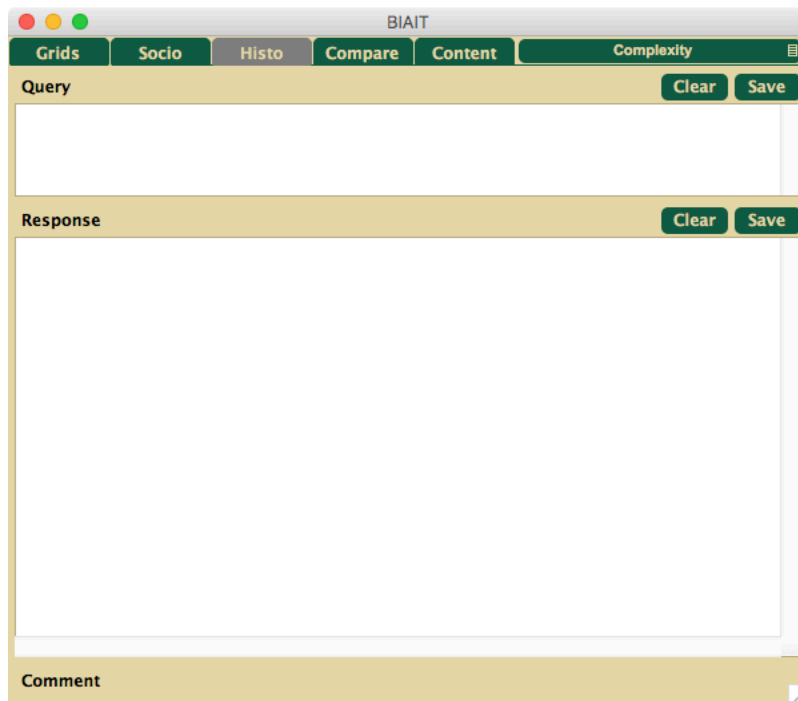


Figure 91: RepGrids window showing the **Scripts** pane

Clicking in the menu symbol on the right of the scripts menu brings up a menu enabling a script to be selected from those available. Clicking on the scripts menu when one has been selected runs the script.



Figure 92: Selecting a script from those available

The menu is shown open with five scripts available:

Complexity Show cognitive complexity measures for the selected grids;

Get Socio Data Show the data used in a *Socio* analysis—grids, matches, mode scores, sorted mode scores;

Grid Data Show the normalized ratings for each selected grid for common elements on common constructs;

Grid Overviews List the elements and constructs in the selected grids;

Test Editing Grids Exemplifies how to open each of the selected grids, edit it as specified and save it—can be run safely and used to test code accessing the grid data as the command to save the grid is commented out.

The scripts are intended to provide examples of how RepScript programs may be used extend RepGrids by accessing and processing its data.

There are two text panels available in the window whose size visibility and content are under script control, the upper one intended for entering queries, and the lower one for the script-generated responses. The labels, “Query,” “Response” and “Comment” are also under script control and generally change according to the script. At the top left is a button to run scripts with a popup menu on its right that allows the script to be run to be selected.

The scripts are written in *RepScript* which is documented in the RepScript manual, including the special commands for RepGrids. Scripts for use in RepGrids are stored in a *GridsScripts* directory (§8.2). RepGrids looks for GridsScripts directories, in order of increasing precedence, in the: application directory; the *Rep Plus* directory in the *Documents* or *My Documents* directories; any GridsScripts directory in the directory from which a set of files have been added, or in directories containing that directory. The *Rep Plus* directory is intended for global scripts developed by the user, and the directories associated with particular collections of files for scripts local to those files.

When a script is run the stream named *q* is set to send its output to the query field, and that name *r* to the response field. In addition the default output stream with a blank name is also initially set to send its output to the response field. This make it easy to write scripts that can send their primary output either to the response field or to some other text window.

7.1 Complexity script

This script demonstrates how to obtain a principal components analysis of each selected grid and use it to compute two complexity measures, the percentage variance in the first component, and the estimated number of components. The latter is available for individual grids in RepGrid’s PrinGrid and Synopsis scree plots, but it is useful to be able to view it for a collection of grids in RepGrids (Figure 93).

The script is structured as a set of subroutines followed by the main program, but the order is irrelevant as the script is fully compiled before it is run. The main work is done by the routine WriteGrids which gets a tab-separated list of the numbers of the selected grids and then iterates through them. The command *SocioGrid(n)* makes the specified grid active so that the *gGet* and *gSet* commands may be used to access its content.

Complexity Measures for Selected Grids			
	Grid	PVAFF	Components
1:	MEMK	25.38	7.4
2:	LYCL	30.51	5.2
3:	GCB	32.88	5.1
4:	FJCS	26.38	4.7
5:	SFG	33.99	4.3
6:	HJ	28.06	4.3
7:	FALC	34.84	4.2
8:	LCCL	41.08	4.0
9:	SMH	33.47	3.9
10:	RKK	32.28	3.1
11:	EM	45.62	3.0
12:	MSC	34.95	2.9
13:	THL	42.73	2.8
14:	BIAIT	45.68	2.3
15:	PEK	41.62	2.3
16:	AH	56.93	1.9
17:	FEP	40.16	1.9

Figure 93: Complexity estimates for the BIAIT dataset

```
// Demonstration script that provides some complexity measures for the selected grid
// PVAFF is value of first component, Components is Frontier's estimate of number of significant
// components
// The output is sorted with the highest number of components first

Class DoublePair
    dim a As Double
    dim b As Double
    Sub Constructor(ana As Double, anb As Double)
        a=ana
        b=anb
    End Sub
End Class

Function Eigen() As Double()
    dim M(-1),N(-1,-1) As Double
    return gGetEigen(false,false,0.0,M,N,N) // no weightings and cut off zero
End Function

// Frontier's expected values of eigenValues
Function EVDistrib(bnd As Integer) As Double()
    dim i,j As Integer, n As Integer=bnd+1, x,tot,d() As Double
    redim d(bnd)
    for i=1 to n
        x=0.0
        for j=i to n
            x=x+1.0/j
        next
        d(i-1)=x
        tot=tot+x
    next
    for i=0 to bnd
        d(i)=d(i)*100.0/tot
    next
    return d
End Function

// Where do Frontier's expected values cross actual eigenValues
Function EVCross(EV() As Double, DI() As Double) As DoublePair()
    dim a,y1,y2,y3,y4 As Double, DP() As DoublePair, lower As Boolean, i,k As Integer, bnd As
    Integer=EV.UBound
    for i=bnd downto 0
        if lower and EV(i)<DI(i) then
            lower=not lower
        elseif not lower and EV(i)>=DI(i) then
            if i=bnd then
```

```

        DP().Append new DoublePair(1.0+i,EV(bnd)) // last component larger -- no intersection
    else
        y1=EV(i)
        y2=EV(i+1)
        y3=DI(i)
        y4=DI(i+1)
        a=(y1-y3)/(y1-y2-y3+y4)
        DP().Append new DoublePair(1.0+i+a,y1*(1-a)+y2*a)
    end if
    lower=not lower
end if
next
return DP
End Function

Sub WriteGridsComplexity()
    dim n As Variant, i As Integer, param As String=SocioGet("Settings"), id As Integer=sGetI(param,6) // IDflags
    dim sel() As Integer=SocioGetAI("Select"), bnd As Integer=sel.UBound // get list of selected grid numbers
    dim k() As Integer, PV(),CP() As Double, GR() As String
    redim k(bnd)
    redim GR(bnd)
    redim PV(bnd)
    redim CP(bnd)

    for i=0 to bnd
        n=sel(i) // get the number of a selected grid
        SocioSet("Grid",n) // make the grid active (data available through gGet functions)
        dim V() As Double=Eigen()
        dim DP() As DoublePair=EVCross(V,EVDistrib(V.UBound))
        k(i)=i
        GR(i)=gGet("Identifier",str(id))
        PV(i)=V(0)
        CP(i)=DP(DP.UBound).a
    next
    CP.Sortwith(GR,PV)

    writeln TAB+"Grid"+TAB+"PVAFF"+TAB+"Components"
    for i=bnd downto 0
        writeln sFormat(bnd-i+1,"###")+":"+TAB+GR(i)+TAB+sFormat(PV(i),"#.00")+TAB+sFormat(CP(i),"#.0")
    next
    wflushln
End Sub

Sub Main()
    SocioSet("Configure","R") // configure for responses only
    SocioSet("Field","R","Complexity Measures for Selected Grids") // set the heading of the response field
    WriteGridsComplexity
end sub

// main program
Main

```

7.2 Get Socio Data script

This script demonstrates how to run a Socio analysis of the selected grids and display the sociogrids and mode grid data (Figure 94) as can be done in Socio itself (Figure 20). The script can be modified

to extend the processing of the Socio analysis data and/or export it in any desired form for use in other programs

Socio Analysis of 7 Selected Grids									
Grid Matches for 7 grids with 12 elements in common									
	1	2	3	4	5	6	7		
1	100.00	79.80	78.42	86.36	79.69	78.37	84.67	Peter	1987-06-04
2	81.51	100.00	80.88	81.91	89.27	82.85	83.85	Mary	1987-06-04
3	77.99	79.98	100.00	76.70	78.54	89.34	78.20	Charlie	1987-06-05
4	81.18	78.36	75.30	100.00	77.60	75.40	79.32	Peter	1987-08-19
5	78.71	85.53	77.68	81.53	100.00	82.21	79.76	Mary	1987-08-20
6	75.39	79.75	88.84	74.53	80.21	100.00	76.86	Charlie	1987-08-21
7	85.94	83.45	79.91	85.13	81.35	80.21	100.00	Peter	1990-08-07
								Peter 1987-06-04	
								Charlie 1987-06-05	
								Peter 1987-08-19	
								Mary 1987-08-20	
								Charlie 1987-08-21	
								Peter 1990-08-07	
Sorted Grid Matches for 7 grids with 12 elements in common									
1:	89.34	Charlie 1987-06-05	→	Charlie 1987-08-21					
2:	89.27	Mary 1987-06-04	→	Mary 1987-08-20					
3:	88.84	Charlie 1987-08-21	→	Charlie 1987-06-05					
4:	86.36	Peter 1987-06-04	→	Peter 1987-08-19					
5:	85.94	Peter 1990-08-07	→	Peter 1987-06-04					
6:	85.53	Mary 1987-08-20	→	Mary 1987-06-04					
7:	85.13	Peter 1990-08-07	→	Peter 1987-08-19					
8:	84.67	Peter 1987-06-04	→	Peter 1990-08-07					
9:	83.85	Mary 1987-06-04	→	Peter 1990-08-07					
10:	83.45	Peter 1990-08-07	→	Mary 1987-06-04					

```

dim n As Integer=(cbnd+1)*wFormat("")
write "      "
for c=0 to cbnd
    writeI(c+1)
next
writeln
write " ---|"
for i=1 to n
    write "-"
next
writeln
for r=0 to rbnd
    write sFormat(r+1,"##0")+ " |"
    for c=0 to cbnd
        writeD(m(r,c))
    next
    if noid then writeln " "+gGet("C",str(r)) else writeln " "+ID(r)
next
write("      ")
for c=0 to cbnd
    writeS("|")
next
writeln
for c=cbnd downto 0
    write("      ")
    for i=c-1 downto 0
        writeS("|")
    next
    writeS("",-1)
    if noid then writeln gGet("E",str(c)) else writeln ID(c)
next
writeln
End Sub

Sub WriteLinks(a,) As Integer, m(,) As Double, ID() As String, link As String)
dim i,r,c As Integer, bnd As Integer=UBound(a,1)
for i=bnd downto 0
    r=a(i,0)
    c=a(i,1)
    writeln sFormat(bnd-i+1,"##0")+ ":"+sFormat(m(r,c),"###0.00")+ " "+ID(r)+link+ID(c)
next
writeln
End Sub

Sub WriteModeSort(a,) As Integer, m() As Double, GN() As Integer, ID() As String, item As String
)
dim i,g,k As Integer, bnd As Integer=UBound(a,1)
for i=bnd downto 0
    g=a(i,0)
    k=a(i,1)
    writeln sFormat(i+1,"##0")+ ":"+sFormat(m(i),"###0.00")+ " "+GetGridItem(g,k,GN,ID,item)
next
writeln
End Sub

Sub WriteSDSort(a,) As Integer, av(,) As Double, sd(,) As Double)
dim i As Integer, c,e As Variant, bnd As Integer=UBound(a,1)
writeln "      SD   Mean"
for i=0 to bnd
    c=a(i,0)
    e=a(i,1)
    writeln sFormat(i+1,"###0")+ ":"+sFormat(sd(c,e),"##0.00")+sFormat(av(c,e),"##0.00")+ " "+gGet
        ("C",c)+": " +gGet("E",e)
next
writeln

```

```

End Sub

Sub Main()
    SocioSet("Configure","R") // configure for responses only
    SocioSet("Field","R","Selected Grids") // set heading of response field
    Call wFormat("-##0.00")
    dim notrun As Boolean, link,tit As String, par As String=SocioGet("Socio","") // get socio
        analysis status

    // examples of how to change parameters before running Socio
    // hSet(0,"DP")
    // hSet("New Title","Title")
    // hSet(1,"ID")
    // hSet(2,"Power")
    // hSet("E","State")
    // SocioSet("Socio")
    // par="" // ensure recomputation

    if par="" then // run Socio if it has not run
        SocioSet("Field","R","Socio Analysis running")
        par=SocioGet("Socio") // run Socio
    end if

    dim ng As String=hGet("NG"), ne As String=hGet("NE"), nc As String=hGet("NC")
    select case par
        case "" // Socio not run
            notrun=true
        case "E" // same elements
            tit=ng+" grids with "+ne+" elements in common"
            link=" --> "
        case "C" // same constructs
            tit=ng+" grids with "+nc+" constructs in common"
            link=" --> "
        case "X" // both
            tit=ng+" grids with "+ne+" elements and "+nc+" constructs in common"
            link=" -- "
    end select
    if notrun then
        SocioSet("Field","R","Socio Analysis has not been run yet")
    else
        SocioSet("Field","R","Socio Analysis of "+ng+" Selected Grids")

        // grids analyzed
        dim GN() As Integer=SocioGetAI("Select"), ID() As String=SocioGetA("ID")
        Title "Grids Analyzed: "+tit
        WriteGrids(GN,ID)

        // socionets matches
        dim matches(-1,-1) As Double=SocioGetA2D("Socio")
        Title "Grid Matches for "+tit
        WriteMatrixD(matches,ID)

        // socionets matches sorted
        Title "Sorted Grid Matches for "+tit
        WriteLinks(SocioGetA2I("Socio"),matches,ID,link)

        // mode items or means
        select case par
            case "E" // same elements -- mode
                Title "Sorted mode constructs based on "+tit
                WriteModeSort(SocioGetA2I("Mode"),SocioGetAD("Mode"),GN,ID,par)
            case "C" // same constructs -- mode
                Title "Sorted mode elements based on "+tit
                WriteModeSort(SocioGetA2I("Mode"),SocioGetAD("Mode"),GN,ID,par)
            case "X" // both -- mean/SD

```

```

        dim av(-1,-1) As Double=SocioGetA2D("Means"), sd(-1,-1) As Double=SocioGetA2D("SD")
        SocioSet("Means","0") // set up mean grid in order to use E and C names
        Title "Mean ratings based on "+tit
        WriteMatrixD(av)
        Title "Standard deviations based on "+tit
        WriteMatrixD(sd)
        Title "Standard deviation and mean sorted by standard deviation based on "+tit
        WriteSDSort(SocioGetA2I("SDSort"),av,sd)
    end select
end if
wflushln
end sub

// main program
Main

```

7.3 Grid Data script

This script demonstrates how to access a 3D matrix of normalized grid values for a dataset of grids having elements and constructs in common, The rating values are normalized to be in the range -1.0 to +1.0 (Figure 95).

Grid Data									
Grid 1 Project67									
	1	2	3	4	5	6	7	8	
1	-1.00	-1.00	-0.50	-0.50	0.00	1.00	0.00	1.00	stolid - fashion trend
2	-0.25	-1.00	-0.50	0.50	0.75	0.75	1.00	1.00	limited - versatile
3	-0.25	1.00	-1.00	-0.25	-0.25	-0.50	-0.75	-0.25	high cost - low cost
4	0.25	0.25	-1.00	0.50	0.50	0.00	0.00	1.00	not so good value - good value
5	0.50	-1.00	0.00	1.00	1.00	1.00	1.00	1.00	weakly marketed - heavily marketed
6	-0.25	-1.00	1.00	0.00	0.00	0.00	0.25	0.25	minimal functions - wide range of functions
7	0.00	-0.75	0.75	0.00	0.00	1.00	0.25	-1.00	lower inventory costs - higher inventory costs
8	-0.50	-1.00	0.50	-0.25	0.00	1.00	0.00	1.00	standard features - innovative features
9	-1.00	-1.00	-1.00	-1.00	0.50	0.75	1.00	-1.00	lower development cost - high development cost
10	0.25	1.00	-1.00	0.25	-0.25	-0.50	-0.50	1.00	complex production - simple production
									ideal new product
									new Starlight + rotator
									new Starlight + colors
									new Starlight
									market lead - Raymark
									existing Starstream
									existing Heatfast
									existing Starlight
Grid 2 Marketing									
	1	2	3	4	5	6	7	8	
1	-0.25	-1.00	0.25	-0.25	0.25	1.00	0.50	0.75	stolid - fashion trend
2	0.25	-1.00	1.00	0.25	1.00	1.00	1.00	1.00	limited - versatile
3	-0.25	1.00	-1.00	0.00	-0.50	-0.75	-0.75	0.00	high cost - low cost

Figure 95: Project67 grid data in normalized form

```

// normalized grid data for grids with both common elements and common constructs

Sub WriteMatrixD(m(,,) As Double, k As Integer)
    dim i,r,c As Integer, rbnd As Integer=UBound(m,2), cbnd As Integer=UBound(m,3)
    dim n As Integer=(cbnd+1)*wFormat("")
    write " "
    for c=0 to cbnd
        writeI(c+1)
    next
    writeln

```

```

write " ---|"
for i=1 to n
    write "-"
next
writeln
for r=0 to rbnd
    write sFormat(r+1,"##0")+ " |"
    for c=0 to cbnd
        writeD(m(k,r,c))
    next
    writeln " "+gGet("C",str(r))
next
write(" ")
for c=0 to cbnd
    writeS("|")
next
writeln
for c=cbnd downto 0
    write(" ")
    for i=c-1 downto 0
        writeS("|")
    next
    writeS("",-1)
    writeln gGet("E",str(c))
next
writeln
End Sub

Sub Main()
    SocioSet("Configure","R") // configure for responses only
    SocioSet("Field","R","Grid Data") // set heading of response field
    call wFormat("-##0.00")
    dim m(-1,-1,-1) As Double=SocioGetA3D("Data")
    dim i As integer, bnd As integer=UBound(m,1), sel() As Integer=SocioGetAI("Select")
    for i=0 to bnd
        Writeln "Grid "+sGet(SocioGet("Grid",str(sel(i))),1,2)+EOL
        WriteMatrixD(m,i)
    next
    wflushln
end sub

// main program
Main

```

7.4 Grid Overviews script

This script demonstrates how the full content of the grids in a RepGrids database can be accessed as part of the open architecture capabilities of Rep Plus. This enables users to add their own application-specific multiple grid analyses; indeed it is possible to perform all the RepGrids analyses including the graphic output within the scripting language.

```

// Demonstration script that provides lists of the elements and constructs of selected grids

Sub WriteElements()
    dim j As Variant, ne As Integer=gGetI("NE") // number of elements
    writeln " Elements: "+gGet("Es") // plural for element
    for j=0 to ne-1
        writeln sFormat(j+1,"###0")+": "+gGet("E",j) // element name
    next
    wflushln
End Sub

```

```

Sub WriteConstructs()
    dim i As Variant, nc As Integer=gGetI("NC") // number of constructs
    writeln "   Constructs: "+gGet("Cs") // plural for construct
    for i=0 to nc-1
        writeln sFormat(i+1,"####0")+": "+gGet("C",i) // construct ID
    next
    wflushln
End Sub

Sub WriteGrids()
    dim n As Variant, i As Integer, sel() As Integer=SocioGetAI("Select"), bnd As Integer=sel.
        UBound // get list of selected grid numbers
    for i=0 to bnd
        n=sel(i) // get the number of a selected grid
        SocioSet("Grid",n) // make the grid active (data available through gGet functions)
        writeln n+": "+SGet(gGet("File"),".",1)+" "+gGet("Identifier")+" "+gGet("Date")+" "+gGet("
            Time") // get the grid file name, identifier, date and time
        writeln
        WriteElements
        WriteConstructs
        writeln -----"
    next
    wflushln
End Sub

Sub Main()
    SocioSet("Configure","R") // configure for responses only
    SocioSet("Field","R","Synopsis of Selected Grids") // set the heading of the response field
    WriteGrids
end sub

// main program
Main

```

7.5 Bulk Editing Grids

This script demonstrates how bulk edits may be effected on all selected grids in a RepGrids dataset. A typical usage is to replace element and construct pole names in one language with translations in another language.

It is a risky process that needs to be used with care because it actually modifies the original grids and saves them back to their own files in changed form. *It is not undoable.*

Good practice is to duplicate the collection of grids and load it into a new RepGrids file (it is not good practice to copy the original RepGrids file because the file pointers are designed to be used to search for the original files and may find them rather than the copies).

It is also good practice to test the script being developed on one grid and check that it has edited it as intended, both by loading and viewing it in RepGrid, and by opening the original and edited versions in a text editor to compare them.

```

// This script is intended as an example of how the grids in a RepGrids dataset may be bulk-
    edited
// Typical usage is replace the construct pole names and the elemnt names from one language to
    another

// It edits the grids themselves and hence should be used with great care

```

```

// Make a copy of the collection of grids and create a new RepGrids file for that copy to ensure
// that your RepGrids file points to the grids you copied
// It is good practice to open the RepGrids file in a text editor to check the pointers

// Experiment with one grid and check you have achieved what you wish before running it on all
// Recollect that grids can be examined in a text editor as well as loaded in RepGrid - check the
// text as well as the grid and compare the changes are those you required

// new element and pole names
const ELE="E1|E2|E3|E4|E5|E6" // new element names in some readily accessible format
const LHP="L1|L2|L3|L4|L5|L6"
const RHP="R1|R2|R3|R4|R5|R6"

Sub EditData()
    dim n As Variant, k As Integer, sel() As Integer=SocioGetAI("Select"), bnd As Integer=sel.
        UBound // get list of selected grid numbers
    'dim range As string="1"+TAB+"9" // used if one want to change the rating scale ranges of
        constructs
    dim s As string
    for k=0 to bnd
        n=sel(k) // get the number of a selected grid
        SocioSet("Grid",n) // setup the accessible grid to be the n'th grid
        writeln gGet("Name")+ " "+gGet("File") // will write to "Response" field
        's=NthField(gGet("File"),".",1) // get the grid file name without the extension
        's=NthField(s,"-",2)
        'gSet("Name",s) // set the name field of the grid
        'gSet("Note","")
        'gSet("Context","new context")
        'gSet("E","new element name")
        'gSet("Es","new elements name")
        'gSet("C","new construct name")
        'gSet("Cs","new constructs name")
        dim i,j As variant, bndc As integer=gGetI("NC")-1, bnde As integer=gGetI("NE")-1
        for i=0 to bndc
            Call gGet("C",i,"") // get the construct specification
            hSet(NthField(LHP,"|",i+1)+TAB+NthField(RHP,"|",i+1),"Labels") // edit the pole names
            gSet("C",i,"") // reset the construct with the new pole names
        next
        for j=0 to bnde
            Call gGet("E",j,"") // get the element specification
            hSet(NthField(ELE,"|",j+1),"Name") // edit the element name
            gSet("E",j,"") // reset the element with the new name
        next
        'gSet("Save") // this command will resave the edited grid back to its file adter editing --
            use with care
    next
End Sub

Sub Main()
    SocioSet("Field","Q","Edit Grid") // set the heading of the query field
    SocioSet("Field","R","Edit Grid") // set the heading of the response field
    SocioSet("Field","QT","") // clear query field
    SocioSet("Field","RT","") // clear response field
    EditData // call editing routine
    wflushln
end sub

// main program
Main

```


8 RepGrids File Organization

The function of the RepGrids tool is to manage and process collections of grids. The grid data is held in the usual RepGrid files and a RepGrids file contains pointers to these rather than directly incorporating the data in them. The RepGrids file also records the parameters entered for the various analyses and the grouping structure created by any content analysis. In addition, RepGrids needs to track the location of a default grid and a default net holding the style information for any grids and nets resulting from analyses, and also the scripts managed and activated from the *Scripts* pane.

8.1 RepGrids file pointers

Grid files may be added to a RepGrids database by the following actions:

- Drag a directory to the Rep Plus application icon;
- Drag a directory to the Rep Plus Manager window;
- Drag a grid file or a directory to a RepGrids window;
- Click the *Open Grid* button in the *Grids* pane of a RepGrids window to select a grid file;
- Click the *Open Grids* button in the *Grids* pane of a RepGrids window to select a directory.

Only grid files that are not already in the database will be added. The first two actions create a new RepGrids window. If a directory is added then all the grid files in that directory and its sub-directories to any depth are identified and added as if they had been individually added. The name of the first directory added will be put in the *Title* field of the *Grids* pane if it is empty.

RepGrids uses the same grid file identifier and decoder as RepGrid so that grid data in a wide range of formats is accepted (including WebGrid pages saved as HTML). Note that files containing a single grid are required so that the spreadsheet format, or pasted/dragged text encoding a grid, as recognized by RepGrid, will not be added to RepGrids unless it is first loaded in RepGrid and saved as a file.

Regardless of how grids are added to a RepGrids database, they are all recorded in the RepGrids file through file pointer structures (Figure 96). These structures contain references both the absolute location of the grid file and its location relative to that of the RepGrids file as well as additional information about the file that might be used to identify it even if its location were changed. This allows, for example the integrity of the database to be maintained even if the contents of a grid file are edited, the RepGrids File and its grids are relocated without changing their relative locations, or, less securely, if grid files are moved to nearby locations.

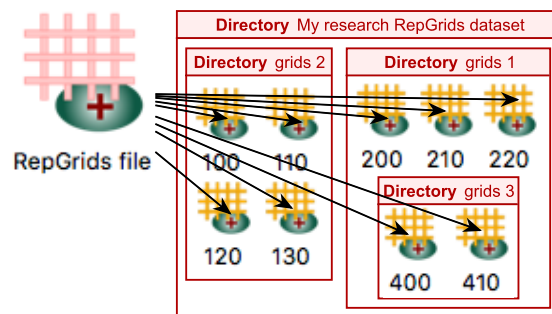


Figure 96: Pointers to grid files in a RepGrids file

In principle the grid data files could be scattered anywhere accessible through the computer's file system. In practice they are a major resource that need to be managed carefully and it is usual to keep them in a single directory, perhaps with sub-directories as illustrated in Figure 96. The RepGrids file needs to be in a location that is well-defined relative to the grid file directories, such as in the same directory as the top level grid file directory or even inside that directory. Both options make it easy to copy the files elsewhere while maintaining the relative location references.

However, there are no rigid requirements and users can set up whatever arrangements suit the way they are managing the data from their study. Note that, because RepGrids recognizes grid files by the structure of their contents, files from other applications may be included with the grid files such as research notes, photographs, statistical data, and so on.

8.2 RepGrids scripts and default grids and nets—GridsScripts directories

RepGrids scripts are located in *GridsScripts* directories which may also contain style grids and nets. The scripts file directories for RepGrids have the same network structure as those for RepGrid and RepNet, that is with an application-supplied GridsScripts directory in the application directory, a user GridsScripts directory in the Documents or MyDocuments directory, and a data-specific GridsScripts directory in the same directory as the RepGrids database file or a surrounding directory.

Figure 97 shows the directory structure corresponding to the style grid and net menus of Figure 58. The files at level 3 appear in the top partition of the menu. If they are used for several RepGrids databases they could be in the surrounding directory at level 2. Those in the documents directory at level 1 appear in the central partition, and those in the application directory at level 0 in the lowest partition.

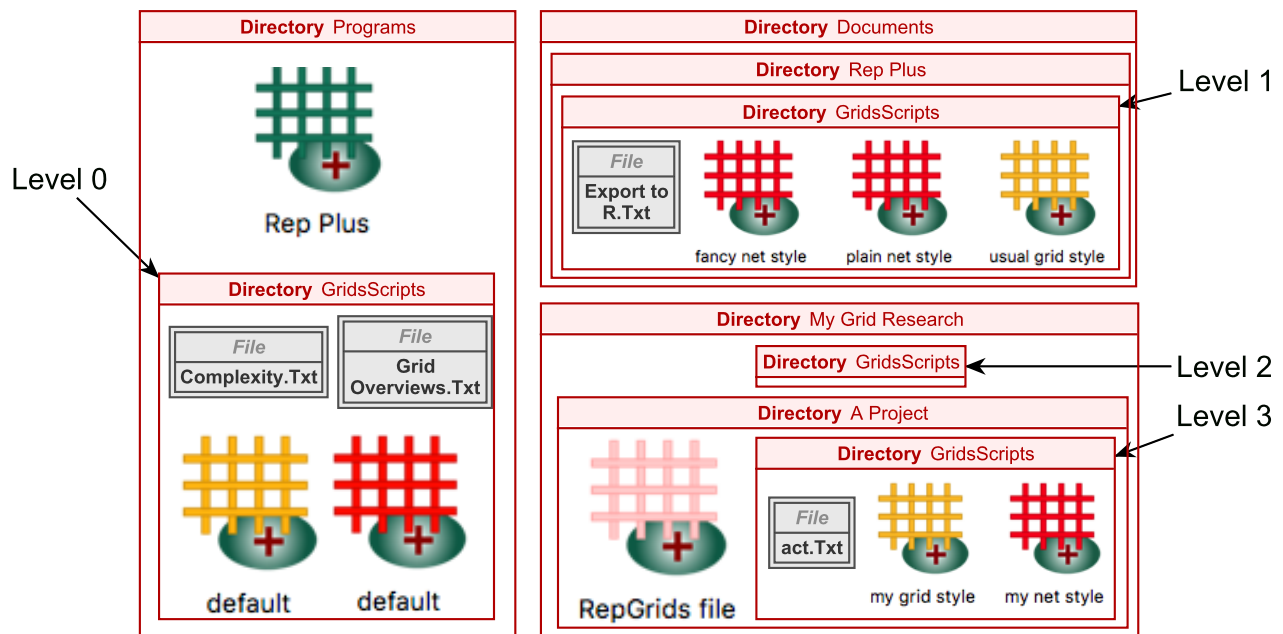


Figure 97: Search paths for GridsScripts directories containing scripts and default grids/nets

For many uses of RepGrids a single directory for each collection of grid data and the default scripts and styles will be totally adequate. The flexibility of organizing the data in the ways shown in Figures 96 and 97 will not be needed. However, in large scale projects involving a team of people over a long time period, it can be useful to organize the data and associated tools in such a way that it provides a form of *corporate memory* of what was used where, not only in data analysis but also in presentations and publications.

The Rep Plus organizational options provide the flexibility to use the operating systems' hierarchical filing system to manage all the notes, data, tools, analyses, presentations and publications in such large projects in a way that allows current and future studies to evolve in a manageable fashion.

8.2.1 Structure of default grids and nets

The grid selected as the default is used to provide the styles for those RepGrids analyses that generate a grid such as a composite grid or mode grid. It can be any grid having the required styles since these are the only information used. The analysis parameters in the generated grids are set to the normal RepGrid defaults except that the notes are set to be shown since, for the non common items, they contain the identifier of the grid from which an element or construct was extracted.

The grid selected as the default is used to provide the styles for those RepGrids analyses that generate a net such as a socionet or histogram. It needs to have the following ten node styles defined:-

1. **Link Label:** style for the optional labels showing the link values in socionets —this must be the first node style as it is identified by position rather than name;

2. **Cut off:** style for the node showing the cut off for link values in socionets —this must be the second node style as it is identified by position rather than name—its type is also usually shown and can be different from “Cut off”, e.g. “Threshold”;
3. **Title:** style for the node showing showing the grids in socionets—this, and the remaining node and link labels, is identified by name and can be in any position other than first or second;
4. **Grid:** style for the nodes showing the title in all RepGrids plots;
5. **Scale:** the label style for this node is used for the scales and scale point values in histograms;
6. **Construct:** style for the nodes showing the construct poles in histograms;
7. **Element:** style for the nodes showing the element names in histograms;
8. **Distribution:** style for the nodes showing the distribution names in histograms—the label colour is also used for the distribution bars;
9. **Distribution+:** style for the nodes showing the comparative distribution data in histograms—the label colour is also used for the comparative distribution bars;
10. **Statistics:** style for the nodes showing the statistics of distribution comparisons.

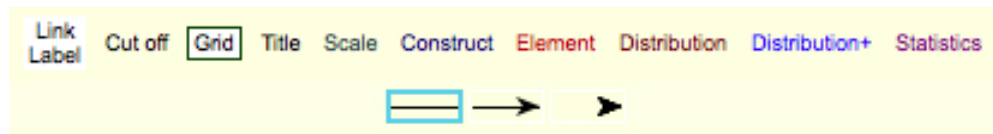


Figure 98: Node and link types in RepGrid default net

It needs to have the following three link styles defined:-

1. **Line:** style for the symmetric links between grids in socionets for grid data analyzed as having both elements and constructs in common;
2. **Arrow:** style for the stronger asymmetric links between grids in socionets for grid data analyzed as having either elements or constructs in common;
3. **Arrow-:** style for the weaker reverse links between grids—this is usually just an arrow head since the line has been plotted as part of the stronger link.

9 Appendix 1: Data Sets Used in the RepGrids Manual

One of the most difficult tasks in writing the RepGrids manual has been to find appropriate data sets to illustrate the tools in action that can be made publicly available so that users may explore the tools with meaningful data. The RepGrids tool were developed to support the analysis of the large sets of conceptual grids that are nowadays being used to study the conceptual structures of communities. Such large-scale studies are generally funded by organizations addressing particular issues that are sensitive and confidential to those organizations. Whilst we have been fortunate in being allowed access to such data in order to develop and test the RepGrids tools, we have used it on a confidential basis and agreed not to share it outside our research group.

We do have some datasets from our own studies where we have secured agreement that, subject to suitable anonymization, the data may be shared, and have conducted further studies with recreational communities specifically targeted on generating datasets where, again, there is no problem with sharing the data provided there is no possibility of identifying the individuals involved. We have also secured permission to use some data in the manual for illustrative purposes provided the entire dataset is not issued.

The following datasets are available in the *Rep Plus Support* directory and enable users to reproduce the analyses in this manual and explore the use of all the RepGrids tools.

9.1 Study of a research group specializing in geographic mapping techniques

The *Geog* data originated in a comparative and longitudinal study of a small research group studying geographic mapping techniques (Shaw and Gaines, 1989). The domain was a scientific one where it might be expected that many of the significant constructs would be *public* but it was also subject to intense ongoing collaborative research that might result in changes over time.

The aim was to determine whether conceptual grid elicitation and analysis would be of value to the researchers in developing their models of the domain, and also to test the stability of the construct systems over time, tracking changes in individual models and in the relationships between those in the research group.

Further details of the study and its methods are available in Shaw and Gaines (1989).

9.2 Study of a ballroom dance community

The *Dance-E* data is from ballroom dance study that was undertaken in order to provide a practical example of WebGrid supporting a research method which we termed *sociocognitive inquiry* (Gaines and Shaw, 2012), designed to elicit conceptual models from a networked community. We chose a recreational topic with a readily accessible community where the outcomes would be readily understandable, either through personal experience, or through the widely available instructional literature.

The *Dance-EC* data is from an extension of the ballroom dance study to a larger community where we used the consensual elements and constructs from the first study in an *exchange grid* and asked a larger community of dancers to rate the elements on the constructs, allowing them to add

additional items if they wished. The focus group had 10 members and the community group 77 members.

This research method has been used in several business studies (Öhman et al., 2006; Bellman, 2012; Bellman et al., 2016; Rad et al., 2013) where: a focus group agrees on a domain and purpose; selects a set of elements within the domain representative of that purpose; constructs are elicited from the focus group that distinguish those elements; a mode grid analysis is used to derive the consensual constructs; and members of the community under study are asked to rate the selected elements on the focus group's consensual constructs, adding additional personal elements and constructs as they see fit; the community is stratified along dimensions representing the objectives of the study; and RepGrids tools are used to analyze the data collected.

9.3 Study of product development in a manufacturing company

The *Project67* data is from a study of the conceptual models of senior management at a small appliance manufacturing company concerned to develop new products to address an erosion in market share. The original data was proprietary but we were allowed to create an available set in which the domain, the element names and some of the construct pole names have been changed to anonymize the data whilst preserving the types of issue that arose in the analyses.

The study was also used to experiment with the use of networked personal computers to support communication in the group Shaw and Gaines (1991) in an era when the World Wide Web had yet to evolve.

9.4 Study of knowledge engineering for expert systems

The *BIAIT* data is from an early study of the applicability of conceptual grid elicitation and analysis to the development of *expert systems* (Shaw and Gaines, 1983). We had suggested this usage in an earlier publication (Gaines and Shaw, 1980) which had aroused industrial interest, and undertook a *proof of concept* study to determine whether the conceptual structure underlying a well-specified business process analysis technique (Carlson, 1979; Leshob et al., 2015) could be reconstructed from grids elicited from individuals having varying degrees of knowledge of the domain but not of the technique.

10 Bibliography

- Aggarwal, C. C. (2013). An introduction to outlier analysis. In *Outlier Analysis*, pages 1–40. Springer, New York, NY.
- Agresti, A. (2010). *Analysis of Ordinal Categorical Data*. Wiley, New York.
- Barcus, F. E. (1960). *Theory and Methodology of Content Analysis*. PhD thesis.
- Bellman, L. (2012). *Auktoriserade fastighetsvärderares syn på värdering: tankemönster om kommersiella fastigheter*. Licentiate thesis.
- Bellman, L., Lind, H., and Öhman, P. (2016). How does education from a high-status university affect professional property appraisers’ valuation judgments? *Journal of Real Estate Practice and Education*, 19(2):99–124.
- Birnbaum, Z. W. (1956). On a use of the Mann-Whitney statistic. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 13–17. University of California Press, Berkeley, LA.
- Carlson, W. M. (1979). Business information analysis and integration technique (biait): the new horizon. *ACM SIGMIS Database*, 10(4):3–9.
- Cliff, N. (1993). Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114(3):494–509.
- Cliff, N. (2014). *Ordinal Methods for Behavioral Data Analysis*. Psychology Press.
- Cumming, G. (2012). *Understanding the New Statistics: Effect Sizes, Confidence Intervals, and Meta-analysis*. Routledge, New York.
- Fitzgerald, M. (2012). *Introducing Regular Expressions*. O’Reilly, Sebastopol, CA.
- Gaines, B. R. and Shaw, M. L. G. (1980). New directions in the analysis and interactive elicitation of personal construct systems. *International Journal of Man-Machine Studies*, 13(1):81–116.
- Gaines, B. R. and Shaw, M. L. G. (2012). Sociocognitive inquiry. In Ting, I.-H., Hong, T.-P., and Liang, L. S., editors, *Social Network Mining, Analysis and Research Trends: Techniques and Applications*, pages 35–55. IGI Global, PA, USA.
- Gower, J. C. and Dijksterhuis, G. B. (2004). *Procrustes Problems*. Oxford University Press, Oxford.
- Hedges, L. V. and Olkin, I. (1985). *Statistical Methods for Meta-analysis*. Academic Press, Orlando.
- Honey, P. (1979). The repertory grid in action: how to use it to conduct an attitude survey. *Industrial and Commercial Training*, 11(11):452–459.

- Howe, J. (2008). *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business*. Crown Business, New York, 1st edition.
- Jankowicz, D. (2004). *The Easy Guide to Repertory Grids*. Wiley, Chichester, UK.
- Kenney, J. F. and Keeping, E. S. (1951). *Mathematics of Statistics 2*. Van Nostrand, Princeton, NJ.
- Krippendorff, K. (2013). *Content Analysis: An Introduction to Its Methodology*. Sage, Thousand Oaks, CA.
- Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and anovas. *Frontiers in psychology*, 4.
- Leshob, A., Mili, H., and Boubaker, A. (2015). Ontology-driven process specialization. In Beny-oucef, M., Weiss, M., and Mili, H., editors, *E-Technologies: 6th International Conference, MCETECH 2015*, pages 3–19. Springer, Cham.
- Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *Annals Mathematical Statistics*, 18(1):50–60.
- Mood, A. M. (1954). On the asymptotic efficiency of certain nonparametric two-sample tests. *The Annals of Mathematical Statistics*, 25(3):514–522.
- Neuendorf, K. A. (2002). *The Content Analysis Guidebook*. Sage, Thousand Oaks, CA.
- Öhman, P., Häckner, E., Jansson, A.-M., and Tschudi, F. (2006). Swedish auditors' view of auditing: Doing things right versus doing the right things. *European Accounting Review*, 15(1):89–114.
- Rad, A., Wahlberg, O., and Öhman, P. (2013). How lending officers construe assessments of small and medium-sized enterprise loan applications: a repertory grid study. *Journal of Constructivist Psychology*, 26(4):262–279.
- Richardson, P. J. S. (1946). *A History of English Ballroom Dancing, 1910-45. The story of the development of the modern English style*. Herbert Jenkins, London.
- Shaw, M. L. G. (1979). Conversational heuristics for eliciting shared understanding. *International Journal of Man-Machine Studies*, 11:621–634.
- Shaw, M. L. G. (1980). *On Becoming a Personal Scientist: Interactive Computer Elicitation of Personal Models of the World*. Academic Press, London.
- Shaw, M. L. G. and Gaines, B. R. (1983). A computer aid to knowledge engineering. In *Proceedings of British Computer Society Conference on Expert Systems*, pages 263–271. British Computer Society, Cambridge.
- Shaw, M. L. G. and Gaines, B. R. (1989). Comparing conceptual structures: consensus, conflict, correspondence and contrast. *Knowledge Acquisition*, 1(4):341–363.

- Shaw, M. L. G. and Gaines, B. R. (1991). Extending electronic mail with conceptual modeling to provide group decision support. In *COCS'91: Proceedings of Conference on Organizational Computing Systems*, pages 153–158. ACM Press, New York.
- Tukey, J. W. (1962). The future of data analysis. *The Annals of Mathematical Statistics*, 33(1):1–67.
- Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley series in behavioral science. Addison-Wesley Pub. Co., Reading, MA.
- Tukey, J. W. (1980). We need both exploratory and confirmatory. *The American Statistician*, 34(1):23–25.
- Wenger, E. (1998). *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, Cambridge.
- West, M. D. (2001a). *Applications of Computer Content Analysis*. Progress in communication sciences,. Ablex, Westport, CT.
- West, M. D. (2001b). *Theory, Method, and Practice in Computer Content Analysis*. Progress in communication sciences,. Ablex, Westport, CT.
- Wilcox, R. R. (1998). How many discoveries have been lost by ignoring modern statistical methods? *American Psychologist*, 53(3):300–314.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.
- Wittgenstein, L. (1953). *Philosophische Untersuchungen: Philosophical investigations* (trans. G.E.M. Anscombe). Blackwell, Oxford.

Most of our publications cited in the references are freely available on the web