CCEA v3

Software for Convergent
Cluster & Ensemble Analysis

(Updated June 9, 2008)

Sawtooth Software, Inc.
Sequim, WA
http://www.sawtoothsoftware.com

Bryan Orme, Editor

About Technical Support

We've designed this manual to teach you how to use our software and to serve as a reference to answer your questions. If you still have questions after consulting the manual, we offer telephone support.

When you call us, please be at your computer and have at hand any instructions or files associated with your problem, or a description of the sequence of keystrokes or events that led to your problem. This way, we can attempt to duplicate your problem and quickly arrive at a solution.

For customer support, contact our Sequim, Washington office at 360/681-2300, email: support@sawtoothsoftware.com, (fax: 360/681-2400).

Outside of the U.S., contact your Sawtooth Software representative for support.

Table of Contents

1 Overview

1.1 Introduction

Convergent Cluster & Ensemble Analysis (CCEA) is Sawtooth Software's tool for discovering groups within data. CCEA may be used to cluster survey respondents in market research applications (although the method is equally applicable for many other types of data). Therefore we describe the entities to be clustered as "respondents," though they need not be. For uniformity we use the term "variables" to describe the attributes on which respondents are measured, though we might equally well have called them "questions," "factors," or "items."

Cluster analysis consists of finding groups of cases (e.g. respondents) that tend to be similar *within* those groups on the basis variables (the variables used in clustering), but different on those same variables *between* the groups. Cluster ensemble analysis consists of leveraging a variety of cluster solutions (an *ensemble* of solutions) to find a single best *consensus* solution that has stronger characteristics than any one of the solutions within the ensemble.

CCEA may be considered the next generation to our previous CCA (Convergent Cluster Analysis) software system. In addition to the ensemble approach, CCEA includes the capabilities of CCA software for k-means cluster analysis. The literature argues that ensembles perform better than standard cluster analysis. Our work with CCEA v3 also supports that conclusion. Our cluster ensemble approach consistently obtains better solutions than the standard approach in CCA of selecting the highest-reproducibility solution for synthetic data sets with known cluster structure.

CCEA operates on data you've saved in comma-separated value (.csv) file formats. CSV files are a very common format, created using Excel, SPSS, SSI Web, and many popular software programs used by researchers. The user specifies the number of clusters desired, either for a single solution or for a range of solutions, such as "two through six clusters."

In using CCEA, you:

1. First select a file to use in your analysis. The file must be a comma-separated value (.csv) file format, with labels on the first row and caseid's in the first column (field).
2. Choose settings for the cluster run, including which variables to use, how many clusters to investigate, starting point strategies, and whether to standardize/center the variables.
3. Run the program and examine the output. Output files may be opened in programs such as Excel or SPSS. The results of CCEA output are often used in subsequent cross-tabulation analysis.

The quality of a cluster solution can depend importantly on the quality of the starting points. For this reason, CCEA dedicates a considerable portion of its resources generating "high quality" starting points and evaluating the reproducibility of solutions obtained from different sets of starting points. The run that is the most representative (reproducible) across multiple tries is returned as the final solution. This protects the analyst from stumbling into a poor solution based on an unlucky draw of starting points.

Because users may already have significant experience with CCA software, and because of its important historical precedent as a fine cluster approach, we include the standard CCA (Convergent Cluster Analysis) capabilities (with some algorithmic improvements over the previous CCA v2) within this software. That said, we are very enthusiastic about Ensemble Analysis, and our recent experience suggests users will obtain consistently better solutions. Users may decide to employ standard CCA cluster analysis within the package to investigate preliminary solutions, or especially to identify outliers (which is not a capability of Ensemble Analysis). After this preliminary work, we suggest applying what you have learned within CCEA's ensemble capabilities to obtain a final, even stronger, solution.

Uses of Cluster Analysis:

People seem to have a general interest in classifying things into groups. Given any large collection of things, we try to arrange them into categories. Marketers find it useful to think of their customers as "heavy users" and "light users." We think of ourselves as living in "big cities," "small towns," or "in the country." We think of occupations as "blue collar" and "white collar," siblings as "brothers" and "sisters," and people as "children" or "adults."

In all these cases it is easier for us to think about a small number of categories instead of a large number of individuals. However, things seldom fall neatly into groups, and the simplification achieved by grouping almost always entails loss of information.

Cluster analysis is a way of categorizing a collection of objects into groups (or "clusters"). Suppose we have a collection of objects, each of which has been described on a number of variables. The descriptions may be physical measurements, subjective ratings, or indications of the presence or absence of features.

We want to organize the objects into groups so that those within each group are relatively similar and those in different groups are relatively dissimilar. Almost any division of objects into distinct subsets can accomplish this.

We also hope that the groups will be "natural," and "compelling," and will reveal structure actually present in the data. If we were to plot the clusters in a multidimensional space, we would like them to be relatively dense aggregations of points, with empty space between them. Ideally, the data should look like a collection of cantaloupes in space, not like a single watermelon. At the very least, the points toward the centers of the clusters should be more densely concentrated than the points between them.

As an example of the motivation for doing a cluster analysis, we might want to buy a new car, or advise others about what car to buy. In either case it might be useful to organize the cars into classes such as "large," "medium," "small," "luxury," "sporty," etc. One way to do this would be with cluster analysis, where the objects would be cars and the variables could be price, fuel economy, top speed, number of passengers, etc.

As another example, in marketing a new breakfast cereal we might suspect that potential users of our product would have a wide range of attitudes toward such benefits as convenience, nutrition, and economy. We might find it helpful to have a sample of them describe their attitudes by expressing agreement or disagreement with a number of statements. We might perform a cluster analysis of those data to see if the individuals fell naturally into groups of potential customers desiring different benefits, each of which would be approached differently.

Underlying any such use of cluster analysis, there are several issues that must be addressed:

1. What measure of "similarity" among objects will be used?

2. What method will be used to assign objects to groups?

3. How will we decide how many groups to consider?

4. How can we be sure that the grouping obtained is "natural," reproducible, and useful?

In designing the first CCA systems and this later CCEA System, several requirements affected our thinking about these issues:

1. We wanted CCEA to be usable in conjunction with other Sawtooth Software products, which find application in market research and the social sciences. In both fields the objects studied are

often people who have responded to surveys. Those projects usually involve large data sets, consisting of hundreds and perhaps thousands of respondents.

2. Although it is always desirable for researchers to be expert in the techniques they are using, that is not always true. Therefore, we wanted to choose techniques most likely to "work every time." We considered it essential to use methods most likely to yield reproducible solutions, and also to give the user an indication of the reproducibility of each solution.

3. We know from our own experience and from the literature that it is not possible to fully "automate" the process of deciding which of several alternative clusterings to accept. However, there are statistical indicators which, although imperfect, can be helpful in this regard. Consequently, we have provided tables of "norms" to aid in such decisions in the appendices of this manual. These are based on approximately 20,000 "Monte Carlo" clusterings.

The literature on cluster analysis is voluminous. In earlier CCA manuals, we recommended an excellent review article (Milligan, G. W. and Cooper, M. C. "Methodology Review: Clustering Methods" in Applied Psychological Measurement, Vol II, No. 4, Dec 87) that provides an exceptionally rich source of information on the general topic of "what works."

However, many new developments have occurred since the 1980s. We are grateful to Joe Retzer and Ming Shan of Maritz Research for calling our attention to cluster ensemble methods in their presentation at the 2007 Sawtooth Software Conference entitled, "Cluster Ensemble Analysis and Graphical Depiction of Cluster Partitions." Retzer and Shan refer to an article by Strehl and Ghosh that has provided ideas that have been especially helpful in developing our unique cluster ensemble approach (Strehl and Ghosh, 2002, "Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions," Journal of Machine Learning Research 3 (2002) 583-617. Available for download at www.strehl.com/download/strehl-jmlr02.pdf.)

1.2 What's New in CCEA v3?

CCEA may be thought of the next generation CCA (Convergent Cluster Analysis) software, which had two previous versions. Because many users will have had experience with the previous CCA software, we'll describe the improvements this software makes over CCA v2.

1. We had added the capability of performing **Cluster Ensemble Analysis**. In repeated tests using data with known group structure, Cluster Ensemble Analysis outperforms CCA's cluster analysis approach. We are enthusiastic about this development, and encourage users to employ ensemble analysis to develop their final cluster solutions.

2. Users can **use cluster solutions obtained from outside sources** within their cluster ensemble. The data are simply pasted as new columns within the .csv file containing the ensemble.

3. **Improved Interface.** The Windows interface is new, and it follows the general feel of our popular CBC/HB system.

4. **Faster Speed.** Version 3 runs about twice as fast as the previous version in standard cluster analysis mode.

5. **Increased Numbers of Clusters.** The maximum number of clusters has been increased from 10 to 30.

6. **Increased Replications.** The maximum number of replications has been increased from 10 to 30 in standard cluster analysis mode. This might improve the quality of CCA's cluster solutions, as the final solution is selected by identifying the most representative (reproducible) solution across 30 replications rather than just 10 replications.

7. **Increased the number of cases used to develop starting points for larger data sets**. In previous versions of CCA, a maximum of 50 respondents were randomly drawn from the sample for developing starting points (hierarchical, distance-based, or density-based points). The idea of using a random subset of the sample to develop starting points is quite useful, as it provides variation in the high-quality starting points. However, for large datasets, drawing a larger subsample is warranted and may be beneficial for improving the starting points. In version 3, when a dataset exceeds 500 cases, we randomly sample 10% of the cases for developing starting points, up to a maximum of 250 cases. Thus, 1000 total cases in a dataset would employ a random draw of 100 cases for developing starting points.

8. We've added the ability to utilize **user-specified group means as starting points** in a user-defined .csv file. As an additional new feature, if multiple replications are specified, either user-specified start (group membership or group means) is used in addition to cycling among the other starting point methods. The most reproducible replication is returned as the final solution.

9. We have **modified the distance-based starting algorithm** so that it can produce different starting points across replications. Previously, the distance-based start was only used once and involved all respondents. However, by drawing a sub-sample of respondents (as done with the other two starting point algorithms), different distance-based starting points can be obtained across replications. Thus the distance-based starting points can be used multiple times in a Mixed starting point strategy.

10. Users may now **set a starting seed** for the random number generator (previously, a specific starting seed was hard-coded). This allows the user to repeat the entire run (that may include up to 30 replicates) from a different starting seed to test if one obtains a similar or the same solution.

11. We have **removed the option of variable weighting**. Variable weighting is a controversial option, and hasn't been used much. Interested users can still accomplish weighting by simply multiplying columns of their data by the desired weight prior to submitting to CCA.

12. We have dropped the ability to cross-tabulate group membership with other demographic variables. Most users have those capabilities with their standard cross-tabulation software.

2 Using CCEA

2.1 Opening a Data File

The first step in using CCEA is click ***File*** **|** ***Open*** and either:

- **Open a data file in the CSV Format.** Select this option to start a new project. The CSV (comma-separate value) file should contain variable labels on the first row (otherwise, default variable names will be used). The respondent number must be the first field (column) in the file. When you open a data file, CCA creates a project file named .CCA in the same location and with the same name as the data file you selected. When you close CCA and return to this project, you open the .CCA file to work with the data.
- **Open an existing CCEA project.** Select this option to open a project you've already been using within CCEA.

Appropriate Data for CCEA

CCEA works best when the variables you are using for finding clusters are continuous. Examples of continuous variables include: age, income, rating scales, Likert scales, importance scores. The variables should all have similar variance. If they do not, you should standardize them (giving them equal means and variances), by selecting the standardization option on the Settings tab. As an example, if one of your variables is respondent's weight (in kilos) and the other is annual household income (typically in the tens of thousands), unless standardized, the income variable will have many times greater impact on the solution than weight. The process of standardization is quite simple for a variable: first, we compute the mean and the standard deviation. Next we subtract the mean from each value (thus centering the values), and then we divide each of those values by the standard deviation. This results in values that average zero and have a standard deviation of 1.0.

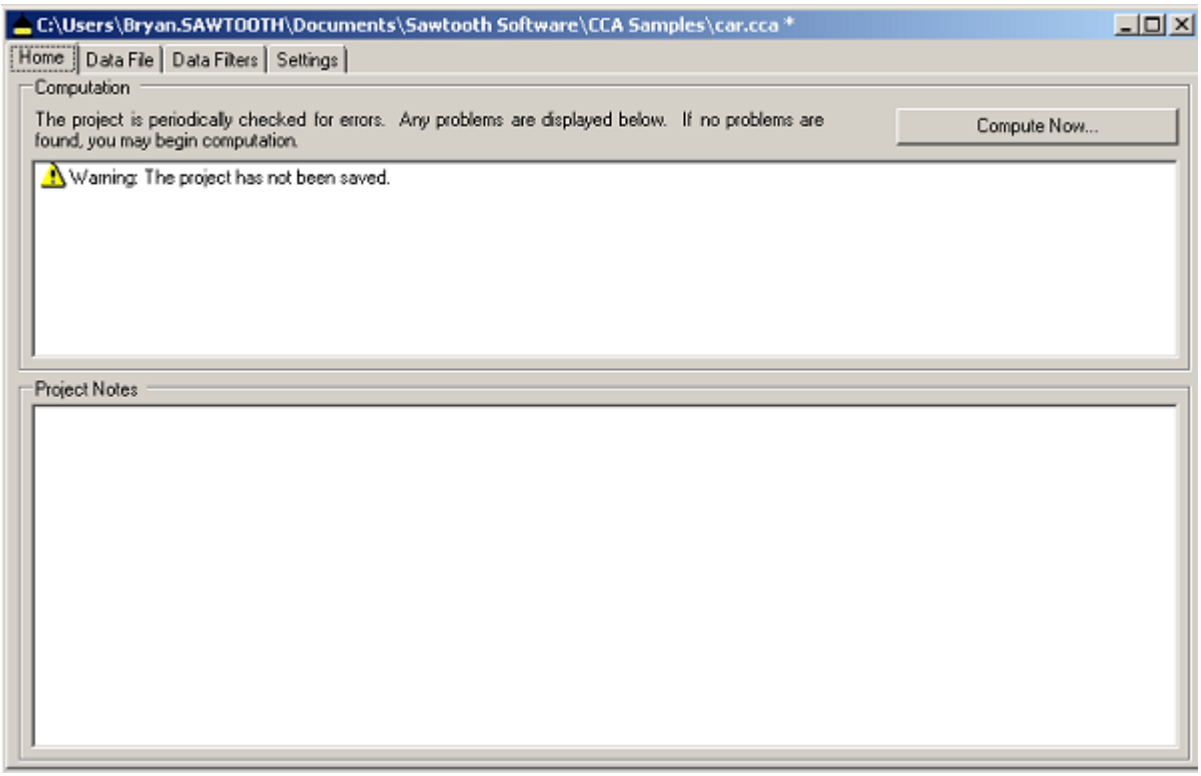
Categorical variables are generally not appropriate for use within CCEA. For example, a variable such as "preferred_color" where 1=blue, 2=red, 3=green, etc. would not be appropriate for use in CCEA. CCA expects increasing values to indicate "more" of the variable and decreasing values to mean "less." Such is not the case with categorical variables.

Although one can cluster on individual-level utilities resulting from an HB analysis of Choice-Based Conjoint or MaxDiff (best-worst scaling), it is probably more appropriate to utilize Latent Class procedures for these cases. Using CCEA would involve the two-stage procedure of first computing utilities using HB, and then secondly using those data within CCEA (where any errors in the first stage would be accepted as "truth" in the second phase). Latent Class provides a way to simultaneously estimate part-worth utilities and divide the sample into meaningful segments.

Note: CCEA's clustering algorithms cannot handle missing data. You can include variables in the .csv file that have missing values. But, variables selected for use in clustering must not have any missing values.

2.2 Home, Data File, Data Filters tabs

When you have opened a data file or an existing CCEA project, the main dialog is displayed:



Home
This tab provides the ***Compute Now...*** button, which runs the cluster/ensemble analysis. It also displays any errors or warnings regarding your project.

You can write or paste text into the *Project Notes* field to help you in your work.

Data File
This tab shows you which data file (.csv format) is currently selected for the current project. The CSV (comma-separate value) file should contain variable labels on the first row (otherwise, default variable names will be used). The respondent number must be the first field (column) in the file. ***View Data File...*** opens the file in view-only mode so you can inspect its contents. ***View Data File Summary...*** scans your data file and provides a summary to the screen: number of cases, number of variables per case, and variable labels.

Data Filters
Here, you select which variables to include in the analysis. The variables you select must contain no missing values.

Main Menu

Note that above the main workspace are the main menu items:

File: Used to open and close studies.

Edit: Used to open/edit the data file. Also used to access a *Preferences* dialog. The preferences dialog lets you customize the output format and the files that are saved.

Analysis: May be used to start the computation or to specify Batch Processing of cluster/ensemble runs.

Tools: Includes options to pre-process the data (standardization and centering) and to save the transformed data to another .csv file. (Note: if using the centering option, all variables in the file will be centered, irrespective of what is selected in the Data Filters tab.)

Window: Offers standard Windows controls.

Help: Accesses the Help files, documentation, edit the User ID files, email for technical support, or check for later versions of this software on the web.

2.3 Settings Tab (Cluster Analysis Mode)

When you click the *K-Means, highest reproducibility replicate* option for Cluster Method, this dialog displays the settings that govern your cluster analysis run.

Minimum and Maximum numbers of groups:
Define the limits to be explored automatically. If you set the minimum at 2 and the maximum at 30 groups (which are the extremes permitted), then solutions are computed for 2 clusters, 3 clusters, 4 clusters, etc., up to 30 clusters. Each clustering is done independently of the others. For most problems, no more than 10 clusters are requested. *Warning: running all solutions between 2 and 30 would potentially require a great deal of time to compute!*

If you want only a single solution, use that number as the minimum and maximum (for instance, specify a minimum of 5 and a maximum of 5 if you want only the 5-cluster solution).

Number of replications for each solution:
Determines how many times each clustering is repeated. For example, if you choose 10 replications, then each clustering is done 10 times. The "most reproducible" solution is chosen, which protects you from stumbling into a poor solution due to an unlucky draw of starting points. CCEA can be run for only one replication, but a specification of "1" fails to take advantage of CCEA's valuable capability of assessing and maximizing reproducibility of its solutions. We urge that you choose at least 10 replications, and 30 is best, though it may take noticeably longer for large datasets.

Minimum group size:
This option lets you specify a minimum number of respondents that you would like to have in every cluster. When there are several replications of a clustering, CCEA normally selects the most reproducible of them. If you specify a minimum size, then only replications with all clusters at least that large will be considered. If no replication qualifies, then the most reproducible is chosen, regardless of minimum cluster size.

Outlier Distance:
Governs CCEA's treatment of outliers. If you want every respondent to be in a cluster, check *Include all cases (no outliers)*. Otherwise, enter a number between 1 and 10. A respondent is treated as an outlier and not assigned to a cluster if his/her distance to the nearest cluster is greater than X times the average distance for all respondents where X is the value you enter (where distance is Euclidean distance squared). Entering a value of 1 will cause many respondents to be regarded as outliers: all those more distant than average from their nearest cluster. Entering a larger value, such as 3, will reject only those that are three times as far as average from their nearest cluster. The largest value, 10, will treat only the most extreme cases as outliers, assigning almost all respondents to clusters.

Standardize variables:
The variables should all have similar variance. If they do not, you should standardize them (giving them equal means and variances), by selecting the standardization option. As an example, if one of your variables is respondent's weight (in kilos) and the other is annual household income (typically in the tens of thousands), unless standardized, the income variable will have many times greater impact on the solution than weight. The process of standardization is quite simple for a variable: first, we compute the mean and the standard deviation. Next we subtract the mean from each value (thus centering the values), and then we divide each of those values by the standard deviation. This results in values that average zero and have a standard deviation of 1.0. For more information, see Preprocessing Options.

Center cases:
It is commonly known that respondents tend to answer ratings questions using certain scale use biases, such as tending to use the upper part or the lower part of the scale. If that is the case, it may be helpful to center the data, helping to reduce the problems due to clustering on data

affected by these biases. This operation computes the mean for each respondent over all the variables, and then subtracts it from all of them so that the new mean for each respondent is zero. If both standardizing and centering are requested, the standardization is done first and then the standardized data are centered. For more information, see Preprocessing Options.

Starting point strategy:
This is detailed further in the Starting Points section of this documentation. The choices are:

Distance-based starting points

Hierarchical-based starting points

Density-based starting points

Mixed strategy (alternates among all the starting point methods.)

User-defined groups
User-defined groups uses an initial solution that you provide. This option will be used only when preliminary work has suggested groupings for these respondents and you want to iterate from that starting point (or use that solution as one of multiple replicates from different starting points, for judging reproducibility). You supply a .csv file containing the respondent numbers and their group membership. The format is the same as the **STUDYNAME_membership.csv** file that is saved after every run. For example, you may have established 2-group, 3-group, and 4-group solutions using a different method. In that case, your .csv file should contain four variables (fields) per row: respondent#, 2Groups, 3Groups, and 4Groups. A header row with labels is required. When viewed within Excel, the file might appear as follows (for the first 5 records):

	A	B	C	D	E
1	Case	2 Groups	3 Groups	4 Groups	
2	1	1	3	2	
3	2	2	1	1	
4	3	1	2	3	
5	4	1	2	3	
6	5	2	1	1	

When viewed as a text-only file, the contents of the file appear as:

```
Case, 2 Groups, 3 Groups, 4 Groups
1, 1, 3, 2
2, 2, 1, 1
3, 1, 2, 3
4, 1, 2, 3
5, 2, 1, 1
```

meaning, respondent #1 is assigned to group 1 for the two-group solution, group 3 in the 3-group solution, and group 2 in the 4-group solution, etc.

User-defined means
If you already have established group profiles in terms of means on the basis variables, you can use these means as starting points for analysis using CCEA. You supply a .csv file containing the groups and their means on the variables. The format is the same as the **STUDYNAME_groups.csv** file that is saved after every run. The file contains a first row of labels: "Number of Groups, Group, variable1, variable2. etc. For example, the following (as

viewed within Excel) supplies a starting point based on just three variables for a 3-group solution:

	A	B	C	D	E
1	Number of Groups	Group	a	b	c
2	3	1	1.43	2.33	0.25
3	3	2	2.50	1.00	3.60
4	3	3	0.25	0.19	2.28
5					

When viewed as a text-only file, the contents of the file appear as:

```
Number of Groups, Group, a, b, c
3, 1, 1.43, 2.33, 0.25
3, 2, 2.50, 1.00, 3.60
3, 3, 0.25, 0.19, 2.28
```

If you specify just one replication, your user-defined starting point will be the only starting method used. If you specify more than one replication, then the other methods employed in the Mixed Strategy will be used and the solution with the highest reproducibility across the replicates will be selected.

There is evidence that all of the methods for getting starting points are capable of producing "best" solutions, and that any of them can also do a bad job on occasion. That is why we prefer at least 10 replications, and we suggest that you use the *Mixed* strategy, which makes use of a variety of methods.

Random Starting Seed:
The entire process may be repeated using a different starting seed. We strongly encourage users to repeat the analysis from different starting seeds, to assess the stability of the solutions.

2.4 Settings Tab (Ensemble Analysis Mode)

When you click the *Ensemble, consensus solution* option as the Cluster Method, this dialog displays the settings that govern your Ensemble Analysis run.

Minimum and Maximum numbers of groups:
Define the limits to be explored automatically. If you set the minimum at 2 and the maximum at 30 groups (which are the extremes permitted), then solutions are computed for 2 clusters, 3 clusters, 4 clusters, etc., up to 30 clusters. For most problems, no more than 10 clusters are requested. *Warning: running all solutions between 2 and 30 would potentially require a great deal of time to compute!*

If you want only a single solution, use that number as the minimum and maximum (for instance, specify a minimum of 5 and a maximum of 5 if you want only the 5-cluster solution).

Standardize variables:
The variables should all have similar variance. If they do not, you should standardize them (giving them equal means and variances), by selecting the standardization option. As an example, if one of your variables is respondent's weight (in kilos) and the other is annual household income (typically in the tens of thousands), unless standardized, the income variable will have many times greater impact on the solution than weight. The process of standardization is quite simple for a variable: first, we compute the mean and the standard deviation. Next we subtract the mean from each value (thus centering the values), and then we divide each of those values by the standard deviation. This results in values that average zero and have a standard deviation of 1.0. For more information, see Preprocessing Options.

Center cases:
It is commonly known that respondents tend to answer ratings questions using certain scale use biases, such as tending to use the upper part or the lower part of the scale. If that is the case, it may be helpful to center the data, helping to reduce the problems due to clustering on data affected by these biases. This operation computes the mean for each respondent over all the variables, and then subtracts it from all of them so that the new mean for each respondent is zero. If both standardizing and centering are requested, the standardization is done first and then the standardized data are centered. For more information, see Preprocessing Options.

Allow CCEA to build the ensemble:
If you do not provide a custom ensemble file (described below), CCEA software will develop an ensemble containing a variety of cluster solutions for you. By default, an ensemble is built with 70 separate cluster solutions, ranging from 2 to 30 groups, employing five different cluster strategies: k-means (distance-based starting point), k-means (density-based starting point), k-means (hierarchical starting point), hierarchical (average linkage criterion), and hierarchical (complete linkage criterion).

Based on our experience, we have found it useful to include a wide variety and large number of solutions in the ensemble.

You may ***Add...*** or ***Remove...*** cluster solutions from the ensemble. You can multi-select runs (so that you can delete multiple runs in the same operation) by clicking on the first run to delete, and while holding down the Shift key, clicking on the last run to delete. You can also muti-select individual runs (rather than a continuous series) by holding down the Ctrl key and selecting multiple runs.

Use an existing ensemble file:
Some users may wish to construct their own ensemble of cluster solutions to employ within CCEA. This is very easy to do, as the ensemble file may be edited with Excel and it follows a very similar format as the STUDYNAME_membership.csv and the STUDYNAME_ensemble.csv files produced by CCEA. The ensemble file may contain cluster solutions produced by CCEA software

along with those provided using other software and approaches. The file format is a .csv file (comma-separate values). When viewed in Excel, the format looks like the following:

	A	B	C	D
1	Caseid	2	5	11
2	1	1	4	10
3	2	1	2	1
4	3	2	5	2
5	4	1	1	7
6	5	2	1	4

The first row contains labels, with the first column containing an arbitrary text label such as "Caseid". Respondent records follow, one per line, with respondent numbers indicated in the first column. In this example, respondent numbers are 1 through 5. Following that first column, each additional column (for as many columns as there are cluster solutions in your ensemble) is headed by a label describing how many groups are found in that segmentation solution. For example, column B row 1 contains the header "2," indicating that this column contains data for a 2-group solution. Respondent number 1 is assigned to group #1, respondent number 3 is assigned to group #2, etc. There are three cluster solutions shown, for 2, 5, and 11 groups.

When viewed with a text editor, this .csv file looks like:

```
Caseid,2,5,11
1,1,4,10
2,1,2,1
3,2,5,2
4,1,1,7
5,2,1,4
```

Random Starting Seed:
The entire process may be repeated using a different starting seed. We strongly encourage users to repeat the analysis from different starting seeds, to assess the stability of the solutions.

2.5 Batch Processing

CCEA lets you submit more than one cluster (or ensemble) analysis run at once. This can be helpful if you have particularly large data sets and you want to run a series of solutions (perhaps with different selections of variables and pre-processing options) unattended. ***Click Analysis | Batch Computation...*** to select multiple projects to run in batch mode.

When you open a data file in CCEA, CCEA creates a project file named .CCA in the same folder as your .csv data file, and with the same name (prefix). Click ***Add...*** to add each project to the list of cluster analyses to be run. Click ***Compute*** to run all of them in batch sequence.

Hint: To facilitate the creation of multiple (similar) projects for selecting in batch mode, do not forget that you can ue the ***File | Save As...*** option to create a copy of an existing project. Then, you can modify that copy to make selected changes.

3 Example Cluster Analysis Run

3.1 Example Cluster Analysis Run

Running either k-means cluster analysis or ensemble analysis follows the same essential steps. We describe the process of running cluster analysis below.

When you install CCEA, a data set is included called "Car". This dataset has data on 62 vehicles on a variety of performance measures, including:

Price (in 1000s of dollars)
Acceleration time 0-60
1/4 mile elapsed time
Top speed
Braking feet from 80 mph
Slalom Speed
Skidpad g factor
Interior noise DB
Fuel mpg

For your information, the 62 cars (cases) used in this data set are:

01 Acura Integra RS
02 Acura Legend
03 Acura Legend Coupe L
04 Alfa Milano Verde 3.0
05 Audi 5000S
06 Bitter SC
07 BMW M3
08 BMW 535i
09 BMW 635i
10 BMW 735i
11 Buick Reatta
12 Cadillac Allante
13 Chevrolet Beretta GT
14 Chevrolet Camaro IROC-Z
15 Chevrolet Cavalier Z24
16 Chevrolet Corvette (Convertible)
17 Chevrolet Nova CL
18 Chevrolet Nova Twin Cam
19 Chrysler LeBaron Coupe
20 Ford Mustang GT
21 Ford Thunderbird Turbo Coupe
22 Honda Accord LXi
23 Honda Prelude 2.0 Si 4ws
24 Jaguar XJ-S
25 Mazda RX-7 Convertible
26 Mazda RX-7 GXL
27 Mazda RX-7 Turbo
28 Mazda MX-6 GT
29 Mercedes 190E 2.3
30 Mercedes 300E
31 Mercedes 420 SEL
32 Mercedes 560 SEC
33 Mercedes 560 SL
34 Mercury Sable LS

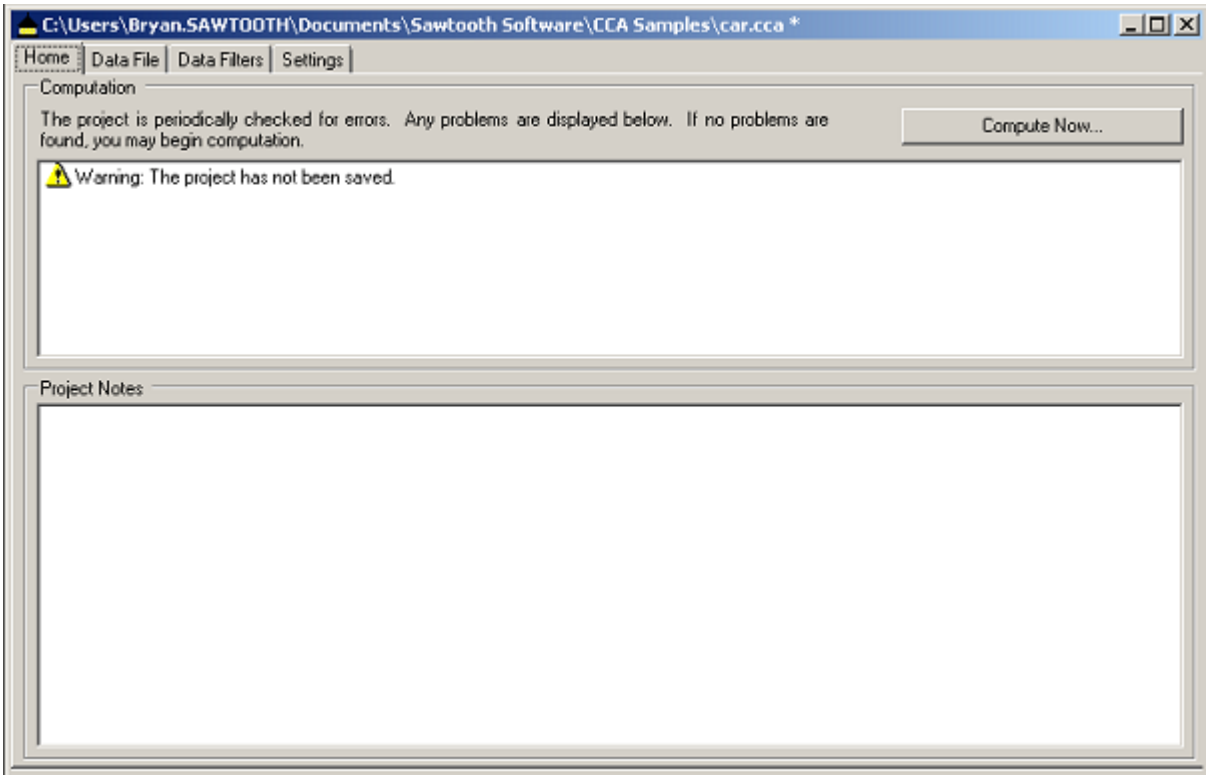
35 Merkur Scorpio
36 Mitsubishi Cordia
37 Nissan Pulsar NX SE
38 Nissan Sentra SE Coupe
39 Nissan 300ZX 2+2
40 Peuogeot 505 STX
41 Pontiac Fiero Formula
42 Pontiac Firebird Trans Am GTA
43 Pontiac Lemans
44 Porsche 924S
45 Porsche 944
46 Porsche 944S
47 Porsche 944 Turbo
48 Porsche 911 Cabriolet
49 Porsche 928S
50 Saab 9000
51 Shelby Omni GLH-S
52 Sterling 825 SL
53 Subaru 4wd Turbo XT Coupe
54 Toyota Celica GT-S
55 Toyota Celica All-Trac
56 Toyota Corolla FX16 GT-S
57 Toyota MR2
58 Toyota Supra
59 Volkswagen Golf 16V
60 Volkswagen Quantum Syknchro
61 Volkswagen Scirocco 16V
62 Volvo 740 Turbo Wagon

Running Cluster Analysis on the Car.csv Dataset

1. To start CCEA, click ***Start | Programs | Sawtooth Software | Sawtooth Software CCEA***. The main menu is displayed, with ***File*** in the upper-left hand corner.

2. Select ***File | Open***. Choose the first option, to *Open a data file in the CSV format*. Browse to the **Car.csv** file (or use **Car (Intl).csv** for some international users) located in your ...**Documents\Sawtooth Software\CCEA Samples** folder.

The workspace dialog appears:

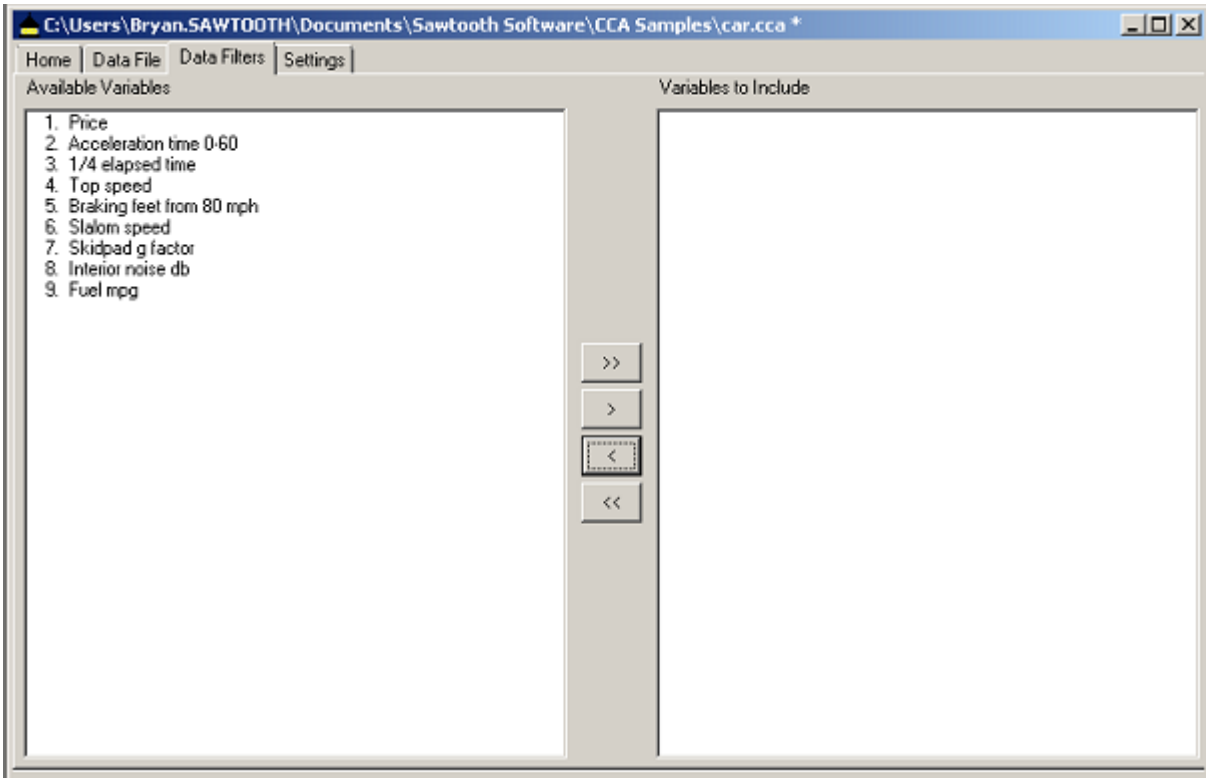


Click the *Data File* tab, to see a summary of your data file. Click the ***View Data File Summary...*** button. CCEA reads the data file and produces a simple summary:

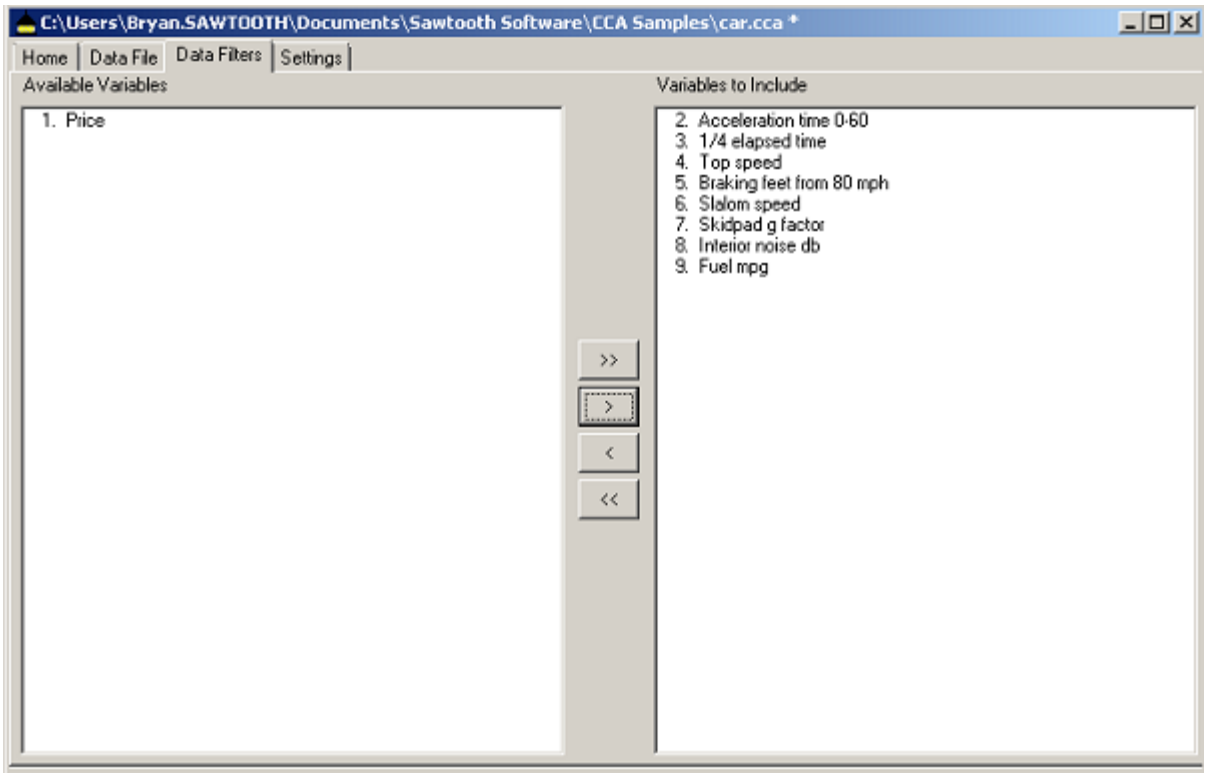
```
Summary of C:\Users\Bryan.SAWTOOTH\Documents\Sawtooth Software\CCEA Samples\car.csv
   Number of cases: 62
   Number of variables per case: 9
   The file contains these variable labels:
      Price
      Acceleration time 0-60
      1/4 elapsed time
      Top speed
      Braking feet from 80 mph
      Slalom speed
      Skidpad g factor
      Interior noise db
      Fuel mpg
```

You can use this report to make sure your data file has been read properly and that the contents are as expected.

Click the *Data Filters* tab to select which variables should be used in clustering. The following is displayed:

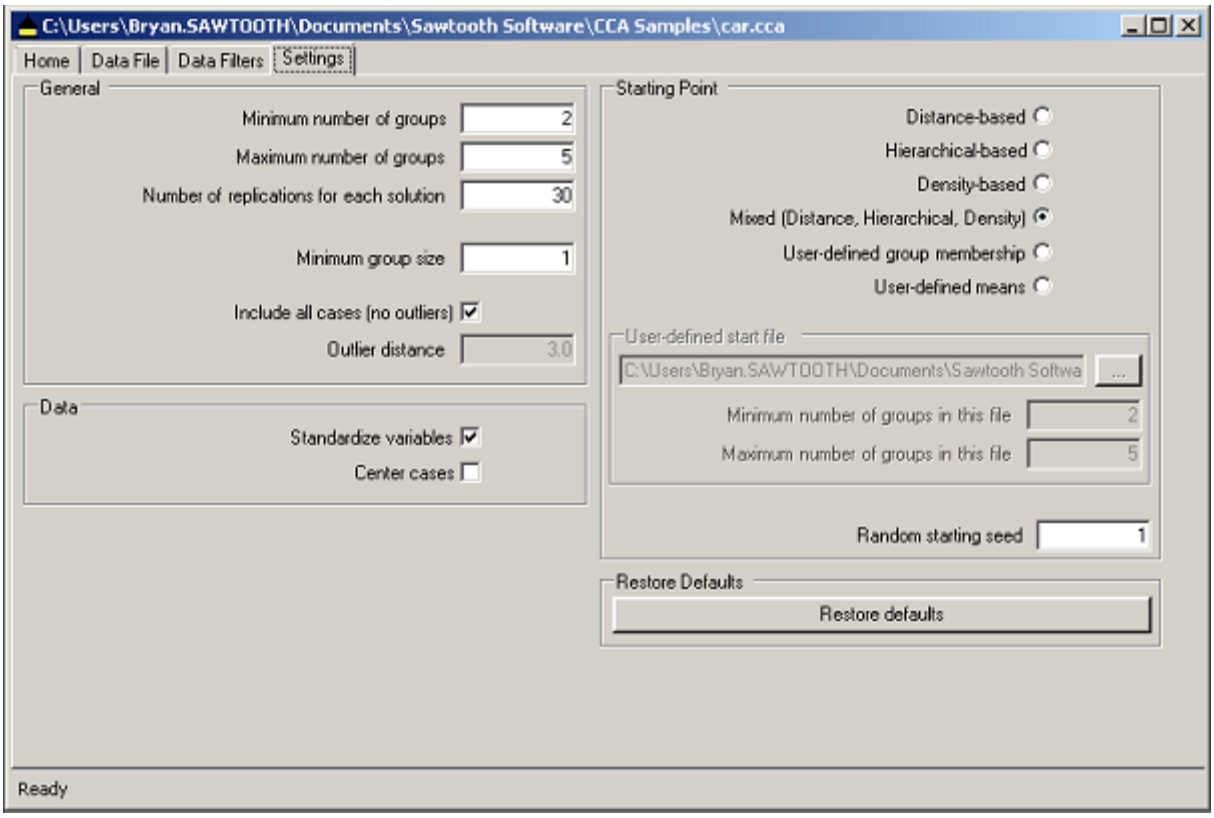


Move the variables you wish to use in your cluster analysis (the basis variables) to the right-hand panel. In this example, we'll use variables 2 through 9 (we'll not use Price in our cluster run). You can move all of the variables to the right-hand panel by clicking the >> button, and then moving just the Price variable back to the left-hand panel by highlighting it and clicking the < button. Or, you can multi-select the eight variables to include in the analysis in the left panel (while holding the Shift key down, click the 2nd variable and then click the 9th variable), and once the group is highlighted click the > button to move the highlighted group to the right-hand panel. In either case, once you've selected the eight variables to include in the analysis, the display looks like:



With k-means cluster analysis, it is important for all variables to have similar scale. Otherwise, a variable with much larger scale will have much greater impact on the final solution. For example, if clustering people (rather than cars) it wouldn't be appropriate to include a variable indicating number of years of school completed with a variable indicating household income. Highest grade in school typically ranges from about 8 to 20, whereas income is typically measured in the thousands. The same logic applies for many of our variables in the cars dataset. To avoid the problem of misweighting the variables, we shall standardize them, giving each a mean of zero and a variance of 1.0. We'll select the standardization option on the *Settings* tab.

Click the *Settings* tab, and specify the following options:



By default, CCEA will investigate 2 through 5-cluster solutions (Minimum number of groups=2; Maximum number of groups=5).

The default is to repeat the analysis 10 separate times (replicates), using a Mixed starting point strategy, but this dataset is so small that it is no additional wait to run 30 replications, and the results will potentially be better. **Change the number of replications from 10 to 30.**

In this example, all cases (cars) will be assigned to clusters (Include all cases--no outliers), rather than permitting classification of outliers. Furthermore, we've decided to standardize the variables, giving each a mean of 0 and a variance of 1.0. Check the *Standardize variables* box.

CCEA Output

To run the computation, return to the *Home* tab and click ***Compute Now...***.

CCEA first returns a brief summary of the data file:

```
Data File: C:\Users\Bryan.SAWTOOTH\Documents\Sawtooth Software\CCEA Samples\car.csv
Variables are standardized.
Number of cases: 62
Number of variables per case: 9
The file contains these variables:
      1. Price
      2. Acceleration time 0-60
      3. 1/4 elapsed time
      4. Top speed
      5. Braking feet from 80 mph
      6. Slalom speed
      7. Skidpad g factor
```

```
      8. Interior noise db
      9. Fuel mpg

Variables included in this computation:
      2. Acceleration time 0-60
      3. 1/4 elapsed time
      4. Top speed
      5. Braking feet from 80 mph
      6. Slalom speed
      7. Skidpad g factor
      8. Interior noise db
      9. Fuel mpg
```

Click ***Continue with estimation***, and CCEA performs its k-means cluster analysis. The procedure is as follows:

Each solution proceeds automatically with these steps:

1. A set of "starting points" is determined. There are as many starting points as clusters desired.
2. Each respondent is classified into a group corresponding to the starting point to which he is most similar.
3. The averages for each variable are computed for the respondents in each group. These averages replace the starting points.
4. Steps 2 and 3 are repeated until no respondents are reclassified from the previous iteration.

The quality of the solution can depend importantly on the quality of the starting points. For this reason, CCEA dedicates a considerable portion of its resources generating "high quality" starting points and evaluating the reproducibility of solutions obtained from different sets of starting points.

The output is quite voluminous, so we'll break it down into sections and describe each part.

First, a summary of the settings for your run is displayed:

```
CCEA Computation (12/13/2007 12:50:41 PM)
=================================================

Minimum number of groups                      2
Maximum number of groups                      5
Number of replications                       30
Maximum number of iterations               1000
Minimum group size                            1
Including all cases (no outliers)
Random number seed                            1
Using a mix of starting point methods.
```

Next, a summary of the replications is displayed for the two-group solution, showing which starting point methods were used, how many iterations it took to converge in each case, and how many cases were in each of the two groups.

```
Solution for 2 Groups
Replication   Start          Iterations   Group sizes
-------------------------------------------------------------------------------
      1       Distance            2         6    56
      2       Hierarchical        2        56     6
      3       Density             6        30    32
      4       Distance            9        42    20
      5       Hierarchical        2         6    56
      6       Density            12        42    20
      7       Distance            3         6    56
      8       Hierarchical        2        55     7
      9       Density             6        40    22
```

10	Distance	3	6	56
11	Hierarchical	2	55	7
12	Density	9	30	32
13	Distance	6	30	32
14	Hierarchical	7	30	32
15	Density	9	30	32
16	Distance	5	16	46
17	Hierarchical	2	55	7
18	Density	9	30	32
19	Distance	5	16	46
20	Hierarchical	5	22	40
21	Density	6	33	29
22	Distance	2	56	6
23	Hierarchical	2	55	7
24	Density	6	40	22
25	Distance	13	20	42
26	Hierarchical	2	53	9
27	Density	9	30	32
28	Distance	5	46	16
29	Hierarchical	2	6	56
30	Density	6	40	22

The first replication used the Distance-based starting point strategy. It took only 2 iterations for the solution to converge, and the two groups found contained 6 and 56 cases. It is evident from this table that there is quite a bit of variability among the 30 replicates of the 2-group solution. It would not appear that the data naturally cluster in a very stable way into 2 groups.

The next output table contains the adjusted pairwise reproducibility for all replicates, displaying how consistently cases were assigned to groups when the procedure was repeated from different starting points. (We've only shown a portion of the table, as 30x30 replicates is too wide to display easily in this documentation.)

Pairwise Reproducibility of Replicates (2 Groups)

	1	2	3	4	5	6	7	8	9	10	11	12	...	30
1	-----	100.0	22.6	54.8	100.0	54.8	100.0	96.8	9.7	100.0	96.8	22.6	...	9.7
2	100.0	-----	22.6	54.8	100.0	54.8	100.0	96.8	9.7	100.0	96.8	22.6	...	9.7
3	22.6	22.6	-----	67.7	22.6	67.7	22.6	25.8	67.7	22.6	25.8	100.0	...	67.7
4	54.8	54.8	67.7	-----	54.8	100.0	54.8	58.1	35.5	54.8	58.1	67.7	...	35.5
5	100.0	100.0	22.6	54.8	-----	54.8	100.0	96.8	9.7	100.0	96.8	22.6	...	9.7
6	54.8	54.8	67.7	100.0	54.8	-----	54.8	58.1	35.5	54.8	58.1	67.7	...	35.5
7	100.0	100.0	22.6	54.8	100.0	54.8	-----	96.8	9.7	100.0	96.8	22.6	...	9.7
8	96.8	96.8	25.8	58.1	96.8	58.1	96.8	-----	6.5	96.8	100.0	25.8	...	6.5
9	9.7	9.7	67.7	35.5	9.7	35.5	9.7	6.5	-----	9.7	6.5	67.7	...	100.0
10	100.0	100.0	22.6	54.8	100.0	54.8	100.0	96.8	9.7	-----	96.8	22.6	...	9.7
11	96.8	96.8	25.8	58.1	96.8	58.1	96.8	100.0	6.5	96.8	-----	25.8	...	6.5
12	22.6	22.6	100.0	67.7	22.6	67.7	22.6	25.8	67.7	22.6	25.8	-----	...	67.7
.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
30	9.7	9.7	67.7	35.5	9.7	35.5	9.7	6.5	100.0	9.7	6.5	67.7	...	-----
Avg	57.1	57.1	55.7	62.6	57.1	62.6	57.1	58.0	38.6	57.1	58.0	55.7	...	38.6

```
Replication 6 (density start) has the best reproducibility (62.6%)
The elapsed time for this solution was 0:00:00.
```

The average reproducibility for each replication is displayed and the replication with the best reproducibility is noted (ties are broken randomly). In this example, replicates 1, 2, 5, 7, and 10 were identical. But the cluster solution reflected by these replicates seems to be a less representative solution. Across all replicates, replication #6 seems to be the most reproducible, with an average adjusted reproducibility of 62.6%. It is taken as the "best" two-group solution, and the cluster membership for each case is saved to the **car_membership.csv** file.

Next, the cluster means and F ratios are computed and displayed. (We chose to standardize variables so the means are centered around zero):

```
Group Means and F Ratios
                                 1          2        --F--
-------------------------------------------------------------
   1. Acceleration time 0-      -0.48       1.01      58.52
   2. 1/4 elapsed time          -0.48       1.02      60.01
   3. Top speed                  0.45      -0.95      46.23
   4. Braking feet from 80      -0.15       0.32       3.16
   5. Slalom speed               0.17      -0.36       3.97
   6. Skidpad g factor           0.38      -0.81      27.48
   7. Interior noise db          0.12      -0.26       1.96
   8. Fuel mpg                  -0.40       0.85      31.96

Group Size                        42         20      23.25
```

The number of cases in each cluster is displayed (Cluster 1 has 42 cases, Cluster 2 has 20). Using the first line as an example, Acceleration time has a mean for Cluster 1 of -0.48, a mean for Cluster 2 of 1.01, and an F ratio of 58.52. The F ratios indicate the relative amount of difference among clusters for each of the variables. F ratios are obtained by dividing a "mean square between clusters" by a "mean square within clusters."

At the bottom of the last column, in the line designated "Group Size," is a "pooled" F ratio (23.25). The pooled F ratio is obtained by summing the numerators and denominators for the individual F ratios separately, and then dividing the sum of the numerators by the sum of the denominators. The pooled F ratio is an overall indicator of the amount of difference between clusters. Comparing these values for different cluster solutions may be helpful in deciding how many clusters to use.

These F ratios are provided as descriptive statistics, not as values that should be tested for statistical significance. Since the clusters were constructed to be as different from one another as possible on these variables, it would be inappropriate to test whether differences on these same variables are greater than would be expected "due to chance alone." These F ratios would almost certainly appear "highly significant" if tested, even if the data had consisted of nothing but random numbers. However, the F ratios are valuable as descriptive measures of the relative importance of each variable in the clustering. Those variables with the largest F ratios are those on which the clusters are most different. Variables with the smallest F ratios could probably have been omitted without much effect on the cluster analysis results.

Then, the means as deviations from grand means are computed and displayed:

```
Group Means as Deviations from Grand Means and F Ratios
                                 1          2        --F--
-------------------------------------------------------------
   1. Acceleration time 0-      -0.48       1.01      58.52
   2. 1/4 elapsed time          -0.48       1.02      60.01
   3. Top speed                  0.45      -0.95      46.23
   4. Braking feet from 80      -0.15       0.32       3.16
   5. Slalom speed               0.17      -0.36       3.97
   6. Skidpad g factor           0.38      -0.81      27.48
   7. Interior noise db          0.12      -0.26       1.96
   8. Fuel mpg                  -0.40       0.85      31.96

Group Size                        42         20      23.25
```

(When variables are standardized, this screen is always identical to the previous screen.)

Then, for each cluster, the variables are sorted by the cluster's means expressed as deviations from grand means. This display lets you quickly see on which variables this cluster has a higher (more positive) or lower (more negative) value than other clusters. For each variable, the number of any other cluster with a more extreme deviation with the same sign is also noted.

For illustration, we'll jump ahead in the output and display the three-group solution result from this same run. The three-group solution found group sizes of 18, 38, and 6 cases:

```
Group Means as Deviations from Grand Means and F Ratios

                                    1          2          3        --F--
 ------------------------------------------------------------------------
    1. Acceleration time 0-      -0.91       0.10       2.14      69.43
    2. 1/4 elapsed time          -0.98       0.16       1.96      62.49
    3. Top speed                  0.72      -0.08      -1.66      21.98
    4. Braking feet from 80      -0.28      -0.11       1.56      11.05
    5. Slalom speed               0.61      -0.11      -1.14       9.49
    6. Skidpad g factor           0.96      -0.28      -1.10      22.95
    7. Interior noise db          0.92      -0.40      -0.22      16.29
    8. Fuel mpg                  -0.41      -0.04       1.51      11.27

  Group Size                       18         38          6      21.73

  Group 1 (18 cases) sorted by Deviations from Grand Means
  Variable                        Mean        Dev    More Extreme (w/ same sign)
     6. Skidpad g factor          0.96       0.96
     7. Interior noise db         0.92       0.92
     3. Top speed                 0.72       0.72
     5. Slalom speed              0.61       0.61
     4. Braking feet from 80     -0.28      -0.28
     8. Fuel mpg                 -0.41      -0.41
     1. Acceleration time 0-     -0.91      -0.91
     2. 1/4 elapsed time         -0.98      -0.98

  Group 2 (38 cases) sorted by Deviations from Grand Means
  Variable                        Mean        Dev    More Extreme (w/ same sign)
     2. 1/4 elapsed time          0.16       0.16    3
     1. Acceleration time 0-      0.10       0.10    3
     8. Fuel mpg                 -0.04      -0.04    1
     3. Top speed                -0.08      -0.08    3
     5. Slalom speed             -0.11      -0.11    3
     4. Braking feet from 80     -0.11      -0.11    1
     6. Skidpad g factor         -0.28      -0.28    3
     7. Interior noise db        -0.40      -0.40

  Group 3 (6 cases) sorted by Deviations from Grand Means
  Variable                        Mean        Dev    More Extreme (w/ same sign)
     1. Acceleration time 0-      2.14       2.14
     2. 1/4 elapsed time          1.96       1.96
     4. Braking feet from 80      1.56       1.56
     8. Fuel mpg                  1.51       1.51
     7. Interior noise db        -0.22      -0.22    2
     6. Skidpad g factor         -1.10      -1.10
     5. Slalom speed             -1.14      -1.14
     3. Top speed                -1.66      -1.66
```

Groups 1 and 3 seem to be more extreme on nearly every variable than Group 2, which has a more central position. Note that Group 1 is characterized as having the highest positive deviation from the grand mean on Skidpad g factor (deviation +0.96), followed by Interior noise (deviation +0.92), Top speed (deviation +0.72), and Slalom speed (deviation +0.61). No other group shows a more extreme deviation from the mean in the same (positive) direction as group 1 on these four variables. Similarly, no other group is as extreme in the negative direction on the next four variables, on which Group 1 is positioned below the grand mean.

After the last cluster solution is printed (the 5-group solution in our case), each pair of solutions is automatically cross-tabulated. For example, the tabulation of adjacent pairs of solutions is as follows:

```
Tabulation of 2 group vs. 3 group solutions

          0     1     2     3   Total
       ----  ----  ----  ----    ----
  0 |     0     0     0     0 |     0
  1 |     0    18    24     0 |    42
  2 |     0     0    14     6 |    20
       ----  ----  ----  ----    ----
Total     0    18    38     6 |    62

Tabulation of 3 group vs. 4 group solutions

          0     1     2     3     4   Total
       ----  ----  ----  ----  ----    ----
  0 |     0     0     0     0     0 |     0
  1 |     0     0     1    13     4 |    18
  2 |     0     0    19     0    19 |    38
  3 |     0     6     0     0     0 |     6
       ----  ----  ----  ----  ----    ----
Total     0     6    20    13    23 |    62

Tabulation of 4 group vs. 5 group solutions

          0     1     2     3     4     5   Total
       ----  ----  ----  ----  ----  ----    ----
  0 |     0     0     0     0     0     0 |     0
  1 |     0     0     0     0     3     3 |     6
  2 |     0     5     0    15     0     0 |    20
  3 |     0     0    12     1     0     0 |    13
  4 |     0    23     0     0     0     0 |    23
       ----  ----  ----  ----  ----  ----    ----
Total     0    28    12    16     3     3 |    62
```

In the tabulation of the two- vs. the three-group solution, the first row and column of the table reflects group 0 (any outliers). No cases were regarded as outliers since we didn't permit outliers--we had checked *Include all cases (no outliers)* on the *Settings* tab.

The 42 cases in group 1 of the 2-group solution were split between groups 1 and 2 of the 3-group solution. The 20 cases in group 2 of the 2-group solution were split into groups 2 and 3 of the 3-group solution.

This example was purely for illustrative purposes. Your data sets will typically have many more cases. In social sciences and marketing research, you usually wouldn't want to draw conclusions regarding groups containing just a few cases. Even so, the question arises whether there seems to be reproducible and natural cluster structure in this car data set. The Monte Carlo runs in the section of this documentation entitled Reproducibility Norms provide benchmarks for assessing whether the structure observed in your data is more organized than what could be expected using random data with no cluster structure.

Below, we've compared the adjusted reproducibility we achieved with this data run to that which could be observed using random data constructed with no cluster structure (as published in Table 1 in the section entitled Reproducibility Norms):

```
Adjusted Reproducibility

  2-group solution:
    Cars Data     63%
    Random Data   58%

  3-group solution:
    Cars Data     79%
    Random Data   46%
```

```
4-group solution:
  Cars Data     82%
  Random Data   43%

5-group solution:
  Cars Data     68%
  Random Data   47%
```

For each cluster solution, the adjusted reproducibility achieved with the cars data set exceeded that expected from a data set (using 10 basis variables) with no group structure. While this certainly is good news, it is a very low standard of achievement.

The 3- and 4-group solutions have in absolute terms the highest reproducibility, and they also display the largest gap between observed reproducibility and the benchmark based on no group structure. The 2-group solution seems to be a poor characterization of these data, barely exceeding the "no structure" threshold in terms of adjusted reproducibility.

4 Cluster Ensemble Analysis

4.1 Cluster Ensemble Analysis

Cluster Ensemble approaches (Strehl and Ghosh 2002, Retzer and Shan 2007, Orme and Johnson 2008) employ multiple cluster solutions as well, but rather than choose the *one* most representative solution, they develop a consensus solution based on a combination of the solutions available within the ensemble. The final solution is almost always different from all of the solutions in the ensemble. Ensemble Analysis benefits from a diverse set of cluster solutions, such as from different cluster methodologies (e.g. hierarchical, k-means, neural networks, etc.), different basis variables, and different numbers of clusters This is made possible by the fact that Ensemble Analysis does not "look at" the original data, but rather examines only the assignments of individuals to clusters. The consensus solution combines information from those many partitionings to find one which is most representative of them all.

For nearly three decades, we have advocated using k-means clustering rather than hierarchical clustering methods. Our opinion has not changed, as we feel that k-means clustering (from multiple, intelligently-drawn starting points) generally is more robust under more conditions than hierarchical methods. However, when employing cluster ensembles, the literature suggests that ensembles benefit from including solutions representing a variety of methods that involve different inductive biases. Therefore, CCEA includes the capability of developing clusters within the ensemble via hierarchical (complete and average linkage) methods. Given our bias towards the k-means methodology, the default setting for our implementation of Cluster Ensemble Analysis provides more k-means solutions within the ensemble than hierarchical.

We should warn you that the hierarchical clustering methods require significant memory resources of the computer, since an n x n matrix of similarities must be constructed, where n is the number of cases. For its hierarchical solutions, we've found CCEA to have exceptional performance up to about 2,000 cases (runtimes typically about 2 minutes or less, for solutions exploring 2 to 6 groups). With even larger samples, runtimes become much slower, and the computer will eventually run out of memory. (Note: the k-means routines are extremely fast, even for very large datasets.)

We encourage researchers to append additional solutions obtained from other reliable sources within the ensemble file. We do not claim that our particular choice of k-means and hierarchical methods is optimal, and it is likely that we'll offer additional clustering methods within ensemble construction in future versions of the software.

A Direct Consensus Method Using "Clustering on Clusters"

In Strehl and Ghosh's 2002 article, the authors discuss multiple approaches for developing a consensus solution, given the availability of multiple segmentation solutions within an ensemble. Strehl and Ghosh use a method they call a *Meta-Clustering Algorithm*, based on the notion of "clustering clusters."

With the Meta-Clustering Algorithm, one develops multiple clustering solutions. These could vary in terms of:

- Method used (hierarchical, k-means under different starting points, etc.)
- Number of dimensions (for example, selected from 2 to 12 groups)
- Basis variables employed
- Pre-processing options (standardization, centering)

The group assignments for multiple cluster solutions (just three in this example) could look like the following when recorded in a data file:

Caseid	Solution#1	Solution#2	Solution#3
1001	1	4	2
1002	2	2	1
1003	2	3	1
1004	1	4	2

Solutions #1 and #3 are 2-group solutions, and across the first four cases they appear to be identical (except that the labels are switched). Solution #2 is a 4-group solution.

It is very easy to modify this file to have "indicator" (dummy) coding. Strehl and Ghosh code the information for a 2-group solution (such as Solution #1) using two columns, where the first column indicates whether the respondent belongs to the first group and the second column indicates membership in the second group.

Indicator Coding for Solution #1:

1001	1	0
1002	0	1
1003	0	1
1004	1	0

All three solutions in the example above could be coded in eight total indicator columns, or:

Indicator Coding for Solutions 1-3:

1001	1	0	0	0	0	1	0	1
1002	0	1	0	1	0	0	1	0
1003	0	1	0	0	1	0	1	0
1004	1	0	0	0	0	1	0	1

Strehl and Ghosh employ a method that involves repeatedly clustering (using a graph partitioning approach) and relabeling the clusterers, so that cluster #1 from the first solution corresponds to cluster #1 from the second solution, etc. This becomes a challenging optimization problem when many groups are included across many replicates, and with somewhat noisy datasets as would be found in practice.

We use Strehl and Gosh's first step, but have chosen to side-step the issue of relabeling altogether by simply clustering again on the indicator matrix (clustering on the cluster solutions, or "CC") without worrying about relabeling. In the example above, we simply use these eight columns as new basis variables in a secondary cluster analysis, where we are looking for a final k-group solution (and the indicator variables could represent cluster solutions with either more or fewer clusters than the final k-group solution we seek). For our work, we leveraged CCA's standard approach of running multiple replicates (30) under k-means (using different, intelligently drawn starting points) and we selected the one solution that was most reproducible as a possible final solution and candidate stopping point. We have found it useful to include a large number of cluster solutions in the ensemble, representing a wide variety of numbers of clusters. There doesn't seem to be any harm (overfitting) in including a very large number of runs in the ensemble. We have had good results using sixty or seventy cluster solutions in the ensemble, ranging from 2-group solutions clear up to 30-group solutions. And, we find the final clustering result is more stable (when employing different starting point seeds) if using large, diverse ensembles. Our software implementation seems very fast, with an ensemble analysis as just described typically requiring only about 30 seconds for 1000 respondents.

If multiple solutions are obtained by "clustering on cluster solutions (CC)", one can compute reproducibility across those replicates (we employ 30 replicates) to ascertain how consistently one obtains the same result from different starting points. We might also consider the most reproducible of these as the best solution; however, it is not strictly necessary to introduce the notion of reproducibility. We can recode those replicates (now all on k-groups) using indicator coding and repeat the process

(clustering on cluster solutions of cluster solutions (CCC)). This loop can continue indefinitely (CCC…C), but we find that the process converges very quickly, usually within 1 to 3 steps. When no respondents are reclassified in a subsequent step, we may take the previous candidate solution (the most reproducible one) as final. As far as we know, our approach is unique, though it owes a great deal to the notions set forth by Strehl and Ghosh.

The literature suggests that cluster ensembles which use diverse clusterers will be more robust to characteristics in the data that do not conform well to traditional k-means, such as elongated clusters. Even though we use k-means as our method to develop a consensus solution from the indicator coding matrix, the cluster solutions in our ensemble include hierarchical methods that add diversity and can yield more flexible final clusterings. However, our approach to ensemble construction and creating a consensus solution is based on the notion that clusters should be generally compact. For that reason, we have not employed single-linkage hierarchical clustering in the "clustering on clusters" consensus step. Therefore, our implementation should not be expected to work very well in recovering the sorts of artificial structures (spirals, rings, etc.) that other authors have used as a standard for prediction. But our approach should work well in detecting meaningful structure more commonly found in market and social research. And, if desired, one could use single-linkage hierarchical clustering to develop the consensus solution (rather than k-means), and this should do a creditable job of capturing data with very elongated or patterned structures.

How Well Does It Work?

We have compared the standard CCA (Convergent Cluster Analysis) approach of using k-means with 30 replicates (and choosing the most reproducible replicate) to our implementation of Ensemble Analysis across many data sets. The results are described in a white paper entitled, "Improving K-Means Cluster Analysis: Ensemble Analysis instead of Highest Reproducibility Replicates," available for downloading from our Technical Papers library at www.sawtoothsoftware.com. We drew comparisons based on over one-dozen comparisons on data sets between CCA and Ensemble Analysis. Here is one representative example of the performance edge for Ensemble Analysis as reported in that paper:

Comparative Test:

For this test, we developed an artificial dataset with true means and group sizes as follows:

```
                  True Group Means:
Group 1 (n=300):  6  4  4  1 10  4  6  1  7  1
Group 2 (n= 50):  4  5  8  5  5  8  7  3  5  2
Group 3 (n=100): 10  4  4  2  5 10  7  3  4  8
Group 4 (n=200):  5  2  2  8  8  5  2  4  3  1
Group 5 (n=150):  2  3  4  9  2  5  5 10  4 10
Group 6 (n=200):  2  5 10  6  7 10  9  9  3  4
```

We created five separate datasets for this test, disturbing the data by normal random error with standard deviation of 1, 2, 3, 4 or 5.

Hit rates (correct classification rates to known groups) by level of error disturbance were:

```
              CCA      Ensemble
Error = 1   100.0%      100.0%
Error = 2    99.0        99.1
Error = 3    89.1        90.0
Error = 4    73.4        76.1
Error = 5    62.3        70.0
```

Conclusions:

After examining over one-dozen data sets, our conclusions were as follows:

> "Our implementation of ensemble analysis generally performs better than CCA's approach of choosing the most reproducible replicate. The ensemble approach seems especially useful when the true sizes of the groups are quite different (which is often true in practice) and when groups have differing degrees of overlap with respect to each other on the basis variables (again more likely in practice). In those cases, it achieves significantly better hit rates, better fit to true group means, and better estimates of the true group sizes. With equal-sized groups that are completely unique with respect to their means on the basis variables, it seems to perform just as well as CCA's approach. Like CCA, our ensemble method provides a measure of reproducibility, which can be used to help determine how many groups provide a good characterization of the data structure. The reproducibility statistic for our ensemble method seems to perform just as well or better than the similar statistic in CCA for indicating the correct number of groups."
>
> "We haven't evaluated other methods of forming consensus solutions for ensembles, and thus cannot comment on the relative performance of our method versus others described in the literature. This remains an avenue for future research."

Usage Hints:

Some users may wish to build ensembles that leverage cluster solutions developed using different basis variables. For example, some cluster solutions may be based on psychographic profiling variables, and other solutions may be based on preference/usage variables.

If you are interested in seeing the results of the clustering runs included in the ensemble, CCEA saves the group memberships for each clustering within the ensemble to a .csv file. You can edit this file to include additional segmentation runs and submit the modified ensemble to CCEA to use in producing a new consensus solution.

References:

Orme, B. and R. Johnson (2008), "Improving K-Means Cluster Analysis: Ensemble Analysis instead of Highest Reproducibility Replicates," available at www.sawtoothsoftware.com/techpap.shtml.

Retzer, J. and M. Shan (2007), "Cluster Ensemble Analysis and Graphical Depiction of Cluster Partitions," Proceedings of the 2007 Sawtooth Software Conference, Sequim WA.

Strehl, A. and J. Ghosh (2002), "Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions," *Journal on Machine Learning Research (JMLR)*, 3:583-617, December 2002.

5 Suggestions for Use

5.1 Preprocessing Options

CCEA offers two preprocessing options: you can *standardize* variables or *center* objects (respondents). Following are descriptions and advice about when to choose each one.

Standardizing Variables

This operation transforms the data for each variable to have mean of zero and variance of unity. The transformation is done by first computing the mean and variance for each variable in its "raw" form across all respondents. Then the mean is subtracted from each observation, and the difference is divided by the square root of the variance.

With social science or marketing data it is often a good idea to standardize your data. Standardization might be inappropriate if all your variables are measured on similar scales, and if whatever differences in their scaling reflect their relative importances. However, in the social sciences most measurement scales are arbitrary, and it is customary to remove the effects of arbitrary scaling by standardizing variables.

It may be preferable not to standardize variables derived from conjoint data. Conjoint utilities and importances are already scaled meaningfully with respect to each other; indeed, much of the value of conjoint analysis lies in the relative importances of different attributes, which is information contained in their relative scaling.

If you do not standardize your variables, you implicitly weight them unequally. Cluster analysis pays more attention to variables with larger variances. For example, if you were clustering people in terms of anthropometric measurements, you might have chosen to measure height in feet, inches, or millimeters. If you standardize those variables it is immaterial which unit you chose. If you do not, a choice of millimeters rather than feet will mean that the actual values given to cluster analysis are much larger, and the height variable will be given much more weight in the analysis.

As another example, in the "car" data distributed with CCEA, price is measured in thousands of dollars. If prices were measured in dollars the numbers would be much larger and clustering those data without standardizing variables would result in almost all of the weight being applied to the price variable.

Centering

This operation computes the mean for each respondent over all the variables, and then subtracts it from all of them so that the new mean for each respondent is zero. If both standardizing and centering are requested, the standardization is done first and then the standardized data are centered.

This operation is usually a good idea when clustering survey respondents except when clustering conjoint utilities or importances, which already have nearly constant sums for each respondent.

It has been widely observed that survey respondents differ systematically in their propensity to answer rating scale questions with high vs. low values. Various names have been given to this phenomenon, including "yea-saying tendency" and "response style."

Suppose data are from 30 questions about the importance of various product benefits. Some respondents will give high ratings to all benefits, and others will give low ratings to all. A two-cluster solution for those data will probably divide people into "high raters" and "low raters." That may or may

not be information that you want to capture.

On the other hand, the product benefits may fall naturally into classes like "performance" and "economy." If you center your data, then the two-cluster solution would probably divide people into more relevant clusters, one concerned with "performance" and the other with "economy."

5.2 Suggestions about Steps to Follow

Most of this manual is concerned with facts—with details about the operation of the CCEA software. This section is based more on opinion. This section is provided for those unfamiliar with cluster analysis or the special features of CCEA. You should keep in mind that there are many ways to do cluster analysis, that experts disagree, and that the procedures we suggest are not necessarily the same as those that others would advocate.

Questionnaire Design: If you are going to cluster survey respondents, it's important to have that in mind in the early stages of questionnaire design. Survey respondents tend to differ from one another in their propensity to provide high vs. low ratings. Ordinarily, these differences are unrelated to the subject matter being studied, and they represent a source of "noise" in the data.

However, sometimes the overall level of a respondent's ratings can contain important information, such as when products are being rated on attributes.

It's a good idea to design your questionnaire so that respondents' levels of enthusiasm for whatever is being rated are not confounded with their general tendency to give high or low ratings. One good way to do this is to balance the questionnaire items so that about half are "positive" and half are "negative." If you're doing political attitude work, balance an item such as "I'm a conservative" with another such as "I'm a liberal." This way you'll be able to use the "centering" option to remove "noise" from the data without taking "signal" with it.

The question often arises whether to cluster on conjoint importances or (normalized) conjoint utilities. Clustering on conjoint importances makes for more parsimonious solutions, since fewer variables are involved. However, if some of your attributes are categorical (such as brand or color), then meaningful differences between respondents in terms of their specific preferences for levels will not be used to determine segments. One possible solution is to use utilities for attributes such as brand, but importances for ordered attributes like speed and price. Because utilities and importances are measured on different scales, you would want to standardize the data prior to clustering.

CCEA gives you the capability of testing reproducibility by seeing if solutions from various starting points are similar; but doing exploratory runs with subsamples lets you also test reproducibility by seeing whether you get groupings based on the entire sample similar to those obtained from subsamples. If you have more than, say, 1000 respondents, we suggest that you do most of your clustering on just a random half of the sample, and that you reserve the other half for confirmation of what you learned from the first half. This will let you be more confident of your final results.

You can identify outliers using CCEA's cluster analysis routine. However, CCEA's ensemble analysis assumes all respondents should be placed within clusters. Therefore, if you wish to identify and exclude outliers, it should be done based on earlier cluster analysis runs within CCEA. The cluster runs you prepare within your ensemble should assign all respondents to clusters.

Number of Variables: CCEA can do clustering on up to 1000 variables. We have provided that large a limit because modern PCs are able to do it (not that we think the limits should be pressed!); but you should almost always use fewer variables. Rather than trying to cluster on them all, you might use factor analysis to try to learn more about their structure, and then you can select a subset for clustering. If the factor analysis indicates 10 factors, for example, you might want to include just three or four variables from each factor.

We also recommend that you cluster on answers to individual questions rather than on factor scores. The averaging that is involved in computing factor scores can smooth things out, and may remove some of the "lumpiness" and richness of the data that cluster analysis depends on.

The First Runs:

We suggest that the first runs be done using the standard cluster analysis within CCEA prior to

employing cluster ensembles to develop a final solution. The first runs should be based on no more than about one thousand respondents, and no more than about 30 variables. If the data are ratings from survey respondents, select the centering option. If the data involve descriptive variables measured on quite different scales, then use the standardization option. If the data are conjoint utilities or importances, we suggest that you not standardize or center. Centering would have little effect since the utilities from ACA's "Points" and "Diffs" output already have constant sums for each respondent, as do conjoint importances. It is usually not appropriate to standardize conjoint variables, since conjoint data are already scaled meaningfully. Ask for all cluster solutions, from 2 through 10. Choose 10 replications for each solution. (You will have the opportunity later to use a larger number for maximum security.)

When the first run is finished you can get a very quick idea of what happened by printing out the report file. The report file shows the average reproducibility and the pooled F ratio for each clustering, and how many individuals were in each cluster. Normally, the average reproducibility and pooled F ratio each diminish as the number of clusters increases. But you may find a particular solution where they increase rather than decrease from the previous solution. (We suggest you plot the values to discover the degree of departure from a decreasing curve.) With artificial data where we know how many clusters there really are, such a pattern often identifies the "correct" solution. With real data, where there is no "correct" answer, such a pattern often indicates a solution worth exploring further.

If you have not used the outlier option or the minimum desired cluster size option, you may find that, as the number of clusters increases, some of them get very small. Sometimes one or more will contain only a single respondent. That may mean that you have gone too far, and you should consider solutions with fewer clusters. However, if you use the outlier option or the minimum desired cluster size option, you may be able to find a meaningful and reproducible solution with a larger number of clusters.

Having formed an idea of which solutions look most promising, you can turn to each cluster's means on the variables. This information is arranged so that you can tell quickly whether a particular clustering "makes sense." The tables showing the variables on which each cluster has outstandingly high and low averages are most useful for getting a quick impression. If you look at the F ratios for the variables, you can see which variables have been most important in defining these clusters, and which have been less so.

Using the Entire Sample and Cluster Ensembles:

Eventually, you will develop an idea about the number of clusters that would be most useful to interpret. When you have gone as far as you can with the subset of respondents under cluster analysis, you should plan a run with the entire sample using cluster ensemble analysis. By this time you should have a good idea of what to expect in terms of cluster structure.

Now you might restrict the range of solutions to those you think are likely to be useful, with perhaps one more "on each side." For example, if you think the solutions with either five or six clusters are likely to be best, you might ask for the range of four through seven. You're near the end of your analysis now, so you want to be sure that what you get is the best that CCEA can provide: Cluster Ensemble Analysis.

When you're through you should be sure that your solution meets all these criteria:

1. It has acceptable reproducibility. Consult the tables provided in the section entitled Reproducibility Norms to see what you could expect due to chance alone.

2. The answer you got with a subsample should be similar to the answer you got with the entire sample.

3. The clusters should make sense to you.

4. The clusters should differ significantly and in meaningful ways on other variables which were not used in clustering (use your separate cross-tabulation software for examining this issue). For example, if you clustered on "benefit segments," the groups that you got should differ on product consumption and brand usage. The cluster that said "economy" was most important should be expected to use more economical products, etc.

6 Reproducibility

6.1 Reproducibility

One of the problems with cluster analysis methods is that they always produce clusters, whether they "really should" or not. Furthermore, our human tendency to find meaning in even the most random of patterns makes it likely that the analyst will find an interesting interpretation of almost any cluster solution.

Suppose there were two variables, x and y, that were truly independent of one another with equal variances. Then, if we were to plot 1000 objects in that two-space, we would expect to find a more or less circular swarm of points.

If we were to subject those data to cluster analysis, we would most assuredly get clusters. If we asked for a two-cluster solution the points might be divided into those on the left vs. those on the right, or those on the top vs. those on the bottom. Or the partitioning might distinguish between those more toward the Northwest and those more toward the Southeast. In any case those in each cluster would be "different" from those in the other. If we made the error of applying tests of significance we would get very large statistics, since the distributions would have no overlap whatsoever.

The problem is that there would be no reason for preferring any of those possible partitionings to any other. The analyst who accepted one of them as "true" would be very likely to face disappointment if he ever repeated his analysis with a slightly different data set.

The data that most researchers subject to cluster analysis may be richer than the hypothetical situation just described, but the same pitfall awaits any user of cluster analysis techniques. There appear to be only two ways to avoid being misled. First, the results must "make sense." This should be a necessary criterion for accepting a cluster analysis solution, but it is by no means sufficient.

Second, the solution must be demonstrated to be "reproducible." Reproducibility can be measured in at least two ways.

First, since it is known that the solution can depend on the starting points, it is important to try different starting points to see if they all produce similar solutions. Reproducibility can be measured by determining to what extent the same objects are grouped together in each solution. CCEA provides the capability of automatically repeating each clustering from as many as 30 different sets of starting points, measuring the reproducibility, and reporting the most reproducible of the solutions obtained.

Second, it is useful to divide the objects randomly into subsets and to cluster each subset separately. In this case reproducibility can be measured by seeing whether the groups can be matched in terms of their profiles of means on the variables. For this purpose it is useful to correlate cluster means across attributes. Corresponding groups should be approximately the same relative sizes. CCEA does not provide the capability of conducting split-sample clustering automatically, but it is a precaution that the prudent researcher should take. With large enough sample sizes, we'd recommend clustering using just the first half of the respondents, and then comparing the results to clustering based on the other half of the data set.

CCEA uses a particularly simple measure of reproducibility. Suppose two sets of starting points were used to produce three-cluster solutions, and the solutions were tabulated against one another as follows:

	Solution 2			
	Group 1	Group 2	Group 3	Total
Solution 1				
Group 1	10	13	311	334
Group 2	321	13	4	338
Group 3	5	307	16	328
Total	336	333	331	1000

In this example a cluster from each solution seems to correspond with one in the other solution, but similarly labeled groups do not match. We can simplify the picture by permuting pairs of columns so as to maximize the sum of numbers appearing in the diagonal. By exchanging columns 1 and 2 and then exchanging columns 1 and 3 we get the following:

	Solution 2			
	Group 1	Group 2	Group 3	Total
Solution 1				
Group 1	311	10	13	334
Group 2	4	321	13	338
Group 3	16	5	307	328
Total	331	336	333	1000

Now we sum the diagonal elements (311 + 321 + 307 = 959) and express that sum as a percent of the total number of objects, or 95.9% reproducibility.

A moment's reflection will show that this figure is sensitive to the relative sizes of the groups in the two clusterings. If they differ from one another, reproducibility cannot reach 100%.

When using CCEA in standard cluster analysis mode you are asked how many replicate clusterings you want to have. Each pair of clusterings is subjected to a pairwise reproducibility analysis like that above. The resulting percentages are arrayed in a table; if you ask for 5 replications, the table is of size 5 x 5. The columns of the table are summed to get the average of each replicate's reproducibility with all the other replicates. The replicate with the highest average reproducibility is chosen to be reported. When using cluster ensemble analysis (consensus solution), reproducibility is also reported, representing reproducibility obtained across 30 replicates of k-means on the initial dummy-coded ensemble matrix.

Adjusted Reproducibility:

In the first version of CCA, we reported reproducibility numbers like the example just described. However, that way of measuring reproducibility had a shortcoming which later was corrected in v2: it unfairly penalized solutions with larger numbers of clusters. Suppose clusterings were entirely random. Then if there were only two clusters, one would expect to see reproducibility of 50% just due to chance. At the other extreme, with 10 clusters, one would expect only 10% of respondents to be clustered identically due to chance.

As with v2 of CCA, CCEA reports "adjusted" reproducibility values which permit more meaningful comparisons when the solutions being compared have different numbers of clusters. Let:

r = the unadjusted reproducibility proportion
k = the number of clusters
a = the adjusted reproducibility value

Then the formula for adjusted reproducibility is

$$a = \frac{k*r - 1}{k - 1}$$

We derive it as follows:

1) Comparing two solutions, the proportion of respondents classified differently is 1 - r.

2) If clustering is only random, the probability of two respondents being classified differently is (k - 1) / k.

3) Thus the relative unreproducibility compared to a chance level, would be
relative error = (1 - r) / [(k - 1) / k].

4) And finally the adjusted reproducibility is 1- the relative error, or

adjusted reproducibility = 1 - (1 - r) / [(k - 1) / k]

which reduces to the formula above.

Since v2 of CCA, we have reported adjusted reproducibility values rather than the unadjusted values reported by the first version. Since adjusted values are somewhat lower, researchers accustomed to seeing unadjusted values may be concerned that their solutions are not as good as before. The differences are probably just due to the change in the way reproducibilities are reported. Following is a table that shows adjusted reproducibility values for many combinations of unadjusted values and numbers of clusters.

ADJUSTED REPRODUCIBILITY INDICES

Number of	Raw Reproducibility										
Clusters	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
2	0.00	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
3	0.25	0.33	0.40	0.48	0.55	0.63	0.70	0.77	0.85	0.92	1.00
4	0.33	0.40	0.47	0.53	0.60	0.67	0.73	0.80	0.87	0.93	1.00
5	0.38	0.44	0.50	0.56	0.63	0.69	0.75	0.81	0.88	0.94	1.00
6	0.40	0.46	0.52	0.58	0.64	0.70	0.76	0.82	0.88	0.94	1.00

6.2 Reproducibility Norms for Cluster Analysis

It may be confidence-inspiring to know that a cluster solution has "reproducibility of 95.9" but that figure isn't of much value without some standard of comparison. We have attempted to provide standards by performing a Monte Carlo simulation. We generated artificial data sets containing different numbers of variables (2, 5, and 10), different numbers of clusters (2, 4, and 6,) and different amounts of expected separation between clusters (1, 2, and 3).

Each cluster center was constructed by creating a cluster centroid vector of random normal numbers with means of zero and standard deviations of 1, and then multiplying the vector by a scale factor of either 1, 2, or 3.

Object points were determined by first randomly selecting a cluster, and then adding to that cluster's centroid vector another vector of normal random numbers with mean of zero and standard deviation of unity. An average of 50 objects were created for each cluster.

Each object point was tested to ensure that it was closer to its own cluster center than to any other. If not, the point was rejected and another computed.

Thus, the variables were independent and of equal variance within clusters. The clusters had expected separations of approximately 1, 2, or 3 times the within-cluster standard deviation, although the amount of actual separation was allowed to vary randomly.

With three numbers of variables, three numbers of clusters, and three amounts of expected cluster separation, there were 3x3x3 = 27 different treatments. Twenty separate data sets were generated for each treatment, for a total of 540 data sets.

Sixty "single cluster" data sets were also created, each consisting of 200 observations distributed around a single "true" cluster centroid. This was done for numbers of variables equal 2, 5, and 10, with 20 data sets for each.

Each data set was analyzed to obtain the "best" solution for each number of clusters from two to eight. Each of those 7 clusterings was repeated 5 times from different starting points, obtained by the "Density-based" method. Thus, the total number of clusterings in this simulation was 600 x 7 x 5 = 21,000.

We first show the results for the case of "one true cluster." Table 1 shows results for analyses where the data sets had no true cluster structure whatsoever, being single multinormal point swarms.

Table 1
Average Adjusted Reproducibility
(Number of True Clusters = 1)

	Number of Clusters in Solution						
	2	3	4	5	6	7	8
#Variables							
2	64	84	68	64	68	67	68
5	80	61	65	58	59	57	57
10	58	46	43	47	45	46	43
Average	67	64	59	56	57	57	56

Perhaps surprisingly, the algorithm is often able to find substantially similar solutions from different random starting points even when there is no true cluster structure at all. This is doubtless due to random irregularities in the data. If the pattern of points were perfectly circular in shape and normal in density, it is doubtful that reproducibility would be this high.

Reproducibility is greatest when only two clusters are sought, and decreases as larger numbers of clusters are sought. It appears also that larger numbers of variables generally provide decreased

reproducibility, at least in the case of "no true" cluster structure.

This table provides a baseline for interpreting reproducibility figures obtained from any particular solution. For example, the case cited about the "95.6% reproducibility" is comfortably above the levels to be expected when the data have no true cluster structure.

We turn now to the case of data sets that do contain cluster structure. Table 2 shows results for two "true" clusters, with average separation of 1, 2, or 3 units. (A unit is the within-cluster standard deviation.)

Table 2
Average Adjusted Reproducibility
(Number of True Clusters = 2)

Cluster Separation	# Variables	Number of Clusters in Solution 2	3	4	5	6	7	8
1	2	98	64	68	71	74	72	71
	5	100	60	56	54	57	56	57
	10	100	69	59	55	57	52	53
2	2	100	69	64	66	66	69	74
	5	100	69	64	61	56	56	53
	10	100	69	61	54	54	53	54
3	2	100	64	64	69	71	72	69
	5	100	79	67	63	60	60	57
	10	100	69	61	52	53	51	54
Average		100	68	63	61	61	60	60

Here we see substantially higher reproducibilities in the column for the two-cluster solution. This is the expected (and desired) result, indicating that the reproducibility figure is sensitive to the true cluster structure. Again, the reproducibilities tend to be lower with larger numbers of variables. Note that they are not dramatically higher, if at all, for clusters with greater separation.

Table 3
Average Adjusted Reproducibility
(Number of True Clusters = 4)

Cluster Separation	# Variables	Number of Clusters in Solution 2	3	4	5	6	7	8
1	2	62	67	85	83	82	76	70
	5	56	94	85	76	65	62	60
	10	96	76	93	80	69	60	60
2	2	64	99	87	74	69	70	70
	5	90	90	93	83	76	66	62
	10	76	72	96	85	72	69	63
3	2	72	100	100	79	71	77	77
	5	88	78	95	83	72	66	63
	10	70	72	93	84	72	69	62
Average		75	83	92	81	72	68	65

Here the average reproducibility for four-cluster solutions is much higher than in the previous table, and average reproducibility for two-cluster solutions is much lower. Most other figures are higher, probably indicating that the data are more richly structured in general.

Table 4 shows results for six "true" clusters.

Table 4
Average Adjusted Reproducibility
(Number of True Clusters = 6)

Cluster Separation	# Variables	Number of Clusters in Solution 2	3	4	5	6	7	8
1	2	100	81	81	79	71	85	77
	5	94	85	92	75	82	80	71
	10	66	85	83	84	87	80	77
2	2	98	99	96	76	75	71	75
	5	68	79	99	81	92	84	78
	10	58	85	80	78	92	85	79
3	2	100	85	99	83	86	83	75
	5	46	69	87	90	90	88	79
	10	64	73	79	79	89	86	81
Average		77	82	88	80	85	82	77

The average reproducibility for six-cluster solutions is higher than in any previous table, but, unfortunately, it is not the highest in this table; the average for 4-cluster solutions is higher. Further, the average reproducibilities for several other numbers of clusters are nearly as high. These are disappointing results, suggesting that reproducibility values may not be of much value in detecting the "true" number of clusters when that is greater than about four.

The main contributions of these tables are:

1. Reinforcement of the notion that cluster analysis is able to produce apparent structure even where none is truly present.

2. To provide guidelines about the level of reproducibility that can be expected just due to chance.

3. To help in the decision of which cluster solution to choose, in the case of fewer than about six clusters.

Reproducibility values are a way of measuring reliability of a clustering method, but are not necessarily related to validity. A method that gave the same wrong answer time after time would be perfectly reproducible, but also completely invalid.

Reproducibility can be measured whenever clustering is done. However, measurement of validity presumes that we know what the right answer is. In these simulations we had knowledge of the true cluster structure. We would only need to see whether the clusters actually recovered are the same ones that were used to generate the data.

Unfortunately, that examination was not made in the large simulation just reported. However, additional data sets were constructed and analyzed for this purpose. Again, there were 2, 5, or 10 variables, 2, 4, or 6 clusters, and cluster separations of 1, 2, and 3 units. Eight data sets were produced for each of these 27 cells, each analysis with 5 replications from different starting points, for a total of 1080 clusterings.

In this analysis we only examined the solutions corresponding to the "true" number of clusters. We measured reproducibility in the same way as described above, but also measured the agreement of the solution with what we knew to be the "true" solution.

Because the results are based on many fewer cases than those for reproducibility shown above, we present them only as averages. Table 5 shows average reproducibilities as well as recoveries of the true cluster solution for each of the main factors in our design.

Table 5
Reproducibility and Recovery of True Cluster Structure

		Average Reproducibility	Average Recovery of True Structure
Number of Clusters	2	99%	99%
	4	88%	90%
	6	91%	93%
Cluster Separation	1	88%	88%
	2	93%	94%
	3	91%	93%
Number of Variables	2	89%	88%
	5	90%	92%
	10	92%	94%

(These results are expressed in terms of unadjusted rather than adjusted reproducibility and recovery percentages.)

Both reproducibility and recovery of the true structure tend to decrease with the number of clusters, and to increase with both cluster separation and with the number of variables. The most encouraging feature of this table, however, is that the second column is moderately larger than the first. The solution involving the "true" number of clusters seems to recover the right answer to about the same extent as it reproduces the same answer in multiple attempts.

6.3 Reproducibility Norms for Ensemble Analysis

As when using CCEA in its cluster analysis mode, when using Ensemble Analysis (consensus solution), a reproducibility statistic is provided. In this case, reproducibility is determined based on the first clustering step of the dummy-coded ensemble (30 replicates of k-means clustering, using a mixed starting point strategy). Reproducibility is a helpful statistic for selecting how many clusters seems to characterize the data well. Naturally, the question arises: "What is a good reproducibility number?" Certainly, we would want to obtain higher reproducibility than could be obtained from random data, where respondents were distributed in a single cloud (rather than segments).

We generated a random data set where respondents were distributed as a single group with random error. The reproducibility obtained for Ensemble Analysis* is shown in the table below. If the reproducibility you obtain is not much higher than the figure below (corresponding to a particular number of segments and number of basis variables), then it suggests that your data don't have very good cluster structure.

Ensemble Analysis Adjusted Reproducibility for Random Data
Number of True Clusters=1

	Number of Clusters in Solution									
	2	3	4	5	6	7	8	9	10	Average
5 basis variables	86	77	72	77	74	76	76	76	75	77
10 basis variables	80	76	80	76	73	74	75	74	75	76
20 basis variables	74	80	77	76	73	72	69	70	68	73
30 basis variables	67	66	70	67	68	67	67	67	66	67
Average:	77	75	75	74	72	72	72	72	71	73

*Default settings for Ensemble Analysis were used. Multiple draws of the random data were used, and results were averaged across multiple runs to provide greater stability to the figures. Sample size was 1000 simulated respondents.

Our comparisons of standard clustering under CCEA versus the Ensemble Analysis method suggest that the reproducibility statistic from Ensemble Analysis may be more accurate in pointing to the "correct" number of groups in the data. For more information, see: Orme, B. and R. Johnson (2008), "Improving K-Means Cluster Analysis: Ensemble Analysis instead of Highest Reproducibility Replicates," available at www.sawtoothsoftware.com/techpap.shtml.

7 Background and Technical Details

7.1 Alternative Clustering Methods

We are concerned with methods of grouping objects based on their patterns of similarity to one another. There are some obvious ways of doing this that we shall dispose of briefly:

> Objects are frequently grouped on an a priori basis. We might ask survey respondents which cereal they eat most often, and then group them using their answers to just that question. This can be a very useful way to group objects, and is much simpler than the methods we are considering.
>
> We might combine information from two or more variables using graphical methods. Suppose we were investigating office copiers, and had already decided we were interested in only two variables: price and speed. We could plot each copier's position in a two-space on the basis of its price and speed. Then we might be able to see just by visual inspection where the points in the plot fell into "natural" clusters.

By contrast, the methods we are considering are appropriate for objects that differ on many variables simultaneously, and are useful in applications too large or complex to be resolved in the simple ways just described. Three main types of methods have received wide use in marketing and the social sciences.

- Hierarchical Cluster Analysis
- Q Analysis
- Partitioning Methods

We shall consider each method briefly.

Hierarchical Cluster Analysis:

Hierarchical methods start with similarity (or dissimilarity) values for all pairs of objects. If there were only a modest number of objects to be clustered, say 100, we could imagine a table with 100 rows and 100 columns, each entry indicating the similarity between the row object and the column object. At the outset we think of each object as defining its own cluster of size "one."

The algorithm is an extremely simple one:

> 1. Scan the entire (n x n) table to find the two most similar objects.
>
> 2. Combine those objects into a single group of size "two." Do this by deleting the row and column for one of the objects being combined, so the reduced table will have only n-1 rows and columns. Modify values in the row and column for the surviving object to indicate similarities of the other n-2 objects with the newly created group rather than with the former group (in this case, the former object). This may be done in several ways, depending on the choice of algorithm. The most common ways are:
>
> - Each new value is the maximum of the two values it replaces. This produces clusters that are not very compact, since a point can be admitted to a cluster if sufficiently similar to one other member. This is sometimes called the "single linkage" method.
> - Each new value is the minimum of the two values it replaces. This produces clusters that are quite compact, since a point can be admitted to a cluster only if sufficiently similar to every other member. This is sometimes called the "complete linkage method."
> - Each new value is a weighted average of the two values it replaces.

3. Carry out steps 1 and 2 a total of n-1 times. At each stage two groups are combined and the number of groups remaining is reduced by 1. The groupings at each of the last few stages define the solutions for 2, 3, 4, ... etc. clusters.

The hierarchical methods are simple and elegant. They are widely used for clustering relatively small numbers of objects, and their popularity is deserved. However, they do present problems, particularly when dealing with large numbers of objects:

1. The table of similarities between objects can get very large, in some cases taxing the memory of today's PCs.

2. In our experience, and according to the literature, the hierarchical methods tend to be less reproducible than others. Relatively trivial-appearing decisions made early in the clustering can have large effects on the final outcome. This is less of a problem if the data are relatively error-free. However, when the data have a high level of error, such as with individual responses to questionnaire items, the hierarchical methods seem to have difficulty producing similar cluster structures when clustering new samples of objects from the same universe.

3. Once a hierarchical method groups two objects together, they will always remain that way. This means that two objects together in the three-cluster solution, for example, will also be together in the two-cluster solution. Each solution is obtained by combining two groups from the previous solution. There seems to be no obvious reason why the "best" solution with two clusters should be precisely the same as what can be obtained by combining two groups from the "best" solution with three clusters.

For these reasons, we have chosen not to base CCEA on hierarchical clustering methods. (However, hierarchical clustering is offered as an option for providing a "starting solution" to be refined by k-means.) We should note that hierarchical methods are provided in CCEA's ensemble analysis, although the consensus solution is developed using k-means.

Q Analysis:

Factor analysis is a technique with a long history of usefulness for exploring relations among variables. The resulting groups are called "factors" rather than "clusters," but the basic similarity to cluster analysis is compelling. Factor analysis starts with a data table similar to that of cluster analysis: a table with objects on one border and variables on the other, with the numbers in the table containing measures of the objects on the variables. Factor analysis starts by computing similarities among variables (usually correlation coefficients) rather than among objects.

It should be no surprise that over the years several researchers have proposed "turning things around" by computing correlation coefficients between objects rather than variables, and then using factor analysis to obtain groups of objects. This technique has been used by psychologists for many years, and is most often known by the name "Q Analysis." As a way of clustering objects, Q Analysis has some strengths and some weaknesses:

1. One strength is that Q Analysis can easily handle a large number of objects. The data for only one object at a time have to be in computer memory for the main portion of the analysis. This means that Q Analysis can handle a virtually unlimited number of objects.

2. Q Analysis is also convenient. Computer programs for factor analysis are widely available.

3. Another strength is that of reproducibility. Our experience with Q Analysis has shown it to do a creditable job of producing similar solutions when used to analyze similar samples of objects from the same population.

4. On the other hand, many statisticians feel a fundamental discomfort with the method. The

assumptions of factor analysis are harder for some to accept when the process is "turned on its side" and correlations are computed between objects and across variables.

5. A potentially most limiting problem is due to constraints relating the number of clusters that can be recovered to the number of variables analyzed. Q Analysis cannot produce more clusters than the number of variables. We can all imagine two dimensional spaces populated with points that fall into more than two clusters, but Q Analysis is not able to produce such a solution.

For both of these last two reasons, we have chosen not to base CCEA on Q Analysis. Fortunately, the method we have chosen shares the three above-mentioned strengths, without sharing the corresponding weaknesses.

Partitioning Methods:

The method we have chosen belongs to a class with many names:

- Partitioning Methods
- K Means Methods
- Iterative Reclassification Methods
- "Sift and Shift" Methods
- Convergent Methods

Like the hierarchical methods, the algorithm is simple and easy to visualize. However, unlike hierarchical methods, the solution for a particular number of clusters is obtained independently of solutions for other numbers of clusters. The two-cluster and three-cluster solutions need not have much in common.

We start with a table of object by variable values. We must also decide in advance how many clusters we want to have (indicated by the algebraic symbol k). The algorithm has these steps:

1. Using some means yet to be determined, select k "starting points." It is easiest to think about these points as being a random selection of k of the objects to be clustered, though they need not be. We shall have more to say later about choice of these points.

2. Compute the similarity of each object with a hypothetical object located at each of the k starting points. Classify each object as "belonging to" the group associated with the most similar starting point.

For example, suppose we had 1000 objects with scores on two variables. We could use that information to plot the objects as 1000 points on a two-dimensional scatter plot. Suppose we wanted to find three clusters. Then we could choose three points at random as "starting points." We could use a ruler to measure the distance of each of the 1000 object points to each of the three starting points. We would classify each object as (initially) "belonging to" the group associated with the closest starting point.

3. Calculate the means on the variables for each of the newly-formed groups of object points. The means define the "centers of gravity" of the new groups.

4. Replace the starting points with new points corresponding to these new means.

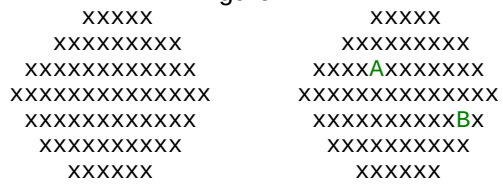
5. Repeat steps 2 - 4 until no object points are reclassified. When the procedure converges in this way, the points in each group define a cluster.

This procedure has an almost "magical" way of converging to reasonable-appearing solutions. Following are "pictures" of the way this procedure converges in only a few iterations, even with

extremely "bad" starting points. (The reader will have to make allowances for imprecision in the location of text characters.)

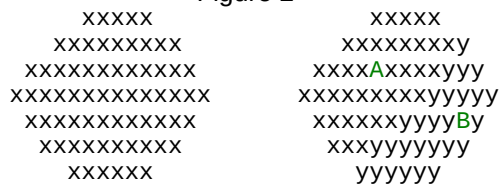
Suppose the two swarms of "x" characters in Figure 1 were two clusters of points in a two-space awaiting discovery. If we want to find the two-cluster solution, we first pick two starting points. As a "bad case," suppose the starting points are at the "A" and "B" in the point swarm on the right.

Figure 1



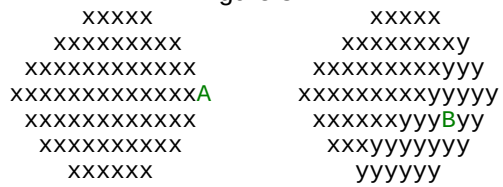
We measure the distance of each "x" to starting points "A" and "B," classifying each "x" into the group associated with the closer of those two. In Figure 2 each point is identified with an "x" or a "y," depending on whether it is closer to "A" or "B."

Figure 2



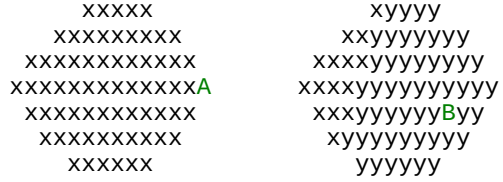
Notice that only the lower right side of the right-hand swarm in Figure 2 is closer to "B" than "A." Now we compute the averages, or "centers of gravity" of all the "x" points and all the "y" points. We indicate those by labels "A" and "B" in Figure 3.

Figure 3



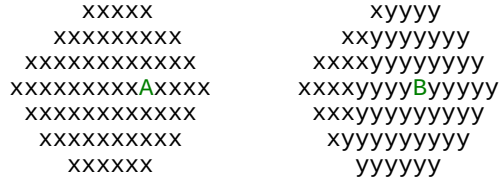
In Figure 4 we have reclassified each point according to whether it is closer to the new "A" or the "B."

Figure 4



Notice that only a miniority of points in the right hand swarm are still closer to the "A" than the "B." Again, we compute the averages of the points now classified as "x" and those classified as "y," indicating those positions by "A" and "B" in Figure 5.

Figure 5



Finally, we would classify as "x" all the points closer to "A" and classify as "y" all points closer to "B."

Since all points on the left would now be identified as "x" and all on the right identified as "y," continuation of this process would result in no further reclassification of points.

This process would have converged even more quickly if our starting points had not been chosen so disadvantageously. For example, if one point had been in the swarm on the left and the other in the swarm on the right, convergence might have been immediate.

It's hard to see how any choice of starting points could lead to failure in this simple example. However, sensitivity to choice of starting point is the most serious problem with this method of cluster analysis. With more complex data structures it is usually found that the final solution depends upon choice of starting point. This means that it is worthwhile to try to find a good set of starting points. It also means that we must be especially careful to make sure that the solution we choose is so compelling that similar solutions would also be obtained when starting from other positions.

Milligan and Cooper, in the review cited above, remark:

> "In summary, the convergent k-means method tended to give the best recovery of cluster structure." (page 341)

They also report that choosing starting points at random is a relatively disadvantageous way to begin. In the next section we describe methods that are available in CCEA for improved choice of starting points.

7.2 Technical Description

Measure of Similarity:

CCEA uses a similarity measure based on Euclidean distances between pairs of points. However, a number of operations could be carried out on the data, either before clustering or as it is in process. Depending on which of those operations may have occurred, the term "Euclidean distance" could mean different things. Under this topic come several sub-issues:

1. How should the data be pre-processed?

- Should we standardize variables to have the same variance, or use the data "as is?"
- Should we weight attributes equally, or give additional weight to those the researcher may consider to be more important?
- Should the data for each object be centered around their mean? (This is often useful with rating scale data from survey respondents, to remove differences due to "response styles.")

Occasions may arise when each of these capabilities can be valuable. CCEA provides the options of pre-processing the data in these ways.

2. Should we combine variables by doing a preliminary factor analysis and then cluster using the factor scores? And how about transforming the variables at each stage so that the data are orthogonal within clusters?

CCEA does not offer the option of doing a preliminary factor analysis and then clustering on factor scores, although there is nothing to prevent the user from doing a factor analysis and then submitting the factor scores to CCEA. It is conceivable that this could be useful, particularly when some way must be found to reduce a very large number of variables.

However, the Central Limit Theorem assures that when variables are grouped into factors a lot of "smoothing" will occur. Cluster analysis can take advantage of the "lumpiness" of data, and will be impeded by any smoothing that takes place.

The same argument can be applied to the issue of orthogonalizing variables. That operation would have advantages if we wished (and if it were appropriate) to make multivariate statistical tests using the data. However, it increases the risk of obscuring the very structure we want to discover, and imposes a substantial computational burden as well. With respect to such additional computations, Milligan and Cooper write:

"Furthermore, method sophistication may have little impact on the quality of the obtained solution. The k-means procedures are fairly direct and simple, whereas (such methods) are rather complex. Yet the methods tended to give equivalent recovery." (page 341)

3. Should we permit the algorithm to choose "automatic" variable weights that give more impact to those variables that contribute more importantly to the clustering?

Some researchers have reported success with methods that automatically give more weight to the variables most useful in differentiating between clusters. This approach should help to suppress variables that do not contribute usefully to the clustering, and it has been demonstrated repeatedly that the presence of "noise" variables can be detrimental to the solution.

However, success with "automatic" weights seems to be limited to "hierarchical" clustering methods (which we consider below). The method used by CCEA is not hierarchical, and the feasibility of automatic weighting for non-hierarchical methods is in doubt. Our experimentation with automatic variable weighting in non-hierarchical clustering suggests that they impede the recovery of known clusters. We are also troubled by the excessive freedom that the clustering algorithm can enjoy in that environment.

Suppose we were searching for two clusters, and that the algorithm were free to give variables weights of zero to all but a single variable. Then a perfect solution could be found by partitioning the objects into those with high vs. low values on that single variable. A three-cluster solution could be found by separating objects with high, medium, and low values on that variable. With many variables, there are an extraordinarily large number of such possible "perfect" solutions, all potentially very different from one another.

Because of our desire for robustness, we have not provided an automatic variable weighting feature in CCEA at this time.

Alternative Starting Points

Three methods of obtaining starting points are currently available in CCEA:

1. Distance-based points. This method searches for a set of starting points relatively distant from each other. The intuitive basis for this procedure is recognition that points far apart are likely to belong to different clusters. If we can locate one starting point in or near each cluster, then we should have a good chance of identifying that cluster in the solution.

The procedure for choosing starting points in this way is as follows:

1. Choose a random sample of objects from the entire set to be clustered (the larger of 50 objects or 10% of the entire set, up to a maximum of 250 objects; or all objects if fewer than 50 are available). If there are k to be chosen, select the first k object points as a temporary set. Compute the pairwise distances among those points. Save the size of the current minimum distance, as well as the identities of those two closest points. Call those two closest points A and B.

2. Examine the next object point in the file (call it C), computing its distance to each of the k points already selected. If the minimum of these distances is greater than the current minimum distance (between A and B), add C to the temporary set and discard either point A or B, whichever is closest to C.

3. Update the minimum distance for the new set of selected points. Call the new closest points A and B.

4. Repeat steps 2 and 3 until all points in the random subsample have been examined.

2. Hierarchical-based starting points. This method selects a random sample of objects (the larger of 50 objects or 10% of the entire set, up to a maximum of 250 objects; or all objects if fewer than 50 are available) and does a hierarchical ("complete linkage") cluster analysis of those objects. The centroids of the resulting clusters are taken as starting points.

In our experience, starting points found by this method are the most successful in leading to reproducible solutions. However, there is a lot of variability in their quality, and starting points chosen by this method can also lead to the worst of solutions. We recommend that this method always be included if there are to be several replications, but that it not be used if there is to be only a single replication of each clustering.

3. Density-based points. The two-cluster example in the previous section was misleading in the sense that within each swarm we portrayed the points as uniformly distributed. That is not true of most data sets, which are commonly thought to be mixtures of multivariate normal distributions. Such distributions are "lumpy," having greater densities of points near the centers of regions that we would normally be interested in recovering as clusters.

This suggests that one way to choose starting points would be to look for points that have many

others relatively close to them, so long as no other starting points are chosen from the same region.

This is the third option in CCEA. The Density-based starting solution is as follows:

1. Choose a random sample objects from the entire set to be clustered (the larger of 50 objects or 10% of the entire set, up to a maximum of 250 objects; or all objects if fewer than 50 are available).

2. Compute pairwise distances between objects.

3. Convert each distance to a "proximity" by the transformation:

proximity = 1 / (1 + distance).

A proximity is 1 if the distance between points is zero, and the proximity is zero for infinitely distant points. The proximity value is much more sensitive to small distances than larger ones.

4. Compute the sum of each point's proximities with the other points. In doing so, we are paying attention mostly to each point's closeness to nearby points.

5. Select the point with highest sum of proximities as a starting point.

6. Adjust all proximities so that those points close to the chosen point are made unlikely to be selected in the future.

This is done by a matrix operation involving a sweep operator. First, replace each diagonal element of the proximity matrix by the largest value in that row or column. Indicate the proximity between points i and j as p(i,j). Suppose the point just chosen has subscript 1. Then the adjustment to proximities to reduce them in proportion to their closeness to point 1 is:

p(i,j) = p(i,j) - (p(i,1) * p(1,j) / (p(1,1),

and any values that become negative are set at zero.

7. Repeat steps 4 - 6 until the desired number of points have been selected.

This procedure seems to find starting points that permit relatively good recovery of known clusters. In the example above, the two points chosen would presumably be near the centers of the two point swarms, and convergence would have been immediate.

However, as with hierarchical starting points, this procedure is based on random samples of only a subset of the points, and it is possible that an important cluster could be present in the data but not represented among the subset of points chosen. This suggests that when starting points are selected by this procedure, the entire process should be repeated several times to make sure that the chosen solution is reproducible.

7.3 Starting Points

Several types of starting points are available for K-Means clustering.

1. "Distance-based" starting points. These are points (respondents) chosen to be relatively far apart. A random subset* of the respondents is chosen for each replication for determining the starting points.

2. "Hierarchical-based" starting points. A random subset* of the respondents are chosen and a hierarchical ("complete linkage") cluster analysis is done. The centroid of each cluster is computed, and those centroids are taken as the starting points. Unless the data set is less than 50 people, the starting solution is likely to be different each time, and this method may be used advantageously to select starting points for several replications.

3. "Density-based" starting points. As with the previous method, a subset of *respondents is chosen at random. An analysis is done to select respondents that are near the centers of relatively dense regions in the space. If there are more than 50 respondents in the data file, this method will also produce different starting solutions each time, and can therefore be used profitably for multiple replications.

4. **Mixed strategy.** This approach cycles among all of the starting point methods, including the user-defined strategy (if provided). We recommend this strategy for most purposes.

5. **User-Defined strategy**. If you have previously done work with this or a similar dataset such that you have an existing solution, this can be used as a starting point. If you previously have established group membership for each respondent, you can select a file containing that information as the starting point. Or, if you have group means on the variables, you can select a file that contains that information. The formats and procedures are described in the Settings tab documentation. You would not use the user-defined strategy options for replicated clusterings, because they would all produce identical results.

*The random subset consists of the larger of 50 respondents or 10% of the entire data set, up to a maximum of 250 respondents; or all respondents if fewer than 50 are available.

Performance of Starting Points

This is a report on the relative performance of Distance-, Hierarchical-, and Density-based starting points. (The analysis was performed using version 1 of CCA, which used 50 cases in the random subsets when drawing starting points.)

Two separate analyses were done with randomly constructed data sets. All data sets contained 300 objects, falling into 5 true clusters. The expected dispersion between cluster means was:

- equal to the dispersion within clusters, or
- twice the dispersion within clusters, or
- three times the dispersion within clusters.

Only the 5-cluster solution was computed for each data set, so we could measure success at recovering the true cluster composition as well as reproducibility.

The first analysis examined 207 data sets with dispersion = 1 and 241 data sets with dispersion = 2. Each data set was clustered 5 times: once with Distance-, twice with Hierarchical-, and twice with Density-based starting points. In each case the most reproducible replication was noted, as was the replication producing the best recovery of the true clusters. In determining which method had produced the best reproducibility, ties were broken at random. In determining which method had produced the

best recovery no such random method was used. The Distance-based measure was indicated as winning in any ties involving it, and the Hierarchical method was indicated as winning in any ties between just it and the Density method. Thus, this analysis favored the Distance and Hierarchical methods in examining recovery of true clusters.

Results of first analysis:

Overall, the Hierarchical-based starting points did best, followed by the Density-based starting points. However, each of the methods won a substantial proportion of the time.

Cluster Dispersion = 1

	Best Reproducibility	Best Recovery of true clusters
Hierarchical-based	45%	36%
Density-based	33%	37%
Distance-based	22%	27%
	100%	100%

Cluster Dispersion = 2

	Best Reproducibility	Best Recovery of true clusters
Hierarchical-based	42%	39%
Density-based	33%	31%
Distance-based	25%	30%
	100%	100%

Overall, the most reproducible replicate averaged 87.5% reproducibility, and the replicate with best recovery of the true clusters had average recovery of 88.4%.

Since each method won a substantial proportion of the time with either type of data, it seems best to use all three of the methods with each clustering.

However, the question arose of whether the Distance-based starting points might have had an unfair disadvantage, only having been used once, whereas the other methods each had two opportunities to win. There was also concern about unfair breaking of ties in assessing recovery of true clusters. These concerns led to the second analysis.

In the second analysis 1419 data sets were examined (511 with dispersion = 1, 767 with dispersion = 2, and 141 with dispersion = 3.)

In these clusterings only three replications were done, with each method of selecting starting points used only once. The average reproducibility and recovery of each method were computed, regardless of ties. Additionally, a count was made of the proportion of times each method won over both other methods (without ties).

Results of second analysis:

Cluster Dispersion = 1

	Best Reproducibility	Best Recovery of true clusters
Hierarchical-based	29%	26%
Density-based	22%	35%
Distance-based	31%	24%
Some methods tied	18%	15%
	100%	100%

Cluster Dispersion = 2

	Best Reproducibility	Best Recovery of true clusters
Hierarchical-based	25%	26%
Density-based	24%	30%
Distance-based	19%	20%
Some methods tied	32%	24%
	100%	100%

Cluster Dispersion = 3

	Best Reproducibility	Best Recovery of true clusters
Hierarchical-based	20%	13%
Density-based	14%	28%
Distance-based	11%	13%
Some methods tied	45%	46%
	100%	100%

The proportion of ties among different methods increases as clusters are more clearly differentiated from one another, as would be expected.

Overall, the Hierarchical-based starting points had a narrow advantage in reproducibility, but the Density-based starting points were more successful at recovering true clusters. However, the Distance-based starting points were most successful a significant proportion of the time.

Average levels of reproducibility and recovery of true clusters were as expected. Overall average reproducibility was 84.6%, and the overall average recovery of true clusters was 83.4%. Hierarchical-based starting points produced slightly higher (less than 1%) average reproducibility than either of the other two methods, and Density-based starting points produced slightly higher (less than 2%) average recovery of true clusters.

The conclusions remain that different starting points produce different answers, and that any of the methods might produce the best answer. The best strategy is to use them all.

Index