Orbital Decomposition

A short user's guide to ORBDE v2.4

What is Orbital Decomposition?

ORBDE is an analysis based on symbolic dynamics that identifies recurring patterns of events in nominally-coded time series data. The events are identified as qualitative categories, rather than metrics. The patterns are potentially chaotic, but not necessarily so. The analysis finds the optimal length of the patterns (string length, *C*), and the recurring patterns at that string length. The search for optimal string length is based on whether any patterns proximally recur, i.e., one example appears immediately after the other. Once the length has been determined, the patterns that appear at least twice – proximally or distally – are identified. The final statistics include the string length, topological entropy in the form of the (largest) Lyapunov exponent, fractal dimension, Shannon entropy, and a chi-square test for goodness of fit.

Source material for theoretical background on ORBDE appear at the end of this document (Guastello, 2000, 2011; Guastello, Hyde, & Odak, 1998; Guastello, Peressini, & Bond, 2011). Some interesting applications have included family and group dynamics (Pincus, 2001; Pincus & Guastello, 2005; Pincus, Ortega, & Metten, 2011), political and domestic violence and self-injurious behavior (Katerndahl et al., 2010; Pincus et al., in press; Spohn, 2008), an analysis of medical interviews (Katerndahl & Parchman, 2010), neuromotor activation patterns (Nathan, Guastello, Prost, & Jeutter, 2012), and task-switching, multitasking, and work performance (Guastello et al., 2012, 2013).

New for Version 2.4

The standard analysis performed by v1.2 started with a series of events that all have unique codes, and only one code is assigned to each event. The standard analysis is still the default mode in v2.4, but now you have the options of assigning more than one code from the same set or assigning codes from multiple sets simultaneously by using aggregation commands. The instructions thus start with the original simple case and then proceed to the new features.

Preparing to Download

Instructions for downloading are different for PC compared to MAC. The PC instructions are presented first and followed by the instructions for MAC.

PC Instructions

Before actually downloading the program file ORBDE.EXE prepare the directory where you want it to land and where you will actually be using it. These instructions are meant for PC Windows.

1. Go to **`Start | accessories | command prompt`**. Note the directory name just before the command prompt itself. One of our computers says: **`C:\documentsandsettings\HP_Administrator>`**

2. Using Windows Explorer, go to the folder shown in the command prompt above. Make a new folder, and call it **`ORBITAL`**. This is the folder into which you will download the program. It will also be where you will run any executions. Any data sets that you make to run with ORBDE would go in this folder also.

3. Download **`either ORBDE.EXE`** or **`ORBDE.ZIP`** into your **`ORBITAL`** folder. Different computer security systems repond differently to an .EXE file, so we have two formats available. If you download the .ZIP version,

you will need to unpack the file **`ORBDE.EXE`** from it.

4. Download the test data sets **`data1.txt`**, **`evendemo.txt`** and **`ragdemo.txt`**.

MAC Instructions

Before actually downloading the program file **`orbde`** (which is a Unix executable) prepare the folder where you want it to land and where you will actually be using it. These instructions are meant for Mac OS X.

1. Open a console window by clicking on **`Console`** from in your **`Applications | Utilities`** folder. At this point the current directory in the console window will be the current user's home folder. It may be show in the prompt which might say something like **`macpro:~ tony`** which indicates the home directory on the machine named MacPro is called "tony."

2. Next you will create an ORBDE working folder in one of the follwoing two ways. Option 1: use **`Finder`** to navigate to your home folder (in our example "Tony"), which usually has a shortcut to it on the side panel of the **`Finder`** window, and then click on the **`File | New Folder`** option at the top of the **`Finder`** window and create a new folder, and call it **`orbital`**. Option 2: From within the console window type at the command prompt **`mkdir orbital`** noting that there is a space between "mkdir" and "orbital". This will create a subfolder called "orbital" within your home folder. This is the folder into which you will download the ORBDE program. It will also be where you will run it. Any data sets that you make to run with ORBDE and its output would go in this folder also.

3. Download either **`orbde`** or **`orbde.zip`** into your **`orbital`** folder. Different computer security systems repond differently to an executable file, so we have two formats available. If you download the .ZIP version, you will need to unpack the executable file **`orbde`** from it.

4. Download the test data sets **`data1.txt`**, **`evendemo.txt`** and **`ragdemo.txt`**.

Run the Program

Once again we start with the PC instructions, which are followed by modifications for MAC. The PC instructions begin with the standard mode, which are followed by instructions for using the new aggregation commands.

PC Instructions

Standard Mode

5. Go back to #1. Enter **`cd orbital`** at the command prompt. The directory will change to your new orbital directory.

6. To the run the program itself, enter the following command at the prompt:

```
orbde_ data1.txt_ -p_5_ -cnz_ > newout.txt
```

The subcommand **`data1.txt`** identifies the data set that you want analyzed. The subcommand **`-p_5`** produces preliminary calculations for string lengths up to the number given, 5 in this case. You can change this number. The gray underbar (_) is a required space. Do not type the underbar itself. The hyphen (-) is a required keystroke, however.

The subcommand **`-cnz`** produces the final statistics for string lengths where the trace of the solution matrix does not equal zero. This is a recommended default. If you want to change it, the syntax is **`-c_#`** where **`#`** is the string length of choice. This version of the subcommand will produce final statistics up to and including the string length of the size you specify in # (change # to a numeral). Although **`-p`** and **`-c`** are not required to run the program, neglecting them could produce voluminous output that could be, uh, not worth keeping.

The pipe **>** means put it in an output file instead of printing to the screen, and **`newout.txt`** is a sample name for an output file. The data and output files are in ASCII, and the extension **`.txt`** in the files names is required. When the computation is finished and the output file is created, ORBDE returns the DOS prompt.

Aggregation Mode

When ORBDE operates in the standard mode, it reads each *keystroke* as an event. When it operates in aggregation mode, it reads each *line* of data as an event. In the first part of the aggregation process ORBDE tallies the lines of code patterns and renames each one as a single letter. The standard ORBDE routine is then applied to the new set of single letters.

There are two types of aggregation modes in ORBDE. The first, for *ragged lines*, allows you to apply multiple codes from one set of codes to an event. To execute ORBDE using this option, use the command `-Ar` in the following syntax:

```
orbde_ data1.txt_ -p_5_ -cnz_ -Ar > newout.txt
```

The second mode, even lines, allows you to apply codes from multiple variables, each of which contains multiple categories. ORBDE then expects to read the same number of characters for each line of data. To execute ORBDE using this option, use the command `-Ae` in the following syntax:

```
orbde_ data1.txt_ -p_5_ -cnz_ -Ae > newout.txt
```

A third subcommand allows you to check the possible aggregation codes without running the analysis. This option might be used by researchers who want to probe their coding systems and perhaps make some adjustments. To use this option for aggregate only, type `-Ao` in the syntax above in place of `-Ar` or `-Ae.`

A fourth subcommand for aggregation that can be used optionally with either `-Ar` or `-Ae`. The first step in the aggregation process could conceivably produce a large number of codes that only appear once, and each of them is given its own letter code for further analysis. Some researchers might prefer to lump all the single-appearances into one code, e.g. "none of the above." The command `-Az` will produce that result, and it can be executed using the following syntax:

```
orbde_ data1.txt_ -p_5_ -cnz_ -Ae -Az > newout.txt
```

MAC Instructions

The execution command is `./orbde_` . All other subcommands and syntax are the same as for PC. Once again the data and output files are in ASCII text and so the extension `.txt` is helpful in keeping things straight and also ensure that the proper program (a text editor) will automatically open them when you click on them from a `Finder` window. When the computation is finished and the output file is created, ORBDE returns the command prompt.

Making Data Sets

Instructions for the PC version are described first and are followed by modifications for MAC units.

PC Instructions

Standard Mode

ORBDE processes data files that are made in ASCII only. You can make ASCII files using the Windows Notepad program. File names will end with the `.txt` extension automatically. If you prefer the convenience of a spreadsheet, you can copy and paste your spreadsheet contents into a Notepad file. Be sure to eliminate tabs, however, from the Notepad file; ORBDE will return an error message if it encounters tabs.

Some editions of Windows Explorer hide the extensions to file names, so, depending on your version of the software, the ASCII data file will appear in your Windows Explorer directory as the name with or without the extension, e.g. `data1.` ORBDE wants to read `data1.txt` and will return an error message if it does not find it. You can type `dir` at the DOS command prompt and see what file names and extensions are really there and rename files as necessary.

Data Contents

The format for the data itself consists of codes of single keystrokes that denote categories or types of events. You decide what the codes mean. ORBDE accepts a total of 52 possible codes that are designated as letters of the Roman alphabet A to Z upper case and lower case. In other words, the data reader is case sensitive. A set of codes is interpreted by ORBDE as representing exclusive categories.

Type the sequence of events on a line. Lines can be of any length (up to 80 characters), and they can vary in length. ORBDE reads lines as given and then moves to the next line of type. In other words, it reads across before it reads down. It conveniently ignores hard returns between lines, and you can copy|paste a column from a spreadsheet (lines of one character), which is a common set-up for time series data.

Data File Size

The program is generally friendly to data files of 1000 events before computational overflow sets in. It can actually process data sets of up to 2400 strokes, depending on the values of **`-p`** and **`-c`** that you set. In the event of an overflow, you will get the DOS prompt back, and the output file will give an error message. Expect a trade-off between the number of codes that are active in a set and the length of the data set to produce reasonable computation times and to prevent overflow.

Missing data

ORBDE will treat blanks in your data series as if they did not exist and will attempt to determine string lengths and so forth by joining the last keystroke to the next available keystroke until it runs out of keystrokes. There could be research problems, however, where the missing entries are dependent on other events in the research. In those cases you probably want to code missing entries as another alphabetical category.

Aggregation Mode

If you are using the aggregation mode that expects ragged lines (subcommand **`-Ar`**), you can apply one or more codes to an event, as you prefer. For instance, if you were coding utterances from a conversation, one utterance might be given only one code, such as “asking a question.” Another event, however, could be coded for two or more criteria, such as “asking a question,” “cracking a joke,” and “offering a new idea.” You can apply up to 52 codes to an event, which is the limit of the possible codes associated with one variable.

If you are using the aggregation mode that expects even lines (subcommand **`-Ae`**), you can apply codes from multiple variables, each of which contains multiple categories. For instance, if the application involved political violence, there could be a code for the nature of the event that was perpetrated, a code for the reaction by the other party, and a code for public opinion immediately afterwards. Each variable is limited to 52 codes as in the standard mode.

Research management tip #1: Researchers often have many more variables in their data bases than they care to use for any one particular analysis. It would probably be a worksaver to use a spreadsheet such as EXCEL as the master file for storing all coded variables. Then, for a particular analysis that uses only a subset of the variables, do SAVE AS **`newfilename.xls`**, delete unwated columns, SAVE. The result can then be exported to an ASCII file for ORBDE analysis.

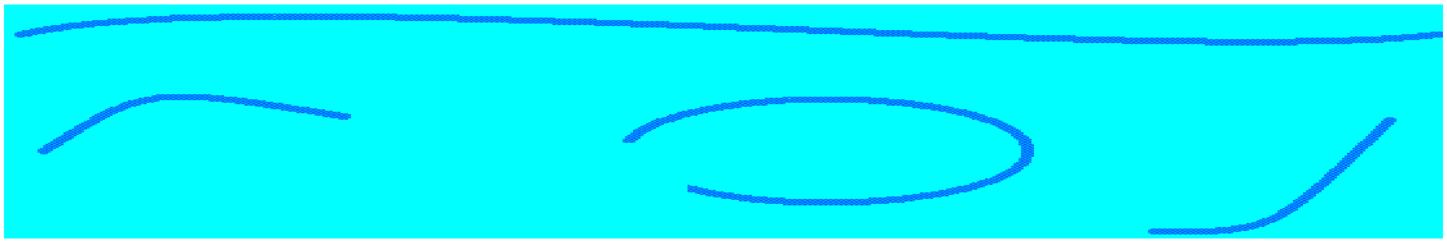
Reminder: ORBDE reads entire lines of data as an event in aggregation mode, so don’t forget to put in the line breaks.

MAC Instructions

ORBDE processes data files that are made in ASCII only. You can make ASCII files using the Mac OS X TextEdit program. File names will end with the **`.txt`** extension automatically. If you prefer the convenience of a spreadsheet, you can copy and paste your spreadsheet contents into a TextEdit file. Be sure to eliminate tabs, however, from the TextEdit file; ORDBE will return an error message if it encounters tabs.

Some versions of OS X hide the extensions to file names, so, depending on your version of the software, the ASCII data file will appear in your **`Finder`** window as the name with or without the extension, e.g. **`data1.`** ORBDE wants to read **`data1.txt`** and will return an error message if it does not find it. You can type **ls** at the console command prompt and see what file names and extensions are really there and rename files as necessary.

The MAC instructions for data content, data file size, missing data, and aggregation mode are the same as those for the PC version above.



Output

Standard Mode

Your output lands in the ORBITAL folder that you made in Step 2. The file name will be the one you gave it on the command in Step 6 (**`newout.txt`** in the example). A portion of the output for a test data set that is produced by (or limited by) the **`-p_2`** subcommand appears below.

```
The list length is 60, and the number of distinct values is 5

--------------------------------------------------------------------------
                         Partial Statistics
 Code  (C=1)          rep?  Freq   Ex Freq    Pr-Obs    Shan      Sq Dev
--------------------------------------------------------------------------
B                      y     18    12.000     0.300     0.361     7.298
A                      -     12    12.000     0.200     0.322     0.000
D                      y     12    12.000     0.200     0.322     0.000
C                      -      9    12.000     0.150     0.285    -2.589
E                      -      9    12.000     0.150     0.285    -2.589
--------------------------------------------------------------------------
                         Partial Statistics
 Code  (C=2)          rep?  Freq   Ex Freq    Pr-Obs    Shan      Sq Dev
--------------------------------------------------------------------------
BB                     -      9     5.310     0.153     0.287     4.749
BC                     -      9     2.655     0.153     0.287    10.987
EA                     y      9     1.770     0.153     0.287    14.636
DD                     -      6     2.360     0.102     0.232     5.599
DE                     -      6     1.770     0.102     0.232     7.325
AB                     -      6     3.540     0.102     0.232     3.166
CD                     -      3     1.770     0.051     0.151     1.583
CB                     -      3     2.655     0.051     0.151     0.367
AE                     -      3     1.770     0.051     0.151     1.583
AD                     -      3     2.360     0.051     0.151     0.720
CA                     -      2     1.770     0.034     0.115     0.244
--------------------------------------------------------------------------
```

The first line of output shows the number of observations and the number of codes in the coding scheme. Separate tables of partial statistics are given for string lengths of 1 and 2 (statistics for longer string lengths are not shown here). The second column **`rep?`** shows **`y`** (= yes) if the code was one that was found to repeat itself proximally; in this case 2 codes proximally repeated and 3 did not. The third column **`Freq`** is the observed frequency of the code. If this were the final solution, you would only interpret the codes with frequencies ≥ 2 (which would be all of them in this example).

The fourth column **`Ex Freq`** is a calculation of expected frequency that will be used in the χ^2 test. The fifth column **`Pr-Obs`** is the observed probability for the event. The sixth column **`Shan`** is a partial calculation of Shannon entropy associated with a particular entry; the completed calculation appears in the output table of final statistics. The final column is a partial calculation of the chi-square test that is associated with a particular code. These partial calculations are included so you can see the relative influence of a particular code on the final statistics.

The second table of partial statistics shows the same information for string lengths of 2. There are 11 duplex codes, of which 1 repeated proximally. Once again, if this were the final solution, you would only interpret the codes with frequencies ≥ 2 (which would be all of them in this example).

The table of final statistics for this data set appears below. The data set was deliberately prepared to contain a repeating string of 20 characters. Thus we see the trace decline as C increases, but it pops up again at $C = 20$. If the string length of 20 had not been implanted, the solution for this data set would be $C = 4$.

The final statistics are: **`trM`** = trace of a square matrix of symbols followed by symbols. The trace counts the number of symbols that are immediately followed by themselves. **`Ht`** = topological entropy, which is, in limiting conditions, equal to the (largest) Lyaponov exponent. **D_L** = fractal dimension based on the Lyapunov exponent, also known as the Lyapunov dimension. **`Chi^2`** is the maximum likelihood χ^2 test that determines the extent to

```
-------------------------------------------------------------------------
                          Final Statistics
-------------------------------------------------------------------------
   C   trM     Ht         Dl       chi^2      df      N*    Phi^2      Hs
-------------------------------------------------------------------------
   1    2     1.000     2.718      4.240      4      60     0.071     1.574
   2    1     0.000     1.000    101.915     11      59     1.727     2.278
   3    1     0.000     1.000    237.429     16      58     4.094     2.676
   4    1     0.000     1.000    387.869     19      57     6.805     2.914
  20   20     0.216     1.241   2336.287     20      41    56.983     2.991
-------------------------------------------------------------------------
```

which of the patterns occurred by chance, based on the observed frequencies of the elements that contribute to the symbol cluster. **`Df`** = degrees of freedom associated with the χ^2 test. **`N*`** is the number of strings of a particular length, which is used to calculate ϕ^2 from χ^2. **`Phi^2`** is an analogue of the familiar correlation-substitute. Unlike the familiar prototype it is not guaranteed to range between 0 and 1. Nonetheless, it is an index of goodness of fit given χ^2 and N^*. **`Hs`** = Shannon entropy.

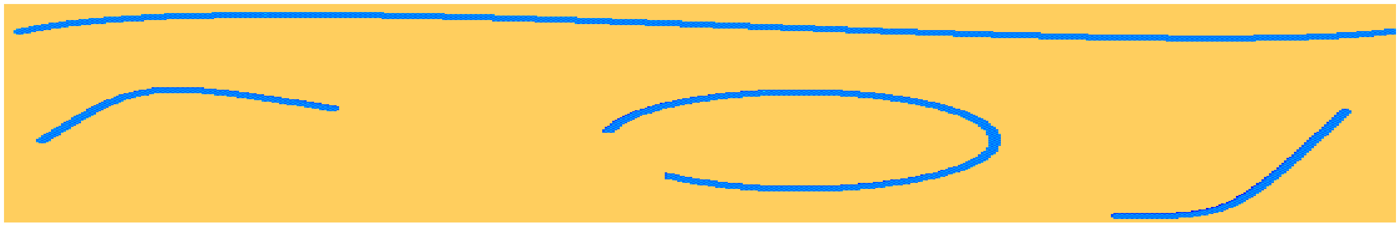
Aggregation Mode with Ragged Lines

In the following demo, there was one variable that had six possible codes, A, B, C, D, E, F. The first part of the output shows the correspondence between the original code combinations and the aggregated codes applied by ORBDE. The "number of distinct lines" refers to the number of unique aggregated codes. The remainder of the analysis carries on as usual using the aggregated codes. In this example the final solution worked out to $C = 4$ using either the minimum *Ht* criterion or the χ^2 criterion. The list of aggregated code sequences is also shown below. There were many sequences that only appeared once, so only the first two are shown here.

```
AGGR-ORBDE: The number of lines is 112 and number of distinct lines is 26.
-------------------------------------------------------------------------
              Frequency Distributions of Code Combinations
           Original                |             Aggregated
         Code Combination           |  Code  |  Number  |    Frequency
-------------------------------------------------------------------------
  DE                                   A          4          0.036
  AE                                   B         17          0.152
  A                                    C          1          0.009
  DF                                   D          2          0.018
  ADE                                  E          6          0.054
  ABC                                  F          2          0.018
  ADEF                                 G          1          0.009
  CE                                   H          5          0.045
  BF                                   I          3          0.027
  CF                                   J          1          0.009
  EF                                   K          2          0.018
  C                                    L          5          0.045
  AD                                   M          3          0.027
  AC                                   N          2          0.018
  BD                                   O         30          0.268
  D                                    P          1          0.009
  BC                                   Q          7          0.063
  B                                    R          7          0.063
  ABD                                  S          2          0.018
  CD                                   T          2          0.018
  DC                                   U          1          0.009
  CB                                   V          1          0.009
  BE                                   W          2          0.018
  BCE                                  X          3          0.027
  CDF                                  Y          1          0.009
  CDE                                  Z          1          0.009
-------------------------------------------------------------------------
```

Research management tip #2: We recommend that the codes in your system be applied in the same (alphabetical) order for each event. ORBDE treats sequences such as AB as different from BA, even though the two are meant to have the same meaning.

```
                           Final Statistics
----------------------------------------------------------------------------
   C    trM      Ht         Dl        chi^2       df      N*      Phi^2       Hs
----------------------------------------------------------------------------
   1    4      2.000      7.389    131.720     25      112      1.176      2.670
   2    3      0.792      2.209     95.570     12      111      0.861      3.649
   3    3      0.528      1.696    141.649     10      110      1.288      4.105
   4    2      0.250      1.284    141.341     10      109      1.297      4.443
----------------------------------------------------------------------------
                          Partial Statistics
----------------------------------------------------------------------------
 Code (C=4)           rep?  Freq   Ex Freq     Pr-Obs     Shan      Sq Dev
----------------------------------------------------------------------------
OOOO                   -     6     0.561      0.055     0.160     14.218
BBBB                   -     3     0.058      0.028     0.099     11.845
QOOO                   y     3     0.131      0.028     0.099      9.395
QBBB                   y     2     0.024      0.018     0.073      8.860
LBBB                   -     2     0.017      0.018     0.073      9.533
OOOS                   -     2     0.037      0.018     0.073      7.958
OOOQ                   -     2     0.131      0.018     0.073      5.453
OOQO                   -     2     0.131      0.018     0.073      5.453
OQOO                   -     2     0.131      0.018     0.073      5.453
RRRR                   -     2     0.002      0.018     0.073     14.184
JKLD                   -     1   107.777      0.009     0.043    -21.682
KLDA                   -     1         -      0.009     0.043          -
-----------------------------------------------------------------------------
```



Aggregated Mode, Even Lines

In the following demo, there were two variables that had six possible codes, A, B, C, D, E, F. The first part of the output, which starts on the next page, shows the correspondence between the original code combinations and the aggregated codes applied by ORBDE. The nuance here is the report of Kolmogorov-Sinai entropy, which is another measure of topological entropy. Its value defaults to Shannon entropy for one- or two-variable problems.

The solution for this analysis was $C = 2$ using the Ht criterion.The list of aggregated code sequences is also shown for $C = 2$. There were many sequences that appeared only once, so only the first is listed in the excerpt from the output.

This example is a case where the χ^2 criterion did not resolve an ambiguity between the values of C in the solution as it was intended to do. Instead it indicated that $C = 1$ could be a better solution, given that there were only two possibilities here. The researcher would then need to inspect the final strings from the two solutions and decide which provided a more cogent or tractable answer to the research problem. Many research problems for which ORBDE is intended could require the analysis of many time series, e.g. one for each participant in the study. In those cases it would be recommended that the researcher apply the same stopping rule to all cases so that comparisons among experimental conditions can be made. On the other hand, watch what happens in the next demo.

```
The number of lines is 70 and number of distinct lines is 20.
-------------------------------------------------------------------------
              Frequency Distributions of Code Combinations
           Original                |            Aggregated
        Code Combination           |  Code  |  Number  |    Frequency
-------------------------------------------------------------------------
  DE                                    A          5           0.071
  AE                                    B          6           0.086
  DF                                    C          3           0.043
  CE                                    D          5           0.071
  BF                                    E          3           0.043
  CF                                    F          3           0.043
  EF                                    G          2           0.029
  CC                                    H          5           0.071
  CA                                    I          1           0.014
  DB                                    J          1           0.014
  DD                                    K          1           0.014
  BC                                    L          6           0.086
  BB                                    M          2           0.029
  CB                                    N          3           0.043
  BD                                    O         12           0.171
  AC                                    P          2           0.029
  CD                                    Q          2           0.029
  DC                                    R          1           0.014
  BE                                    S          5           0.071
  AD                                    T          2           0.029
-------------------------------------------------------------------------

AGGR-ORBDE: You have 2 original categorical variables.  Kolmogorov-Sinai entropy defaults
to Shannon entropy for C = 1 in the ORBDE output that follows.

-------------------------------------------------------------------------
                          Final Statistics
-------------------------------------------------------------------------
   C   trM     Ht        Dl       chi^2      df     N*    Phi^2      Hs
-------------------------------------------------------------------------
   1    7    2.807    16.566    31.860    19     70     0.455     2.768
   2    1    0.000     1.000    25.867     7     69     0.375     3.942
-------------------------------------------------------------------------
                         Partial Statistics
-------------------------------------------------------------------------
 Code (C=2)          rep?  Freq   Ex Freq    Pr-Obs    Shan      Sq Dev
-------------------------------------------------------------------------
OO                    y     5     2.028     0.072     0.190     4.513
OL                    -     3     1.014     0.043     0.136     3.254
LO                    -     3     1.014     0.043     0.136     3.254
NO                    -     2     0.507     0.029     0.103     2.745
HE                    -     2     0.211     0.029     0.103     4.496
HB                    -     2     0.422     0.029     0.103     3.110
SS                    -     2     0.352     0.029     0.103     3.474
GH                    -     1    63.452     0.014     0.061   -11.913
-------------------------------------------------------------------------
```

Even Lines with **`-Az`** *Subcommand*

The next demonstration repeats the aggregation analysis for even lines with the same data as in the previous demo, but with the **`-Az`** subcommand added. On the first page of output, all the same sequences of original codes appear, but this time all the codes that had an initial frequency of 1 are given the same aggregate code of lower case z. The substitution not only reduces the number of codes, but also produces a less ambiguous result. The solution is now *C* = 2 using both the *Ht* and χ^2 criteria.

```
The number of lines is 70 and number of aggregated codes including 'z' is 17.
-------------------------------------------------------------------------
              Frequency Distributions of Code Combinations
           Original                |            Aggregated
        Code Combination           |  Code  |  Number  |    Frequency
-------------------------------------------------------------------------
  DE                                   A          5          0.071
  AE                                   B          6          0.086
  DF                                   C          3          0.043
  CE                                   D          5          0.071
  BF                                   E          3          0.043
  CF                                   F          3          0.043
  EF                                   G          2          0.029
  CC                                   H          5          0.071
  CA                                   z          1          0.014
  DB                                   z          1          0.014
  DD                                   z          1          0.014
  BC                                   I          6          0.086
  BB                                   J          2          0.029
  CB                                   K          3          0.043
  BD                                   L         12          0.171
  AC                                   M          2          0.029
  CD                                   N          2          0.029
  DC                                   z          1          0.014
  BE                                   O          5          0.071
  AD                                   P          2          0.029
-------------------------------------------------------------------------
                          Final Statistics
-------------------------------------------------------------------------
   C   trM     Ht         Dl       chi^2      df      N*     Phi^2      Hs
-------------------------------------------------------------------------
   1    8     3.000     20.086     20.198     16      70     0.289     2.689
   2    1     0.000      1.000     36.768      9      69     0.533     3.901
-------------------------------------------------------------------------
                         Partial Statistics
-------------------------------------------------------------------------
 Code (C=2)           rep?  Freq   Ex Freq    Pr-Obs    Shan     Sq Dev
-------------------------------------------------------------------------
LL                     y      5     2.028     0.072     0.190     4.513
LI                     -      3     1.014     0.043     0.136     3.254
IL                     -      3     1.014     0.043     0.136     3.254
zz                     -      2     0.225     0.029     0.103     4.367
zI                     -      2     0.338     0.029     0.103     3.556
KL                     -      2     0.507     0.029     0.103     2.745
HE                     -      2     0.211     0.029     0.103     4.496
HB                     -      2     0.422     0.029     0.103     3.110
OO                     -      2     0.352     0.029     0.103     3.474
CA                     -      1    62.889     0.014     0.061   -14.385
-------------------------------------------------------------------------
```

Technical References

Guastello, S. J. (2000). Symbolic dynamic patterns of written exchange: Hierarchical structures in an electronic problem solving group. *Nonlinear Dynamics, Psychology, and Life Sciences, 4*, 169-188.

Guastello, S. J. (2011). Orbital decomposition: Identification of dynamical patterns in categorical data. In Guastello, S. J., & Gregson, R. A. M. (Eds.). *Nonlinear dynamical systems analysis for the behavioral sciences using real data* (pp. 499-516). Boca Raton, FL: C R C Press/Taylor & Francis.

Guastello, S. J., Gorin, H., Huschen, S., Peters, N. E., Fabisch, M., Poston, K., & Weinberger, K. (2013). The minimum entropy principle and task performance. *Nonlinear Dynamics, Psychology, and Life Sciences, 17, 405-424.*

Guastello, S. J., Gorin, H., Huschen, S. Peters, N. E., Fabisch, M., & Poston, K. (2012). New paradigm for task switching strategies while performing multiple tasks: Entropy and symbolic dynamics analysis of voluntary patterns. *Nonlinear Dynamics, Psychology, and Life Sciences, 16,* 471-497.

Guastello, S. J., Hyde, T., & Odak, M. (1998). Symbolic dynamic patterns of verbal exchange in a creative problem solving group. *Nonlinear Dynamics, Psychology, and Life Sciences, 2, 35-58.*

Guastello, S. J, Peressini, A. F., & Bond, R. W., Jr. (2011). Orbital decomposition for ill-behaved event-sequences: Transients and superordinate structures. *Nonlinear Dynamics, Psychology, and Life Sciences, 15,* 465-476.

Katerndahl, D., Ferrer, R., Burge,S., Becho, J., & Wood, R. (2010). Recurrent patterns of daily intimate partner violence and environment. *Nonlinear Dynamics, Psychology, and Life Sciences, 14,* 511-524.

Katerndahl, D. A., & Parchman, M. L. (2010). *Dynamical differences in patient encounters involving uncontrolled diabetes. Journal of Evaluation in Clinical Practice, 16,* 211-219.

Nathan, D. E., Guastello, S. J., Prost, R. W., & Jeutter, D. C. (2012). Understanding neuromotor strategy during functional upper extremity tasks using symbolic dynamics. *Nonlinear Dynamics, Psychology, and Life Sciences, 16,* 37-59.

Pincus, D. (2001). A framework and methodology for the study of nonlinear, self-organizing family dynamics. *Nonlinear Dynamics, Psychology and Life Sciences, 5,* 139-174.

Pincus, D., Eberle, K., Walder, C. S., Kemp, A. S., Lenjavi, M., & Sandman, C. A. (in press). The role of self-injury in behavioral slexibility and resilience. *Nonlinear Dynamics, Psychology and Life Sciences, 18*.

Pincus, D., & Guastello, S. J. (2005). Nonlinear dynamics and interpersonal correlates of verbal turn-taking patterns in group therapy. *Small Group Research*, *36,* 635-677.

Pincus, D., Ortega, D. L., & Metten, A. (2011). Orbital decomposition for multiple time series comparisons. In Guastello, S. J., & Gregson, R. A. M. (Eds.). *Nonlinear dynamical systems analysis for the behavioral sciences using real data* (pp. 517-538). Boca Raton, FL: C R C Press/Taylor & Francis.

Spohn, M. (2008). Violent societies: An application of orbital decomposition to the problem of human violence. *Nonlinear Dynamics, Psychology, and Life Sciences, 12,* 87-115.