



**Introduction to IBM SPSS
Modeler and Data Mining**

Student Guide

Course Code: 0A002

ERC 1.0

Authorized

IBM | Training

Introduction to IBM SPSS Modeler
and Data Mining

0A002

Published October 2010

Licensed Materials - Property of IBM

© Copyright IBM Corp. 2010

US Government Users Restricted Rights - Use,
duplication or disclosure restricted by GSA ADP
Schedule Contract with IBM Corp.

IBM, the IBM logo and ibm.com are trademarks
of International Business Machines Corp.,
registered in many jurisdictions worldwide.

SPSS, and PASW are trademarks of SPSS Inc.,
an IBM Company, registered in many
jurisdictions worldwide.

Microsoft, Windows, Windows NT, and the
Windows logo are trademarks of the Microsoft
Corporation in the United States, other countries,
or both.

UNIX is a registered trademark of The Open
Group in the United States and other countries.

Other product and service names might be
trademarks of IBM, SPSS or other companies.

This guide contains proprietary information which
is protected by copyright. No part of this
document may be photocopied, reproduced, or
translated into another language without a legal
license agreement from IBM Corporation.

Any references in this information to non-IBM
Web sites are provided for convenience only and
do not in any manner serve as an endorsement
of those Web sites. The materials at those Web
sites are not part of the materials for this IBM
product and use of those Web sites is at your
own risk.

TABLE OF CONTENTS

LESSON 1: INTRODUCTION TO DATA MINING	1-1
1.1 INTRODUCTION TO DATA MINING.....	1-1
1.2 KEY QUESTIONS FOR A DATA-MINING PROJECT	1-2
1.3 A STRATEGY FOR DATA MINING: THE CRISP-DM PROCESS METHODOLOGY	1-2
1.4 SKILLS NEEDED FOR DATA MINING.....	1-9
1.5 PLAN OF THE COURSE	1-10
LESSON 2: THE BASICS OF USING PASW MODELER	2-1
2.1 PASW MODELER AND PASW MODELER SERVER.....	2-1
2.2 STARTING PASW MODELER	2-2
2.3 USING THE MOUSE	2-4
2.4 VISUAL PROGRAMMING	2-5
2.5 BUILDING STREAMS WITH PASW MODELER.....	2-10
2.6 GETTING HELP	2-11
2.7 CUSTOMIZING PALETTES	2-13
EXERCISES.....	2-17
LESSON 3: READING DATA FILES	3-1
3.1 READING DATA FILES INTO PASW MODELER	3-1
3.2 READING DATA FROM FREE-FIELD TEXT FILES	3-2
3.3 FIRST CHECK ON THE DATA	3-9
3.4 READING IBM SPSS STATISTICS DATA FILES.....	3-14
3.5 READING DATA USING ODBC	3-18
3.6 READING DATA FROM EXCEL SPREADSHEETS.....	3-29
3.7 DATA FROM PASW DATA COLLECTION PRODUCTS.....	3-31
3.8 SAS SOFTWARE COMPATIBLE DATA	3-31
3.9 XML DATA	3-31
3.10 DEFINING FIELD MEASUREMENT LEVEL	3-32
3.11 FIELD ROLE	3-36
3.12 SAVING A PASW MODELER STREAM	3-36
3.13 APPENDIX A: READING DATA FROM FIXED-FIELD TEXT FILES	3-37
3.14 APPENDIX B: WORKING WITH DATES	3-42
3.15 DECLARING DATE FORMATS IN PASW MODELER	3-42
EXERCISES.....	3-48
LESSON 4: DATA UNDERSTANDING.....	4-1
4.1 INTRODUCTION.....	4-1
4.2 MISSING DATA IN PASW MODELER	4-1
4.3 ASSESSING MISSING DATA	4-3
4.4 USING THE DATA AUDIT NODE FOR MISSING DATA	4-5
4.5 AUTO CHECKING FOR MISSING AND OUT-OF-BOUNDS VALUES	4-21
4.6 FIELD DISTRIBUTIONS AND SUMMARY STATISTICS.....	4-25
4.7 APPENDIX: ADVICE ON HANDLING MISSING VALUES	4-32
EXERCISES.....	4-33
LESSON 5: OUTLIERS AND ANOMALOUS DATA	5-1

5.1	INTRODUCTION	5-1
5.2	WHAT IS ANOMALOUS DATA?.....	5-1
5.3	OUTLIERS IN CATEGORICAL FIELDS	5-3
5.4	OUTLIERS IN CONTINUOUS FIELDS	5-5
5.5	OUTLIERS IN TWO FIELDS (CATEGORICAL AND CONTINUOUS)	5-15
5.6	OUTLIERS IN TWO CONTINUOUS FIELDS	5-17
5.7	THE ANOMALY NODE	5-19
	EXERCISES	5-25
	LESSON 6: INTRODUCTION TO DATA MANIPULATION.....	6-1
6.1	INTRODUCTION	6-1
6.2	A BRIEF INTRODUCTION TO THE CLEM LANGUAGE	6-2
6.3	FIELD OPERATIONS AND THE FILTER NODE	6-3
6.4	FIELD REORDERING	6-6
6.5	THE DERIVE NODE.....	6-8
6.6	RECLASSIFY NODE.....	6-17
6.7	EXECUTING FIELD OPERATION NODES SIMULTANEOUSLY.....	6-19
6.8	AUTOMATICALLY GENERATING OPERATIONAL NODES.....	6-23
	EXERCISES	6-29
	LESSON 7: LOOKING FOR RELATIONSHIPS IN DATA.....	7-1
7.1	INTRODUCTION	7-1
7.2	STUDYING RELATIONSHIPS BETWEEN CATEGORICAL FIELDS.....	7-2
7.3	MATRIX NODE: RELATING TWO CATEGORICAL FIELDS	7-2
7.4	THE WEB NODE	7-6
7.5	CORRELATIONS BETWEEN CONTINUOUS FIELDS.....	7-17
7.6	MEANS NODE: ANALYZING THE RELATIONSHIP BETWEEN CONTINUOUS AND CATEGORICAL FIELDS.....	7-21
7.7	USING THE GRAPHBOARD NODE TO EXAMINE RELATIONSHIPS	7-24
	EXERCISES	7-33
	LESSON 8: COMBINING DATA FILES	8-1
8.1	INTRODUCTION	8-1
8.2	USING THE APPEND NODE TO COMBINE DATA FILES	8-2
8.3	USING A MERGE NODE TO COMBINE DATA FILES	8-12
8.4	SUPERNODE	8-18
8.5	EDITING SUPERNODES.....	8-21
8.6	SAVING AND INSERTING SUPERNODES	8-22
	EXERCISES	8-24
	LESSON 9: AGGREGATING DATA	9-1
9.1	INTRODUCTION	9-1
9.2	SUMMARIZING DATA USING THE AGGREGATE NODE.....	9-1
9.3	RESTRUCTURING SET FIELDS USING THE SETTOFLAG NODE.....	9-8
9.4	COMBINING AGGREGATION AND SETTOFLAG OUTPUT	9-12
9.5	RESTRUCTURING DATA USING THE RESTRUCTURE NODE	9-14
	EXERCISES	9-20
	LESSON 10: SELECTING, SAMPLING AND PARTITIONING RECORDS	10-1

10.1	INTRODUCTION.....	10-1
10.2	USING THE DISTINCT NODE TO REMOVE DUPLICATES	10-1
10.3	SORTING RECORDS	10-4
10.4	SELECTING RECORDS	10-8
10.5	AUTOMATICALLY GENERATING A SELECT NODE.....	10-12
10.6	USING THE SAMPLE NODE TO SELECT RECORDS	10-14
10.7	BALANCING DATA.....	10-21
10.8	THE PARTITION NODE	10-22
10.9	DATA CACHING.....	10-27
	EXERCISES.....	10-30
	LESSON 11: MODELING TECHNIQUES IN PASW MODELER.....	11-1
11.1	INTRODUCTION.....	11-1
11.2	NEURAL NETWORKS	11-2
11.3	RULE INDUCTION	11-3
11.4	BAYES NETWORKS	11-5
11.5	SUPPORT VECTOR MACHINES	11-6
11.6	SELF-LEARNING RESPONSE MODEL.....	11-7
11.7	LINEAR REGRESSION.....	11-8
11.8	LOGISTIC REGRESSION	11-9
11.9	DISCRIMINANT ANALYSIS	11-10
11.10	GENERALIZED LINEAR MODELS	11-11
11.11	COX REGRESSION.....	11-11
11.12	AUTOMATED MODELING.....	11-12
11.13	CLUSTERING.....	11-13
11.14	ASSOCIATION RULES.....	11-15
11.15	SEQUENCE DETECTION.....	11-15
11.16	PRINCIPAL COMPONENTS	11-16
11.17	TIME SERIES ANALYSIS.....	11-16
11.18	WHICH TECHNIQUE, WHEN?.....	11-17
	LESSON 12: RULE INDUCTION	12-1
12.1	INTRODUCTION.....	12-1
12.2	RULE INDUCTION IN PASW MODELER	12-1
12.3	RULE INDUCTION USING C5.0.....	12-2
12.4	BROWSING THE MODEL.....	12-9
12.5	GENERATING AND BROWSING A RULE SET.....	12-15
12.6	DETERMINING MODEL ACCURACY	12-18
12.7	RULE INDUCTION USING CHAID.....	12-30
	EXERCISES.....	12-35
	LESSON 13: AUTOMATING MODELS FOR CATEGORICAL TARGETS	13-1
13.1	INTRODUCTION.....	13-1
13.2	CREATING A FLAG FIELD	13-1
13.3	USING THE AUTO CLASSIFIER	13-5
	EXERCISES.....	13-16
	LESSON 14: AUTOMATING MODELS FOR CONTINUOUS TARGETS	14-1

14.1	INTRODUCTION	14-1
14.2	AUTO NUMERIC STREAM.....	14-1
14.3	USING THE AUTO NUMERIC.....	14-3
	EXERCISES	14-13
	LESSON 15: MODEL UNDERSTANDING.....	15-1
15.1	INTRODUCTION	15-1
15.2	REVIEWING MODEL ACCURACY WITH THE ANALYSIS NODE	15-1
15.3	MODEL PREDICTIONS FOR CATEGORICAL FIELDS.....	15-7
15.4	MODEL PREDICTIONS FOR CONTINUOUS FIELDS.....	15-12
	EXERCISES	15-20
	LESSON 16: COMPARING AND COMBINING MODELS.....	16-1
16.1	INTRODUCTION	16-1
16.2	COMPARING MODELS WITH THE ANALYSIS NODE.....	16-1
16.3	EVALUATION CHARTS FOR MODEL COMPARISON	16-4
16.4	COMBINING MODELS	16-8
	EXERCISES	16-16
	LESSON 17: DEPLOYING AND USING MODELS	17-1
17.1	INTRODUCTION	17-1
17.2	DEPLOYING A MODEL.....	17-1
17.3	EXPORTING MODEL RESULTS.....	17-5
17.4	ASSESSING MODEL PERFORMANCE.....	17-6
17.5	MODEL LIFETIME.....	17-8
17.6	UPDATING A MODEL.....	17-8
	EXERCISES	17-10
	APPENDIX A: PASW MODELER OPTIONS AND STREAM PROPERTIES	A-1
A.1	SETTING PASW MODELER OPTIONS	A-1
A.2	STREAM PROPERTIES	A-7
	APPENDIX B: RUNNING STATISTICS COMMANDS FROM PASW MODELER	B-1
B.1	THE STATISTICS OUTPUT NODE	B-1
B.2	USING AN EXISTING SYNTAX FILE.....	B-3
	DATA MINING REFERENCES	R-1

Lesson 1: Introduction to Data Mining

Objectives

- To introduce the concept of Data Mining
- To introduce the CRISP-DM process model as a general framework for carrying out Data Mining projects
- To describe successful data mining projects and reasons projects fail
- To describe the skills needed for data mining
- To sketch the plan of this course

1.1 *Introduction to Data Mining*

With increasingly competitive markets and the vast capabilities of computers, many businesses find themselves faced with complex databases and a need to easily identify useful patterns and actionable relationships.

Data Mining is a general term which encompasses a number of techniques to extract useful information from (large) data files, without necessarily having preconceived notions about what will be discovered. The useful information often consists of patterns and relationships in the data that were previously unknown or even unsuspected. Data mining is also sometimes called *Knowledge Discovery in Databases* (KDD).

A common misconception is that data mining involves passing huge amounts of data through intelligent technologies that, alone, find patterns and give magical solutions to business problems. This is not true, although there is more automation than in traditional statistical applications.

Data mining is an interactive and iterative process. Business expertise must be used jointly with advanced technologies to identify underlying relationships and features in the data. A seemingly useless pattern in data discovered by data-mining technology can often be transformed into a valuable piece of actionable information using business experience and expertise.

Many of the techniques used in data mining are referred to as “machine learning” or “modeling.” Several of these techniques require a different approach to model generation and testing compared to standard statistics. Existing data is used to “train” a model, and then ‘test’ it to determine whether it should be deemed acceptable and likely to generalize to the population of interest. Due to the typically large files and weak assumptions made about the distribution of the data, data mining tends to be less focused on statistical significance tests and more on practical importance.

Data mining has been used in hundreds of applications. Some of these include:

- Developing models to detect fraudulent phone or credit-card activity
- Predicting good and poor sales prospects
- Predicting next page browsed on a website.
- Identifying customers who are likely to cancel their policies, subscriptions, or accounts
- Classifying customers into groups with distinct usage or need patterns
- Predicting who is likely to not renew a contract for mobile phone service
- Finding rules that identify products that, when purchased, predict additional purchases
- Identifying factors that lead to defects in a manufacturing process

- Predicting whether a heart attack is likely to recur among those with cardiac disease

1.2 Key Questions for a Data-Mining Project

Before considering which specific data-mining technique is suitable, the business problem and the data need to be assessed for any potential data-mining project. There are several aspects which should be considered:

1. Are data available?

Data need to be in an easily accessible format. It is often the case that relevant data files are stored in several locations and/or in different formats and need to be pulled together before analysis. Data may even not be in electronic format, possibly existing only on paper and needing data coding and data entry before data mining can be done. A data miner should also be aware of potential drawbacks, such as political or legal reasons why the data cannot be accessed.

2. Do data cover the relevant factors?

To make a data-mining project worthwhile, it is important that the data, as far as possible, contain all relevant factors/variables. Obviously, it is often the object of data mining to help identify relevant factors in the data. However, greater accuracy of predictions can be achieved if thought is given beforehand to this question

3. Are the data too noisy?

The term “noise” in data mining refers to errors in data, and also sometimes missing data. Some noise in data is not unusual, and the machine learning capabilities of PASW® Modeler have been shown to successfully handle data containing up to 50% noise. However, the more noise in data, the more difficult it will be to make accurate predictions.

4. Are there enough data?

The answer to this question depends on each individual problem. Very often it isn’t the size of the data that causes difficulties in data mining but more whether it is representative of the target population and covers all possible outcomes. As with most data analysis techniques, the more complex the patterns or relationships, the more records required to find them. If the data provide good coverage of possible outcomes, reasonable results can often be achieved using data sizes as small as a few thousand (or even a few hundred) records.

5. Is expertise on the data available?

Successful data mining requires what is called *domain expertise*, which is practical and relevant knowledge about how the data were generated, the data characteristics, how the data are used by the organization, and what are the intended uses of the outcome of a data-mining project. When you are responsible for mining data from another organization or department on which you don’t have firsthand knowledge, it is extremely important that experts who understand the data and the problem are available. They not only guide you in identifying relevant factors and help interpret the results, but also can often sort out the truly useful pieces of information from misleading artifacts often due to oddities in the data or relationships uninteresting from a business perspective.

1.3 A Strategy for Data Mining: the CRISP-DM Process Methodology

As with most business endeavors, data mining is much more effective if done in a planned, systematic way. Even with cutting edge data-mining tools such as PASW Modeler, the majority of the work in

data mining requires the careful eye of a knowledgeable business analyst to keep the process on track. To guide your planning, answer the following questions:

- What substantive problem do you want to solve?
- What data sources are available, and what parts of the data are relevant to the current problem?
- What kind of preprocessing and data cleaning do you need to do before you start mining the data?
- What data mining technique(s) will you use?
- How will you evaluate the results of the data mining analysis?
- How will you get the most out of the information you obtained from data mining?

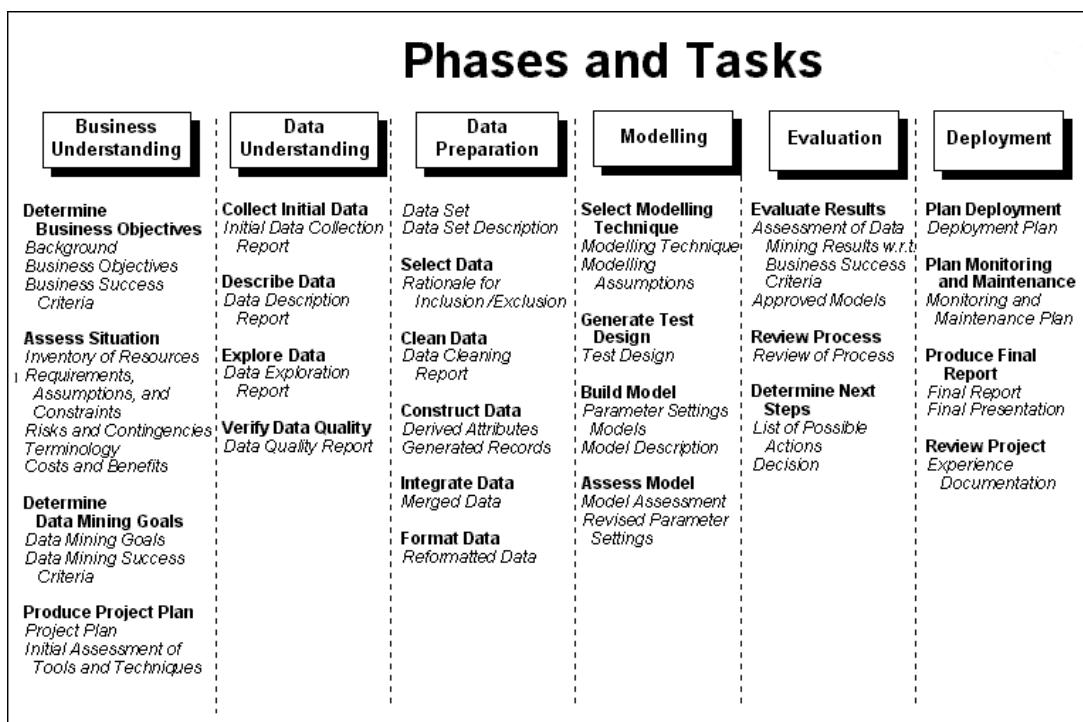
The typical data-mining process can become complicated very quickly. There is a lot to keep track of—complex business problems, multiple data sources, varying data quality across data sources, an array of data mining techniques, different ways of measuring data mining success, and so on.

To stay on track, it helps to have an explicitly defined process model for data mining. The process model guides you through the critical issues outlined above and makes sure that the important points are addressed. It serves as a data mining road map so that you won't lose your way as you dig into the complexities of your data.

The data mining process model recommended for use with PASW Modeler is the Cross-Industry Standard Process for Data Mining (CRISP-DM). As you can tell from the name, this model is designed as a general model that can be applied to a wide variety of industries and business problems. The first version of the CRISP-DM process model is now available. It is included with PASW Modeler and can be downloaded from www.crisp-dm.org.

The general CRISP-DM process model includes six phases that address the main issues in data mining. The six phases fit together in a cyclical process.

These six phases cover the full data mining process, including how to incorporate data mining into your larger business practices. These phases are listed in the diagram in Figure 1.1.

Figure 1.1 CRISP-DM Model

The six phases include:

Business understanding. This is perhaps the most important phase of data mining. Business understanding includes determining business objectives, assessing the situation, determining data mining goals, and producing a project plan. Activities in this phase include:

- Identify business objectives and success criteria
- Perform a situational assessment (resources, constraints, assumptions, risks, costs, and benefits)
- Determine the goals of the data-mining project and success criteria
- Produce a project plan

Data understanding. Data provides the "raw materials" of data mining. This phase addresses the need to understand what your data resources are and the characteristics of those resources. It includes collecting initial data, describing data, exploring data, and verifying data quality.

Data preparation. After cataloging your data resources, you will need to prepare your data for mining. Preparations include selecting, cleaning, constructing, integrating, and formatting data. These tasks will likely be performed multiple times, and not in any prescribed order. These tasks can be very time consuming but are critical for the success of the data-mining project. Some activities at the Data Understanding and Data Preparation phases include:

- Extracting data from a data warehouse or data mart
- Linking tables together within a database or in PASW Modeler
- Combining data files from different systems
- Reconciling inconsistent field values
- Identifying missing, incorrect, or extreme data values

- Data selection
- Restructuring data into a form the analysis requires
- Transforming relevant fields (taking differences, ratios, etc.)

Modeling. This is, of course, the flashy part of data mining, where sophisticated analysis methods are used to extract information from the data. This phase involves selecting modeling techniques, generating test designs, and building and assessing models. Developing a model is an iterative process—as it can be in standard statistical modeling—and you should expect to try several models, and modeling techniques, before finding a best model. As we demonstrate in this course, another feature that separates data-mining from other approaches is the use of multiple models to make predictions, building on the strengths of each technique.

Evaluation. Once you have chosen your models, you are ready to evaluate how the data mining results can help you to achieve your business objectives. At this stage in the project, you have built a model (or models) that appears to have high quality, from a data analysis perspective. Before writing final reports and deploying the model, it is important to more thoroughly evaluate the model, and review the steps executed to construct the model, to be certain it properly achieves the business objectives. A key aim is to determine if there is some important business issue that has not been sufficiently considered. At the end of this phase, a decision will be made on the use of the data-mining results.

Deployment. Now that you've invested all of this effort, it's time to reap the benefits. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data-mining process.

Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that the organization can use for decision-making. So in essentially all projects, a final report will need to be produced and distributed.

Most critical is deployment of the model to make predictions or create scores against new data. This might be relatively simple if done within the data-mining software, or more complex if the model is to be applied directly against an existing database.

A plan should be developed to monitor the model's predictions and success in order to verify that the model still holds true. This might comprise automated analyses, which produce warnings if certain events occur (for example, if the gap between the predicted and observed value exceeds a specified amount).

There are some key points to the CRISP-DM process model. First, while there is a general tendency for the process to flow through the steps in the order outlined above, there are also a number of places where the phases influence each other in a nonlinear way. For example, data preparation usually precedes modeling. However, decisions made and information gathered during the modeling phase can often lead you to rethink parts of the data preparation phase, which can then present new modeling issues, and so on. The two phases feed back on each other until both phases have been resolved adequately. Similarly, the evaluation phase can lead you to reevaluate your original business understanding, and you may decide that you've been trying to answer the wrong question. At this point, you can revise your business understanding and proceed through the rest of the process again with a better target in mind.

The second key point is the iterative nature of data mining. You will rarely, if ever, simply plan a data mining project, execute it and then pack up your data and go home. Using data mining to address

your customers' demands is an ongoing endeavor. The knowledge gained from one cycle of data mining will almost invariably lead to new questions, new issues, and new opportunities to identify and meet your customers' needs. Those new questions, issues, and opportunities can usually be addressed by mining your data once again. This process of mining and identifying new opportunities should become part of the way that you think about your business and a cornerstone of your overall business strategy.

Model Validation

Since most data-mining methods do not depend on specific data distribution assumptions (for example, normality of errors) to draw inferences from the sample to the population, validation is necessary (and is basically equivalent to statistical testing). It is usually done by fitting the model to a portion of the data (called the Training data) and then applying the predictions to, and evaluating the results with, the other portion of the data (called the Test or Validation data). In this way, the validity of the model is established by demonstrating that it applies to (fits) data independent of that used to derive the model. Statisticians often recommend such validation for statistical models, but it is crucial for more general (less distribution bound) data-mining techniques.

Measures of Project Success

The CRISP-DM model tells us, in the Evaluation phase, to assess the results with respect to business success, not statistical criteria. And indeed, from the moment you begin to develop a research question, the eventual evaluation of the results should be foremost in your mind. The initial assessment will be directly tied to the modeling effort. That is, you will be concerned with predictive accuracy (who is a churner) or with finding interesting relationships between products or people (in association or cluster analysis). But in the long run, the success of a data-mining effort will be measured by concrete factors such as reduced savings, return on investment or profitability, and so forth.

To determine success, you must monitor the model after it is deployed. Once a model has been deployed, plans must be put in place to record the data and information that make it possible to assess the model's success. Thus, if a real-time model is being used to supply sales reps with offers for customers, both the suggested offer and the customer's decision, among other factors, must be retained in a database for future analysis.

As you develop a model and think about its deployment, consider what other measures you can use to determine how successful and useful it is—from a business or organization perspective—some time downstream from deployment. Don't wait to mention these factors until after deployment, but bring them to the attention of your colleagues and management early on, so that tracking systems or reports can be developed. So in the case of a financial institution using data mining to predict customer retention, there are many other factors to investigate beyond simple retention. Changes in average account balance, account activity, account profitability, the opening of other accounts, and use of other services (ATM card) can all be investigated after the model is deployed to see if they are changing also.

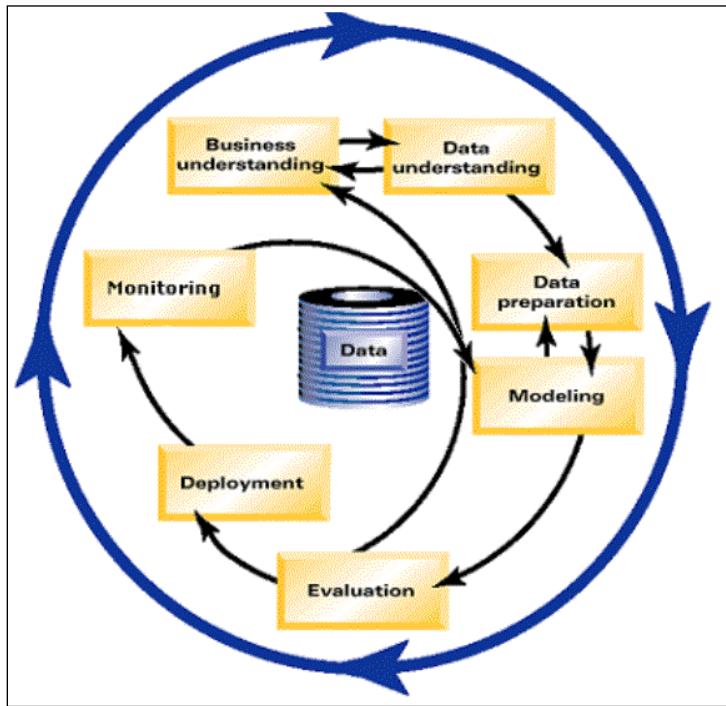
Don't forget to consider the cost of errors as another yardstick of success. We tend to focus on success but there will always be errors, and sometimes the cost of making errors can be high. For example, mispredicting which insurance claims are fraudulent may be expensive because of the effort involved to investigate the claim further.

Some data-mining tools allow you to take cost into account when estimating the model. Use this feature if it is possible to make even a rough cost estimate. But even when you can't use costs in the modeling phase, be sure to think carefully about the costs of errors before deployment. And if no

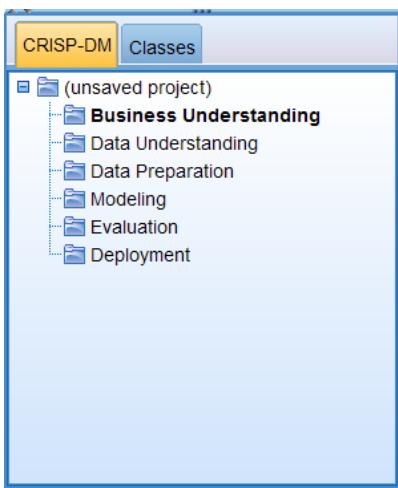
reliable cost estimates are possible beforehand, then try to gather this information after the fact for use in future data mining projects and as ad hoc evaluation criteria.

The figure below illustrates the main stages in a successful data mining process: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment (we have also added monitoring of model performance, which as just noted is crucial). The outer circle represents the fact that the whole process is iterative. The clockwise order of the tasks represents the common sequence.

Figure 1.2 Stages in the CRISP-DM Process



To assist you in organizing your PASW Modeler programs (streams) around the CRISP-DM framework, PASW Modeler contains a Project window. In the Project window, a project folder contains subfolders corresponding to the phases in CRISP-DM. This makes it easier to organize your PASW Modeler programs and other documents that are associated with a data-mining project. You can save a project file that contains links to PASW Modeler streams and other documents.

Figure 1.3 PASW Modeler Project Window

Causes of Failure in Data Mining

Not every data-mining project is successful, or, at the least, not as successful as you might have anticipated. As with any research, lots of things can go wrong. In this section we review some of the other more serious problems that can occur.

Bad data. We stressed earlier the need for clean and valid data as input to the data mining effort. If instead the data have large amounts of error, no data mining technique will be able to compensate for this problem. In the worse case, a potentially good set of predictors may fail because of error that masks their effect. Never scrimp on the time you spend on data preparation and cleaning, and continue to check the data as you modify it during the analysis and afterwards. The time to learn about bad data is before, not after, the report has been written or the model deployed.

Organizational resistance. While not strictly a failure of the data mining per se, difficulties implementing a solution are still part of the whole data-mining effort. An HMO investigated ways to reduce costs by looking at patterns of treatment and care, and found that there was an optimal length of stay in the hospital for several types of major surgeries. While not requiring doctors to rigidly follow the statistical results (which would be inappropriate for any specific patient), the HMO encouraged doctors to take this information into account. But after a few months, it was clear that length of stay decisions were not changing, i.e., that the physicians were sticking to their current practices.

When resistance occurs, the best strategy is usually further education on the potential benefits of the solution, or perhaps, implementation in only a portion of the organization. For the HMO, this could mean convincing a few doctors initially to change their release decisions, hoping that eventually more will follow this lead.

Results that cannot be deployed. Sometimes a model cannot be deployed for factors other than organizational opposition. The most common reason is because factors found to be important are out of the control of the organization, or can't legally be used in marketing or in making decisions. A consumer products company discovered that certain types of promotions were successful and led to repeat business, but could only offer these promotions to customers it could readily identify, which in practice were those who returned a registration card or bought a service contract.

Some obstacles can be anticipated, and the data-mining process adjusted accordingly. If a model can be only partially implemented, as with the consumer products firm, it may still be worthwhile to do the analysis when sufficiently good results would justify the effort (this is always a judgment call).

Problems of cause and effect: Research methodology is important for the data-mining effort. One reason is because a carefully formulated study will consider whether there is a cause-and-effect relationship between the predictors and outcome variable. For example, customer satisfaction research often uses attitudes about product/service attributes to predict overall satisfaction, willingness to buy again/to remain a customer, or willingness to recommend a product/service. In terms of cause and effect, all these attitudes about attributes and future actions or satisfaction occur at one point in time, i.e., when the survey is conducted. It can then be argued that while these attitudes may be correlated, claiming that one attitude “causes” another is ill advised. Instead, the attitudes may be mutually reinforcing. When this is true, the predictions from a model about how changes in attribute attitudes affect the outcome variables may be invalid. The basic point is that you must be certain that inputs/predictors in a model occur before the outputs.

1.4 Skills Needed for Data Mining

For a successful data-mining project, several disparate skills are useful, and they rarely reside in a single individual.

Understanding of the Business

Framing the business question to be answered by data mining, evaluating the results in terms of business objectives, and presenting the recommendations all require knowledge of the specific business area and organization. Thus someone who knows the critical issues facing the organization is well suited to pose questions that data mining might address. He or she can also evaluate a data mining solution in terms of business objectives and whether it makes sense. It should be pointed out that experienced data mining consultants who focus within an industry could develop a good knowledge of these issues. Without this component, a data-mining project runs the risk of producing a good technical solution to a question unimportant to the business.

Database Knowledge

There is an old saying that an army travels on its stomach. Similarly a data-mining project cannot succeed without good data. The most sophisticated analytic techniques cannot be expected to overcome inconsistent and incomplete data. For this reason a database administrator (DBA) is usually a key member of the data-mining project team. Typically, neither the business expert nor the analyst has a sufficiently deep knowledge of the data available on the company’s systems to do this. What data tables or files are available? How are they linked? How are the fields coded and which require aggregation? What are reasonable and what are incorrect or outlying data values? Only someone familiar with the corporate data systems can usually answer these and other questions. Without this component, you run the risk of producing an incorrect answer to the right question using the best method, or of failing to find a reachable solution.

Data Mining Methods

Although data-mining tools are available that allow pushbutton ease of running an analysis, as you would expect, knowledge of data mining techniques is needed. Deciding on the best tools to use for a specific question, knowing how to tweak a technique to its optimum, being able to assess the effects of odd data values or missing data, and recognizing that something doesn’t look right, can all contribute to the success of the project. Some of these skills would reside in a trained and experienced analyst, since they are useful for all analyses. However, given that our focus is on data-mining

methods, an analyst skilled in these techniques (trained or self-taught) is needed. Without this component, you may fail to answer or incorrectly answer an important question, even with the benefit of good data.

Deployment

The deployment of a model on new data may be done outside of PASW Modeler in the database. Or it might use a generated model from PASW Modeler but embed it in another application. Specific skills are needed to implement these types of deployments, and this may call for other team members with programming skills that a data-mining analyst doesn't possess.

Usually a Team

For these reasons, teams perform most data mining projects and individuals contribute differently to the various steps in the data mining process. It would be ideal if all the needed skills were to reside in one person, but this is rarely the case. Occasionally, a team member can serve multiple functions (business and database knowledge, or database and data mining knowledge), but it is relatively rare that all these skills reside in one individual. Of necessity, this confluence of skills in an individual is more likely to occur in small companies and small projects (those that are resource challenged), and that can be limited in the various types of software employed.

1.5 Plan of the Course

PASW Modeler can be thought of as a “work bench,” combining multiple tools and technologies to support the process of data mining. The course is structured roughly along the phases of the CRISP-DM process model. Because we don’t have a specific data-mining project to complete (although we will focus, for the most part, on one data file), we won’t discuss any further the Business Understanding phase, but we will cover the other stages from Data Understanding to Deployment.

In the early lessons of this course we will develop a number of skills that can be used to perform Data Understanding, like reading data from diverse sources, browsing and visualizing the data using tables and graphs, how to handle missing values, etc.

Next we will pay attention to Data Preparation. Conceptually, we can distinguish three types of data manipulation:

- Manipulating records, like sorting and selecting records
- Manipulating fields, like deriving new fields
- Manipulating files, like adding records or merging fields

Next, modeling techniques will be covered. PASW Modeler provides a number of so-called “supervised learning” and “unsupervised learning” techniques:

- Supervised techniques model an output variable based on one or more input variables. These models can be used to predict or forecast future cases where the outcome is unknown. Neural Networks, Rule Induction (decision trees), Linear Regression, and Logistic Regression are some of these supervised techniques
- Unsupervised techniques are used in situations where there is no field to predict but relationships in the data are explored to discover its overall structure. Kohonen networks, Two Step, K-means, and Apriori belong to this category.

This course focuses on supervised techniques, chiefly decision trees, to illustrate the general process of building and testing models. We will also use the automated modeling capability of PASW Modeler, and show how to combine models.

In the concluding lesson we will address the Deployment phase of the CRISP-DM model and also review some additional PASW Modeler features.

Lesson 2: The Basics of Using PASW Modeler

Objectives

- To provide an introduction to PASW Modeler
- To familiarize you with the tools and palettes available in PASW Modeler
- To introduce the idea of visual programming
- To discuss customization of the Nodes palette

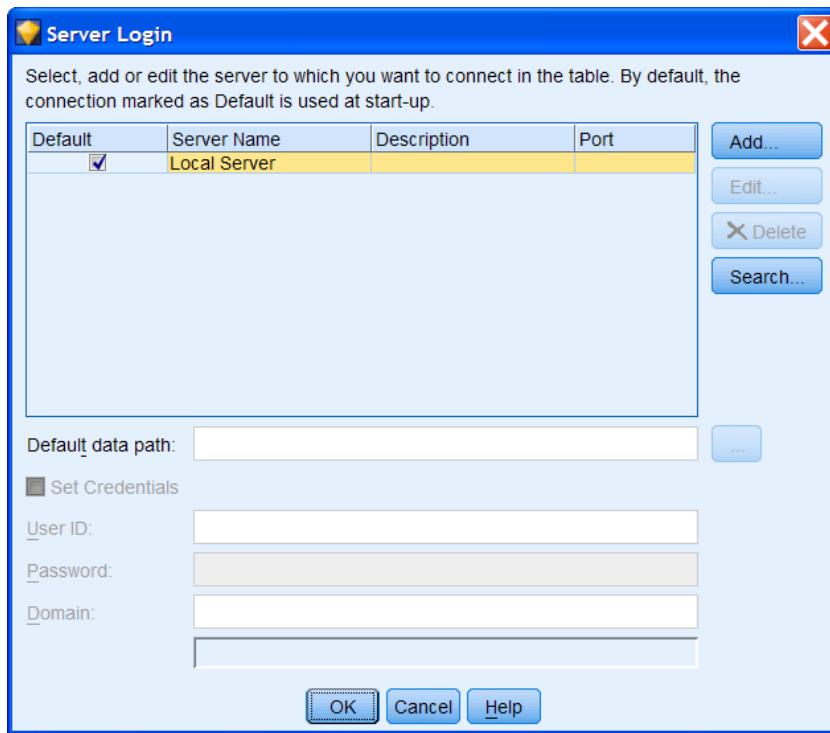
Note Concerning Data for this Course

Data for this course are assumed to be stored in the directory *c:\Train\ModelerIntro*. At IBM® training centers, the data will be located in the folder *c:\Train\ModelerIntro* of the training room PC. If you are working on your own computer, the *c:\Train\ModelerIntro* directory can be created on your machine and the data copied into that folder. (Note: if you are running PASW Modeler in distributed (Server) mode then the data should be copied to the server machine or the directory containing the data should be mapped from the server machine).

2.1 PASW Modeler and PASW Modeler Server

By default, PASW Modeler will run in local mode on your desktop machine. If PASW® Modeler Server has been installed, then PASW Modeler can be run in local mode or in distributed (client-server) mode. In this latter mode, PASW Modeler streams are built on the client machine, but executed by PASW Modeler Server.

Since the data files used in this training course are relatively small, we recommend you run in local mode. However, if you choose to run in distributed mode then make sure the training data are either placed on the machine running PASW Modeler Server or that the drive containing the data can be mapped from the server. To determine in which mode PASW Modeler is running on your machine, examine the connection status area of the PASW Modeler status bar (left-most area of status bar) or click Tools...Server Login (from within PASW Modeler) if the choice is active; if it is not active, then PASW Modeler Server is not licensed. See within the Server Login dialog whether the Connection is set to Local or Network. This dialog is shown in Figure 2.1.

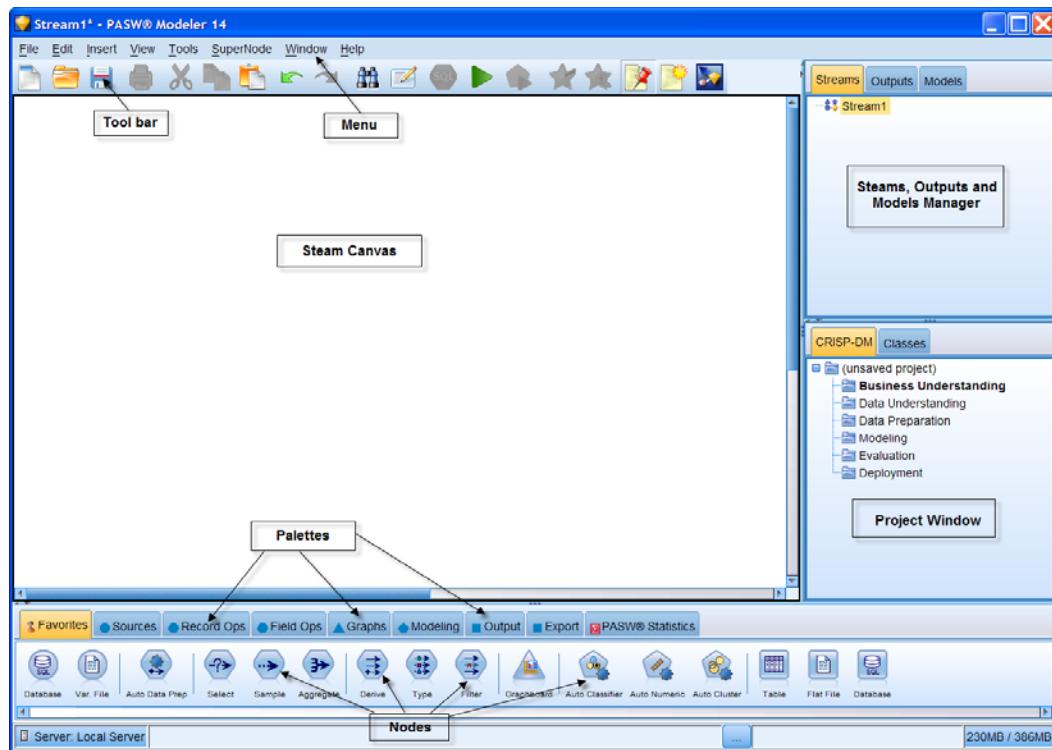
Figure 2.1 Server Login Dialog Box in PASW Modeler

2.2 Starting PASW Modeler

To run PASW Modeler:

From the Start button, click **All Programs...SPSS Inc....PASW Modeler 14... PASW Modeler 14**

At the start of a session, you see the PASW Modeler User Interface.

Figure 2.2 PASW Modeler User Interface

PASW Modeler enables you to mine data visually using the Stream Canvas. This is the main work area in PASW Modeler and can be thought of as a surface on which to place icons. These icons represent operations to be carried out on the data and are often referred to as nodes.

The nodes are contained in palettes, located across the bottom of the PASW Modeler window. Each palette contains a related group of nodes that are available to add to the data stream. For example, the Sources palette contains nodes that you can use to read data into your model, and the Graphs palette contains nodes that you can use to explore your data visually. Which icons are shown depends on the active, selected palette. The Modeling palette contains many nodes so it is grouped into sub palettes.

The Favorites palette is a customizable collection of nodes that a data miner uses most frequently. It contains a default collection of nodes, but these can be easily modified within the Palette Manager (reached by clicking Tools...Manage Palettes).

Once nodes have been placed on the Stream Canvas, they can be linked together to form a stream. A stream represents a flow of data through a number of operations (nodes) to a destination that can be in the form of output (either text or chart), a model, or the export of data to another format (e.g., a Statistics data file or a database).

At the upper right of the PASW Modeler window, there are three types of manager tabs. Each tab (Streams, Outputs, and Models) is used to view and manage the corresponding type of object. You can use the Streams tab to open, rename, save, and delete streams created in a session. PASW Modeler output, such as graphs and tables, are stored in the Outputs tab. You can save output objects directly from this manager. The Models tab is the most important of the manager tabs as it contains the results of the machine learning and modeling conducted in PASW Modeler. These models can be browsed directly from the Models tab or added to the current stream displayed in the canvas.

At the lower right of the PASW Modeler window is the Projects window. This window offers you a best-practice way to organize your data mining work. The CRISP-DM tab helps you to organize streams, output, and annotations according to the phases of the CRISP-DM process model (mentioned in Lesson 1). Even though some items do not typically involve work in PASW Modeler, the CRISP-DM tab includes all six phases of the CRISP-DM process model so that you have a central location for storing and tracking all materials associated with the project. For example, the Business Understanding phase typically involves gathering requirements and meeting with colleagues to determine goals rather than working with data in PASW Modeler. The CRISP-DM tab allows you to store your notes from such meetings in the Business Understanding folder of a project file for future reference and inclusion in reports.

The Classes tab in the Project window organizes your work in PASW Modeler categorically by the type of objects created. Objects can be added to any of the following categories:

- Streams
- Nodes
- Generated Models
- Tables, Graphs, Reports
- Other (non-PASW Modeler files, such as slide shows or white papers relevant to your data mining work)

There are eight menu choices in the PASW Modeler menu bar:

- File allows the user to create, open and save PASW Modeler streams and projects. Streams can also be printed from this menu.
- Edit allows the user to perform editing operations: for example copy/paste objects; clear manager tabs; edit individual nodes.
- Insert allows the user to insert a particular node, as alternative to dragging a node from the palette.
- View allows the user to toggle between hiding and displaying items (for example: the toolbar or the Project window).
- Tools allows the user to manipulate the environment in which PASW Modeler works and provides facilities for working with Scripts.
- Supernode allows the user to create, edit and save a condensed stream.
- Window allows the user to close related windows (for example, all open output windows), or switch between open windows.
- Help allows the user to access help on a variety of topics or view a tutorial.

2.3 Using the Mouse

When working with PASW Modeler, the mouse plays an important role in performing most operations. PASW Modeler takes advantage of the middle button in a three-button mouse (or a mouse with a scroll wheel), yet also works with a standard two-button mouse.

There are alternatives to using the mouse, e.g., using function keys or menus. Throughout this course, however, we will mainly use the mouse in our demonstrations.

2.4 Visual Programming

As mentioned earlier, data mining is performed by creating a stream of nodes through which the data pass. A stream, at its simplest, will include a source node, which reads the data into PASW Modeler, and a destination, which can be an output node, such as a table, a graph, or a modeling operation.

When building streams within PASW Modeler, mouse buttons are used in the following ways:

Left button	Used for icon or node selection, placement and positioning on the Stream Canvas.
Right button	Used to invoke Context (pop-up) menus that, among other options, allow editing, renaming, deletion and execution of the nodes.
Middle button (optional)	Used to connect two nodes and modify these connections. (When using a two-button mouse, you can right-click on a node, select Connect from the context menu, and then click on the second node to establish a connection.)

Note

In this guide, the instruction to “click” means click with the primary (usually the left) mouse button; “right-click” means to click with the secondary (usually the right) mouse button; “middle-click” means to click with the middle mouse button.

Adding a Node

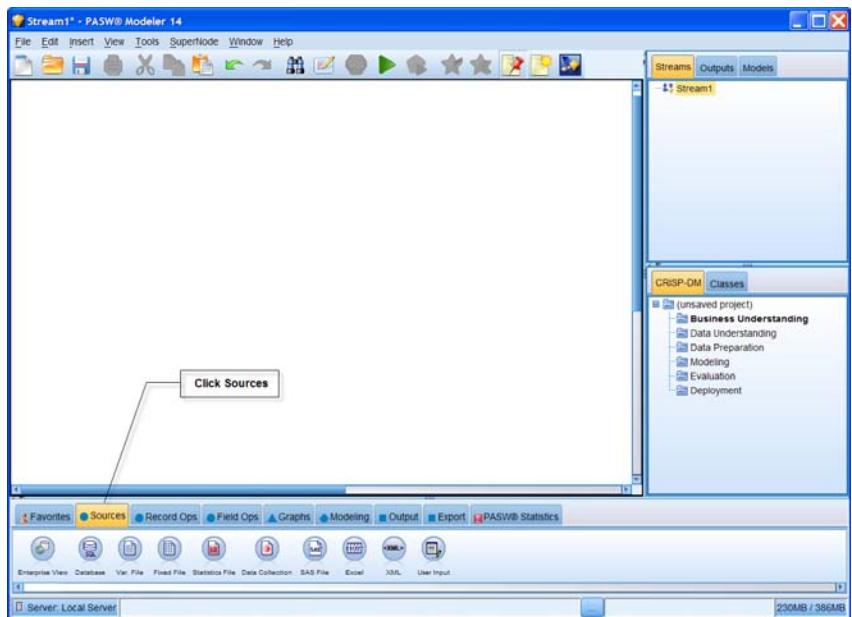
To begin a new stream, a node from the Sources palette needs to be placed on the Stream Canvas. There are three ways to add nodes to a stream from the nodes palette:

- Double-click a node on the palette. Note: Double-clicking a node automatically connects it to the current stream.
- Drag and drop a node from the palette to the stream canvas.
- Click a node on the palette, and then click on the stream canvas.

In this example we will illustrate the third of these methods. We will also assume that data are being read from a previously saved Statistics data file (we will cover methods of reading data into PASW Modeler in the next lesson).

Activate the Sources palette by clicking the **Sources** tab

Figure 2.3 Sources Palette



Select the **Statistics File** node from the Sources palette by clicking



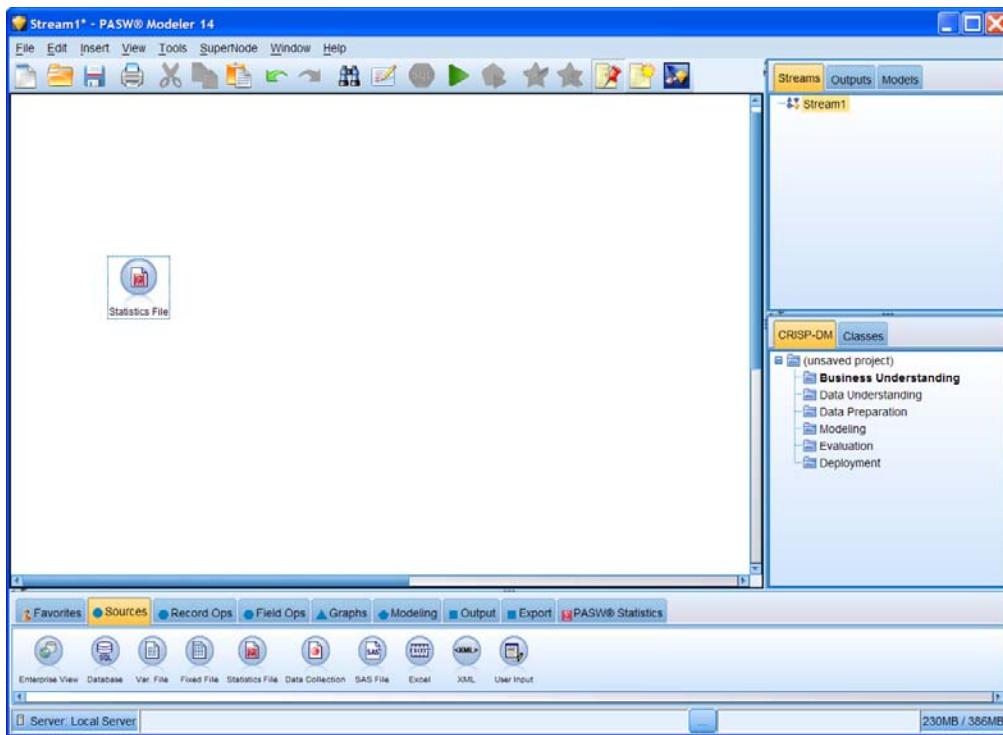
This will cause the icon to be highlighted.

Move the **cursor** over the **Stream Canvas**

The cursor will change to a positioning icon when it reaches the Stream Canvas.

Click **anywhere** in the Stream Canvas

A copy of the icon should appear on the Stream Canvas. This node now represents the action of reading data into PASW Modeler from a Statistics data file.

Figure 2.4 Placing a Node on the Stream Canvas

Moving a Node

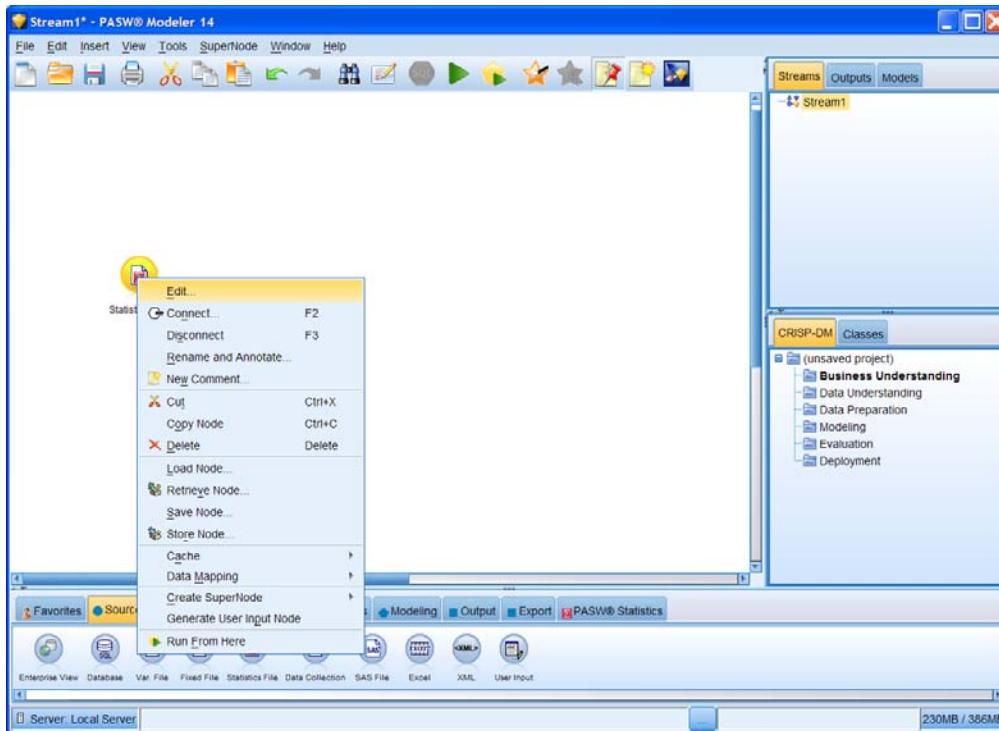
If you wish to move the node within the Stream Canvas, select it (using the left mouse button), and while holding this button down, drag the node to its new position.

Editing a Node

In order to view the editing options for a node, right-click on the icon to reveal a Context (pop-up) menu.



Right-click on the **Statistics File** node in the Stream Canvas

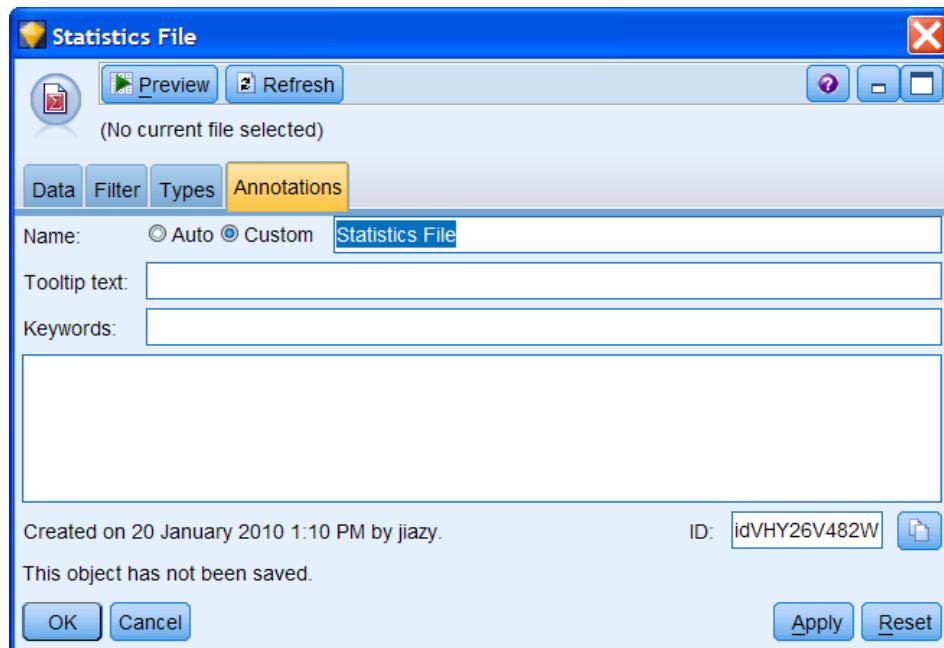
Figure 2.5 Context (Pop-up) Menu When a Source Node Is Right Clicked

The first choice, Edit, opens a dialog box specific to each node type. Double-clicking on the node also opens the dialog box. We will skip the edit options for the moment as we review the edit options for the Statistics File node in the next lesson.

Renaming and Annotating a Node

In order to label a node in the Stream Canvas with a more descriptive name:

Right-click on the **Statistics File** node
Click **Rename and Annotate** on the context menu

Figure 2.6 Rename and Annotate Dialog

We can specify a name and even tooltip text for the node. The tooltip text feature is extremely useful to aid in distinguishing between similar nodes on the stream canvas. In the text area, additional information can be attached to the node in order to aid interpretation or to act as a reminder to what it represents. The *Keywords* text box allows the user to enter keywords that are used in project reports and to search or track objects in the Model Manager repository (keywords can be specified for a stream, model, or output object in PASW Modeler). For the moment, however, we will cancel and look at the other options.

Click **Cancel**

Notice, that a node can be annotated as well by adding comment to it (right-click the node and select New Comment from the context menu; alternatively, select the node and click the Insert new comment  button in the Tool bar). In the same way, a stream can be annotated with comment.

Copy and Paste a Node

Copy and paste helps you to, for instance, duplicate a node (an action used later in this course). To duplicate a node:

- Right-click the node and select **Copy Node** from the context menu
- Right-click in an empty area of the Stream Canvas
- Select **Paste** from the context menu

A duplicate of the copied node will appear in the Stream Canvas. If needed, the node can be moved, renamed and annotated as described previously.

Deleting a Node

To delete a node:

- Right-click on the node

Select **Delete** from the context menu (alternatively, select the node and then press the Delete key)

2.5 Building Streams with PASW Modeler

Once two or more nodes have been placed on the Stream Canvas, they need to be connected to produce a stream. This can be thought of as representing a flow of data through the nodes.

To demonstrate this we will place a Table node in the Stream Canvas next to the Statistics File node (if you just deleted the Statistics File node, please replace it on the Stream Canvas). The Table node presents the data in a table format, similar to a spreadsheet view.

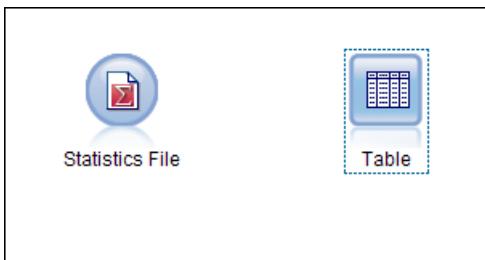
Click the **Output** tab to activate the Output palette



Click on the **Table** node  in the Output palette

Place this node to the **right** of the **Statistics File** node by clicking in the Stream Canvas

Figure 2.7 Unconnected Nodes



Connecting Nodes

There are a number of ways to connect nodes to form a stream: double-clicking, using the middle mouse button, or manually.

The simplest way to form a stream is to double-click nodes on the palette. This method automatically connects the new node to the currently selected node on the stream canvas (the one outlined in the blue-dotted box). For example, if the canvas contains a Database node, you can select this node and then double-click the next node from the palette, such as a Derive node. This action automatically connects the Derive node to the existing Database node.

To manually connect two nodes:

Right-click on the Statistics File node, and then select **Connect** from the context menu (note

the cursor changes to include a connection icon 

Click the **Table** node

Alternatively, with a three-button mouse:

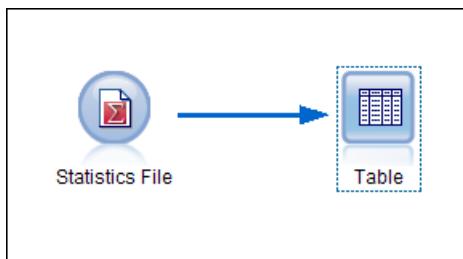
Click with the **middle** mouse button on the **Statistics File** node

While holding the middle button **down**, drag the cursor **over** the **Table** node

Release the middle mouse button

A connecting arrow appears between the nodes. The head of the arrow indicates the data flow direction. (Alternatively, if your mouse does not have a middle button, you can simulate this by pressing the Alt key while holding down the left mouse button and dragging from one node to another.)

Figure 2.8 Stream Representing the Flow of Data Between Two Nodes



Disconnecting Nodes

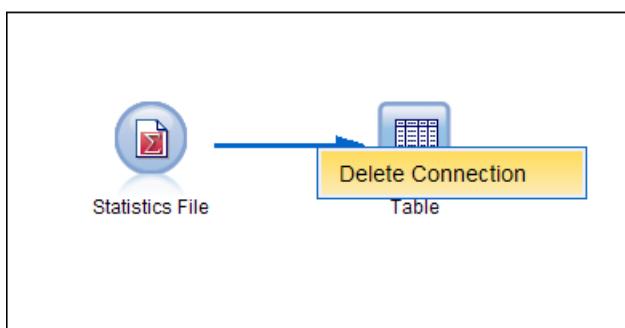
Nodes can be disconnected in several ways:

- By right-clicking on one of the nodes and selecting the Disconnect option from the context menu
- By right-clicking on the actual connection and selecting the Delete Connection option
- By double-clicking with the middle mouse button on one of the connected nodes (for intermediate nodes this will make existing arrows “bypass” the node)

We will demonstrate one of these alternatives.

Right-click on the **connecting arrow**
Click **Delete Connection**

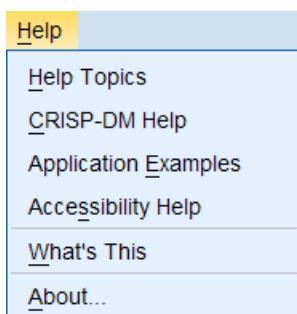
Figure 2.9 Disconnecting Nodes



2.6 Getting Help

Help can be accessed via the Help menu:

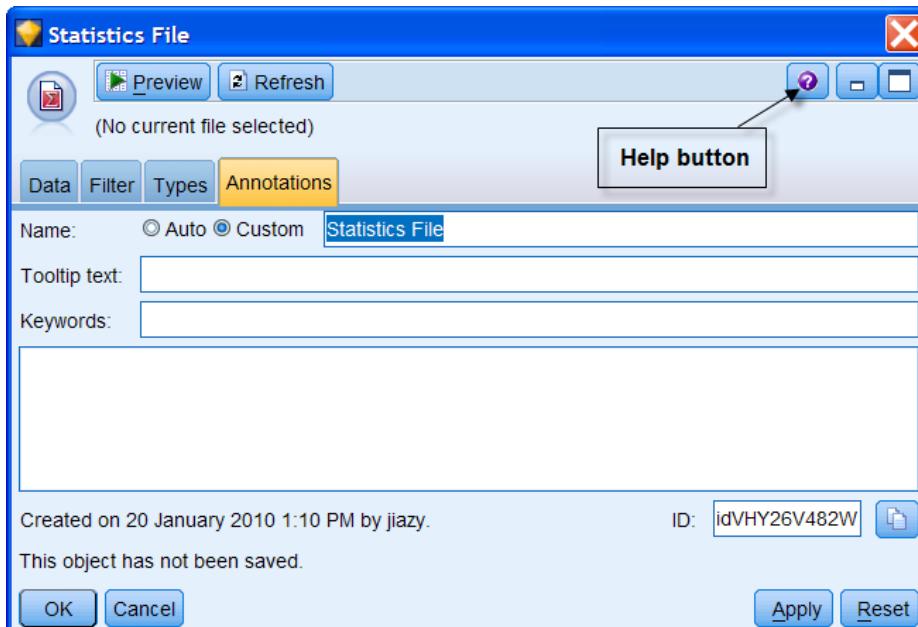
Click **Help**

Figure 2.10 Help Menu

The Help menu contains several options. The Help Topics choice takes you to the Help system. CRISP-DM Help gives you an introduction to the CRISP-DM methodology. Application Examples leads you to a variety of real-life examples of using common data mining techniques of data preparation and modeling. Accessibility Help informs you about keyboard alternatives to using the mouse. “What’s This” changes the cursor into a question mark and provides information about any PASW Modeler item you click.

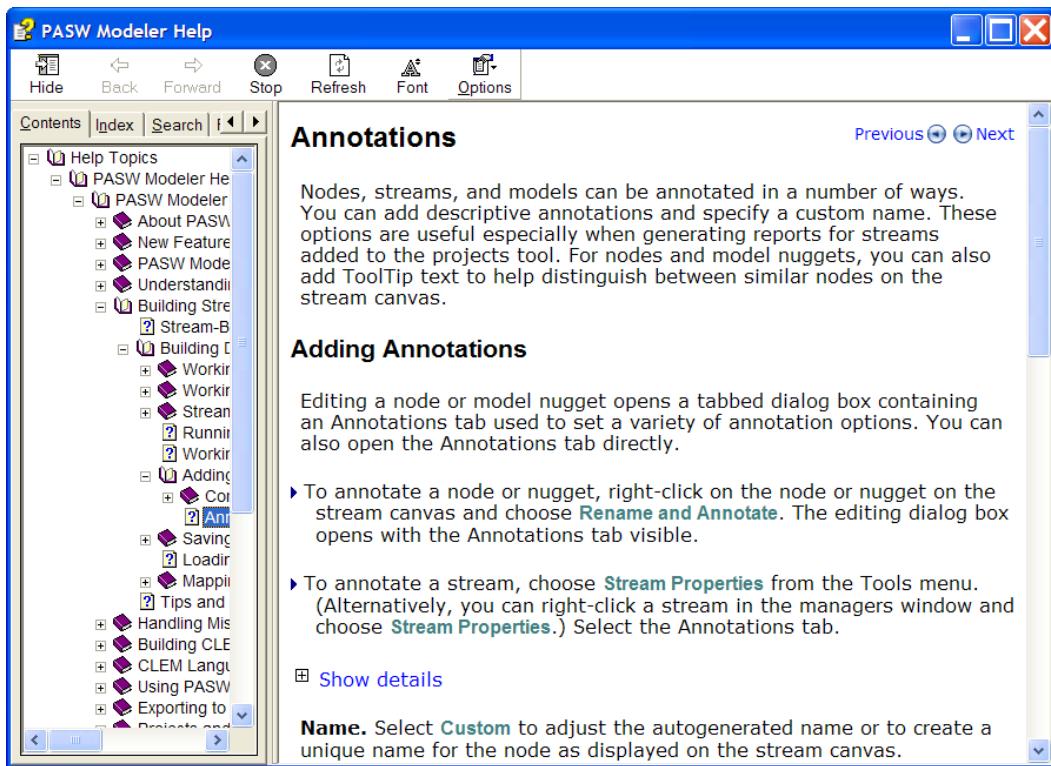
Besides the help provided by the Help menu, you always have context sensitive help available in whatever dialog box you are working. As an example we look at the help when we rename or annotate the Statistics File node.

Click away from the Help menu
Right-click the **Statistics File** node, and then click **Rename and Annotate**

Figure 2.11 Context Sensitive Help

Click the **Help** button

Information about the options in this specific dialog box can be accessed in this manner.

Figure 2.12 Context Sensitive Help on Annotating Nodes

Click the close button to close the Help window and to return to the Stream Canvas

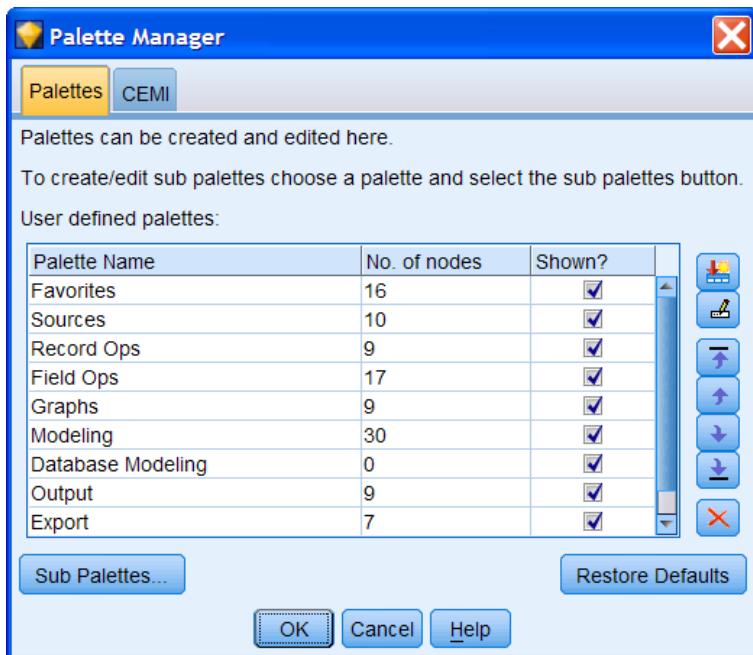
2.7 Customizing Palettes

The Nodes palettes are organized in tabs, such as the Source tab, Record Ops tab, and so further. They can be reorganized in two main ways:

1. You can use the Palette Manager to create unique tabs containing nodes common to your work, change the order of the tabs, or edit the default node selections on the Favorites tab.
2. You can change how palette tabs that contain sub palettes are displayed.

Let's access the Palette Manager to view the available options.

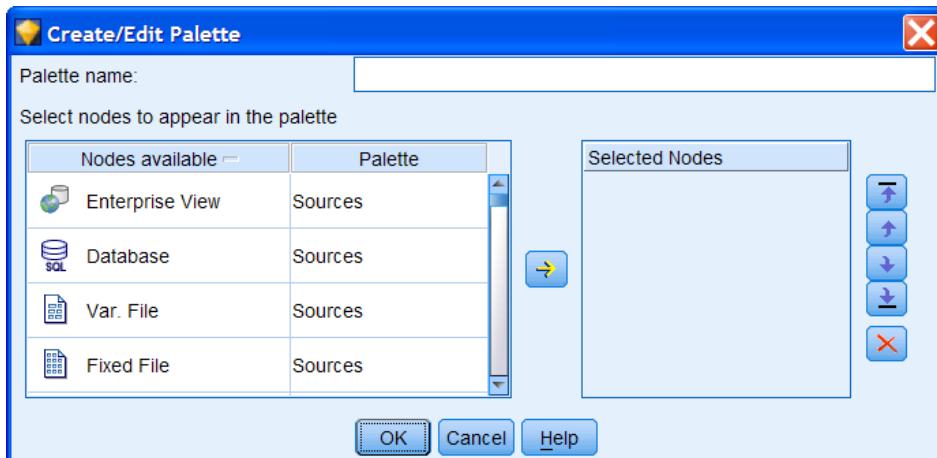
Click **Tools...Manage Palettes**

Figure 2.13 Palettes Manager

Each available palette is listed, along with its number of nodes, and a check box which will either display, or not, that palette. Palette order can be modified with the up and down arrow buttons. Sub palettes contained within a palette, such as those for the Modeling palette, can be edited by clicking on the *Sub Palettes* button. The CEMI tab contains options for displaying nodes created using the Clementine External Module Interface (CEMI). This functionality has been enhanced with the addition of the Extension Framework (CLEF).

Let's demonstrate how to create a custom tab called MyAnalysis.

Click on **Add a new item** button

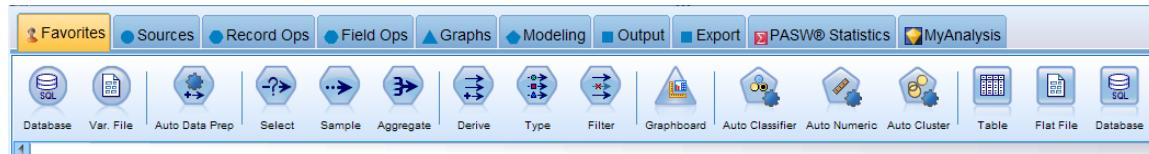
Figure 2.14 Create/Edit Palette Dialog

All the available nodes are listed on the left side. We need to give the new palette a name and then add some nodes.

- Add the text **MyAnalysis** in the Palette name box
- Click on the **Database** node and the arrow to add it to the Selected Nodes area
- Do the same for **Sample**, **Derive**, **Graphboard**, **Auto Classifier**, and **Neural Net** (not shown)
- Click **OK**, then click **OK** again

The new palette has been added to the right of existing palettes.

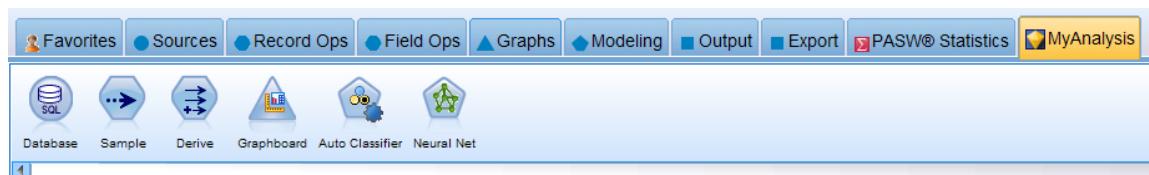
Figure 2.15 MyAnalysis Palette Added



Click on **MyAnalysis** tab

The nodes we selected are contained in the palette. We could use the Palette Manager to move this tab to the left since it contains nodes we use often.

Figure 2.16 Nodes in the MyAnalysis Tab



Sub Palettes

The Modeling tab contains more sub palettes because there are so many modeling nodes. You can modify which nodes appear on a sub palette, create new sub palettes, and display only some sub palettes.

Click the **Modeling** tab

There are four sub palettes on the Modeling palette. Each contains models that perform a similar or related analysis, such as classification (prediction), or segmentation (clustering). By default all nodes are displayed, but you can click on one of the sub palette buttons to display only those nodes.

Figure 2.17 Modeling Tab Sub Palettes



Click on the **Segmentation** button

Figure 2.18 Nodes in the Segmentation Sub Palette

All four of these nodes perform cluster analysis. You can manage sub palettes from the Palette Manager, including creating sub palettes for existing palettes.

With these methods you can modify the palette display of PASW Modeler to make your use of this interface more efficient.

We'll clean up by deleting the MyAnalysis palette we created.

Click **Tools...Manage Palettes**

Click on **MyAnalysis**

Click on the Delete selection button (not shown)

Click **OK**

Summary

In this lesson you have been given a brief tour of the PASW Modeler User Interface.

You should now be able to:

- Place an icon on the Stream Canvas to create a node
- Move, copy and delete nodes
- Rename and annotate nodes
- Connect two or more nodes to create a stream
- Obtain help within PASW Modeler
- Customize the Node palettes

Exercises

In these exercises, you have the opportunity to practice using the PASW Modeler interface and familiarize yourself with the menus and help facilities. No data file is used.

1. Start PASW Modeler.
2. Familiarize yourself with the PASW Modeler environment, including the menus and the help facilities. Search the help for any topics you are familiar with, or terms you have heard. Locate the help topic on the Select node and familiarize yourself with its operation.
3. Practice placing nodes on the Stream canvas.
4. Select the Var. File node from the Sources palette and place it on the Stream canvas.
5. Select the Table node from the Output palette and place it next to the Var. File node.
6. Connect these two nodes.
7. Disconnect the two nodes.
8. Delete one of the nodes in the stream.
9. Exit PASW Modeler without saving the stream.

Lesson 3: Reading Data Files

Objectives

- Data formats read by PASW Modeler
- Reading free-field text data files
- Reading Statistics data files
- Reading databases using ODBC
- Reading Excel files
- Viewing the data
- Field measurement level
- Field role
- Saving PASW Modeler Streams
- Appendix: Reading fixed-field text data files
- Appendix: Reading Date fields

Data

In this lesson a data set in several different formats will be used to demonstrate the various methods of reading data into PASW Modeler. The data file contains information concerning the credit rating and financial position of 514 individuals. The file also contains basic demographic information, such as marital status and gender.

In order to demonstrate ODBC we will use a Microsoft® SQL Server database, *Northwind.mdf*. The database has several tables we can select from, for example, Suppliers, Categories and Products.

To discuss how PASW Modeler reads date fields we use the data file *fulldata.txt* which contains account information for customers of a financial institution.

3.1 *Reading Data Files into PASW Modeler*

PASW Modeler reads a variety of different file types, including data stored in spreadsheets and databases, using the nodes within the Sources palette.

Data can be read in from text files, in either free-field or fixed-field format, using the Var. File and Fixed File source nodes. Statistics and SAS data files can be directly read into PASW Modeler using the Statistics File and SAS File nodes. Excel files can be read directly with the Excel node.

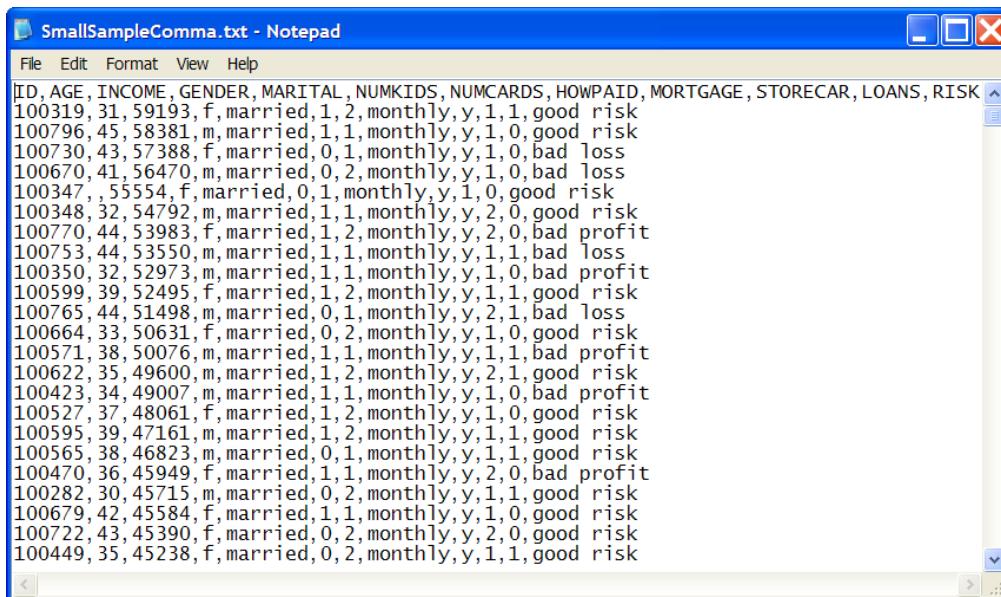
If you have data in an ODBC (Open Database Connectivity) source, you can use the Database source node to import data from server databases, such as Oracle™ or SQL Server™ and from a variety of other packages including, dBase™ and FoxPro™. PASW Modeler can also simulate data with the User Input node in the Sources palette. This node is useful for generating data for demonstrations or testing.

Further information on the types of data imports available in PASW Modeler can be found in the *PASW Modeler User's Guide*.

3.2 Reading Data from Free-Field Text Files

The Var. File node reads data from a free-field (delimited) text file. We demonstrate this by reading a comma-separated data file with field names in the first record. The figure below shows the beginning of the file (using Notepad).

Figure 3.1 Free-field Text File



The screenshot shows a Windows Notepad window titled "SmallSampleComma.txt - Notepad". The menu bar includes File, Edit, Format, View, and Help. The main window displays a series of comma-separated data records. The first few lines of the data are:

```
ID,AGE,INCOME,GENDER,MARITAL,NUMKIDS,NUMCARDS,HOWPAID,MORTGAGE,STORECAR,LOANS,RISK
100319,31,59193,f,married,1,2,monthly,y,1,1,good risk
100796,45,58381,m,married,1,1,monthly,y,1,0,good risk
100730,43,57388,f,married,0,1,monthly,y,1,0,bad loss
100670,41,56470,m,married,0,2,monthly,y,1,0,bad loss
100347,,55554,f,married,0,1,monthly,y,1,0,good risk
100348,32,54792,m,married,1,1,monthly,y,2,0,good risk
100770,44,53983,f,married,1,2,monthly,y,2,0,bad profit
100753,44,53550,m,married,1,1,monthly,y,1,1,bad loss
100350,32,52973,m,married,1,1,monthly,y,1,0,bad profit
100599,39,52495,f,married,1,2,monthly,y,1,1,good risk
100765,44,51498,m,married,0,1,monthly,y,2,1,bad loss
100664,33,50631,f,married,0,2,monthly,y,1,0,good risk
100571,38,50076,m,married,1,1,monthly,y,1,1,bad profit
100622,35,49600,m,married,1,2,monthly,y,2,1,good risk
100423,34,49007,m,married,1,1,monthly,y,1,0,bad profit
100527,37,48061,f,married,1,2,monthly,y,1,0,good risk
100595,39,47161,m,married,1,2,monthly,y,1,1,good risk
100565,38,46823,m,married,0,1,monthly,y,1,1,good risk
100470,36,45949,f,married,1,1,monthly,y,2,0,bad profit
100282,30,45715,m,married,0,2,monthly,y,1,1,good risk
100679,42,45584,f,married,1,1,monthly,y,1,0,good risk
100722,43,45390,f,married,0,2,monthly,y,2,0,good risk
100449,35,45238,f,married,0,2,monthly,y,1,1,good risk
```

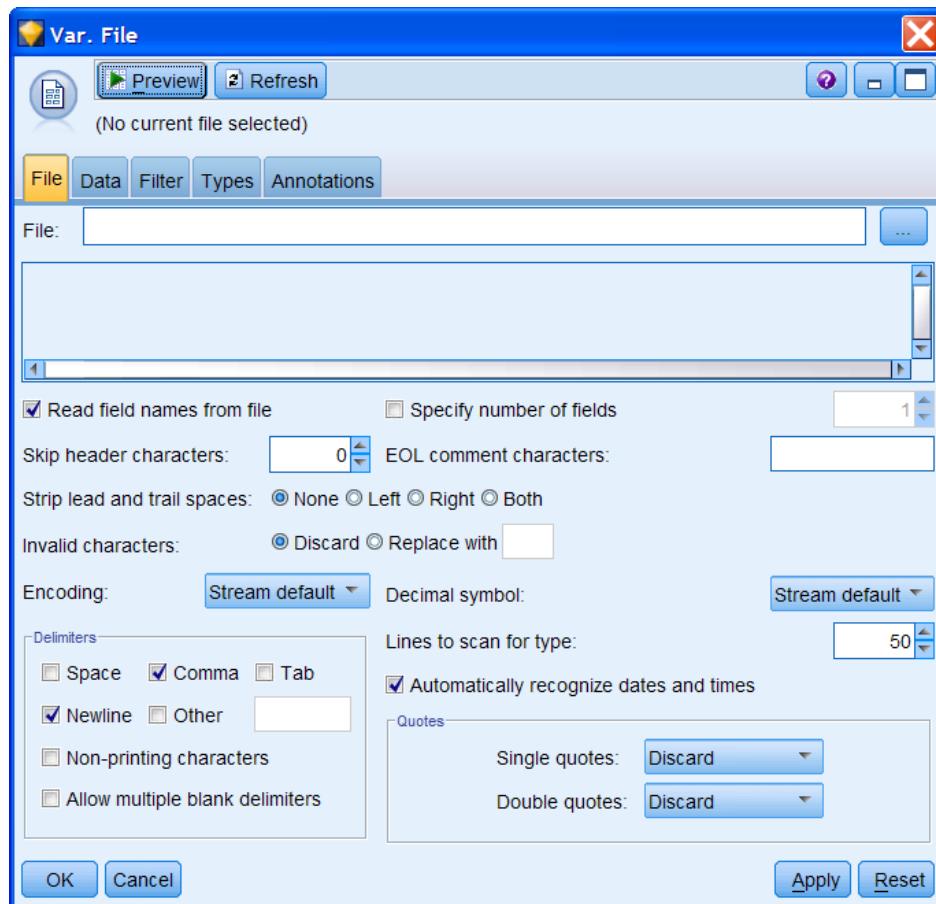
The file contains demographic information like age, income, gender, marital status, number of children and also information about the credit risk group of the person (good risk, bad profit and bad loss). Note that we have a mix of continuous fields (e.g., id, age, income) and categorical fields (e.g., gender, marital, risk).

We will read this file into PASW Modeler using the Var. File source node. Before we do this, however, we advise you to empty the stream canvas and to start from scratch.

If the Stream Canvas isn't empty, clear the Stream Canvas by choosing **Edit...Clear Stream**
Click the **Var. File** node in the Sources palette
Position the cursor on the **left** side of the Stream Canvas and **click once** (not shown)

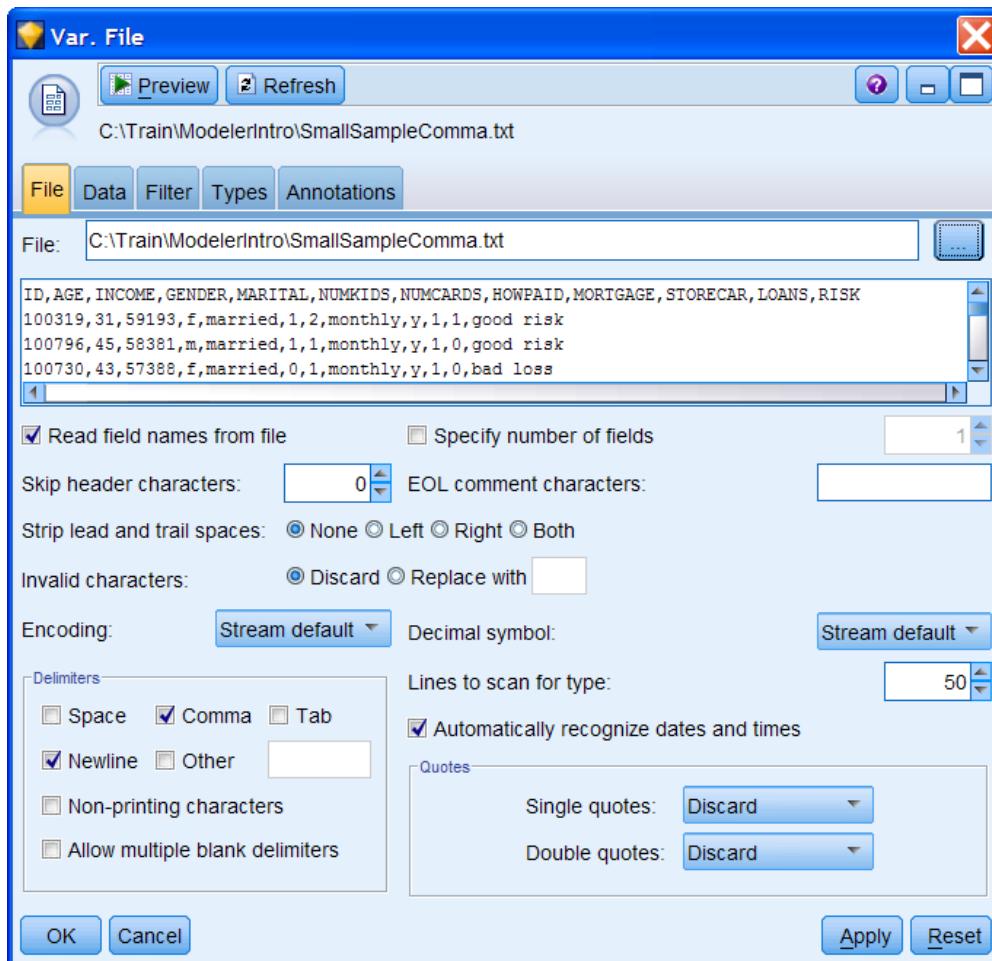
A copy of the icon should appear in the Stream Canvas. This source node represents the process of reading a text data file into PASW Modeler. To link this node to a specific file, it needs to be edited.

Right-click on the **Var. File** node, then click **Edit** (alternatively, double-click the **Var. File** node)

Figure 3.2 Variable File Node Dialog

The first thing to do is to specify the file name. The file list button is used to browse through directories and to identify the data file.

Click the file list button , and then navigate to the **c:\Train\Modeler\Intro** directory
Click **SmallSampleComma.txt**, and then click **Open**

Figure 3.3 Variable File Node Dialog

The Var. File dialog gives a preview of the first lines of data. Note, that the first line of data contains field names. These names can be read directly into PASW Modeler by checking the *Read field names from file* check box. By default, this option is already checked.

The *Skip header characters* text box allows you to specify how many characters are to be read and ignored before the first record begins. This is not relevant here.

The *EOL comment characters* text box allows the declaration of one or more characters that denote the start of a comment or annotation. When PASW Modeler comes across these characters in the file, it will ignore what follows until a new line is started. This is not relevant in our situation.

The *Specify number of fields* option allows you to specify how many fields are in each record in the data file. Alternatively, if all fields for a data record are contained on a single line and the number of fields is left unspecified, then PASW Modeler automatically calculates the number of fields.

Leading and trailing blanks can be removed from string fields in several ways using their respective options.

The characters used to separate the individual fields within the file are specified under *Delimiters*. White space (blanks) and tabs can also be declared as delimiters using their check box options. Single

and double quotes are dealt with using the drop-down menus and can either be discarded (*Discard*), matched in pairs and then discarded (*Pair & Discard*) or included as text within the field (*Include as text*).

If there are invalid characters in a file, which are null characters or any character that doesn't exist in the encoding for your operating system, you can either remove them or replace them with a one character valid value in the *Invalid characters* area.

Encoding refers to the text encoding method used to represent the text. The available choices are *Stream default*, *Server* (your own PC if not running from a server), and *UTF-8*, which is a Unicode standard that is portable across platforms and languages.

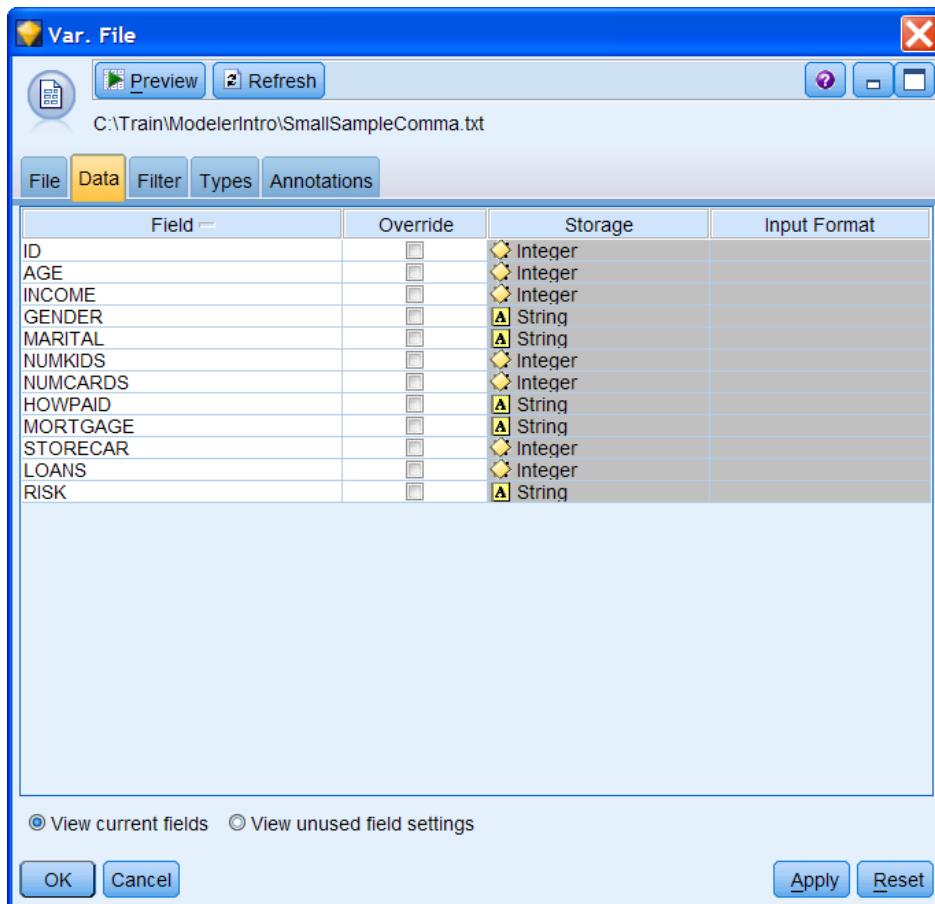
The *Decimal symbol* is the same as the Windows decimal symbol and could be set to either a comma or a period by using the drop-down menu.

By default PASW Modeler will scan 50 rows of data to determine the field measurement level. Field measurement level is one of the main concepts to be aware of when working with PASW Modeler and we will discuss this in detail later in this lesson.

Let's now see how PASW Modeler reads the field names from the specified data file.

Make sure that **Read field names from file** is checked
Click the **Data** tab

PASW Modeler scans the first 50 lines of data and reports the field names found. These field names are displayed within the dialog box shown below.

Figure 3.4 Data Tab Displaying Field Names

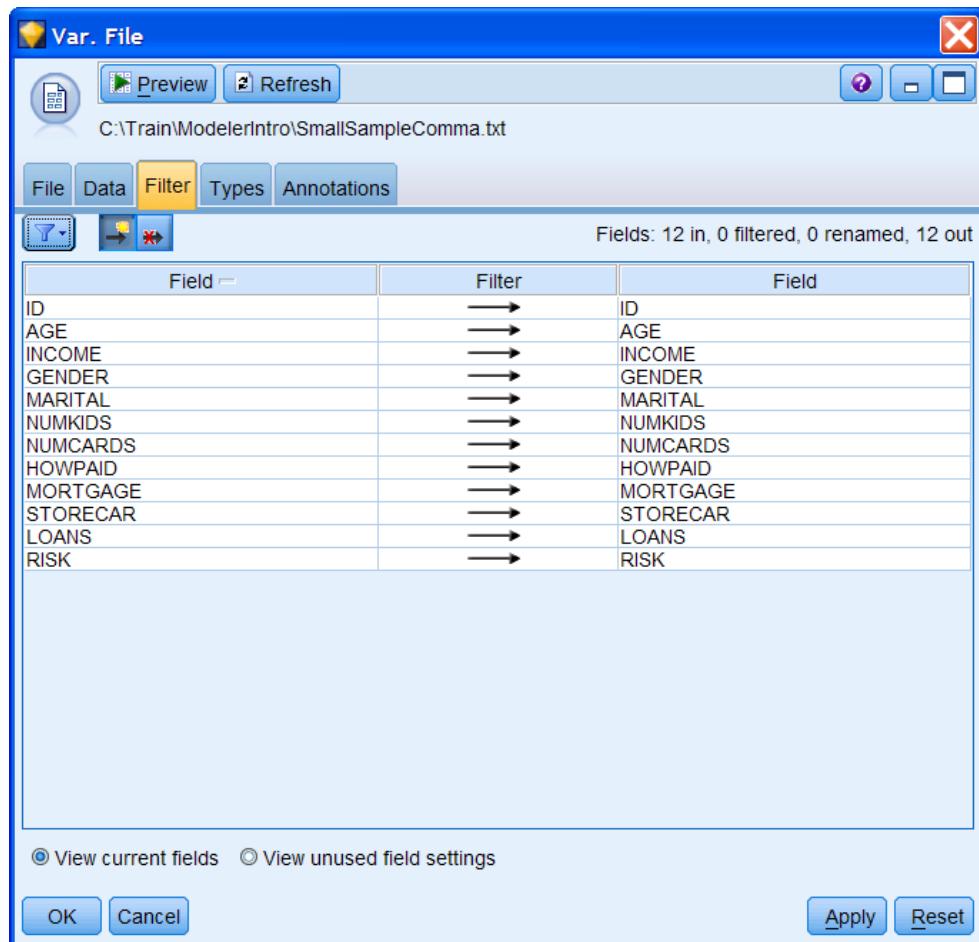
If there were no field names in the file and the *Get field names from file* option was not checked, PASW Modeler would assign the names, field1, field2, etc.

In Figure 3.4 we see *Override* and *Storage* columns. *Storage* describes the way data for a field is stored by PASW Modeler and this has implications for how the field can be used within PASW Modeler. For example, fields with integer or real data storage would be treated as continuous by modeling nodes (their measurement level would be continuous, unless changed). If you had numeric data values for a field that should be treated as categorical—for example numeric codes for marital status—one way to accomplish this would be to override the default data storage for such a field and set it to string. Storage options include string, integer (integer numeric), real (decimal numeric), time, date, timestamp, and unknown.

We can set the data storage for a field by checking its *Override* box, clicking in its *Storage* cell, and then selecting the storage from the drop-down list. Generally, the storage and measurement level determined automatically by PASW Modeler will be appropriate for your analyses, but you can change it if needed.

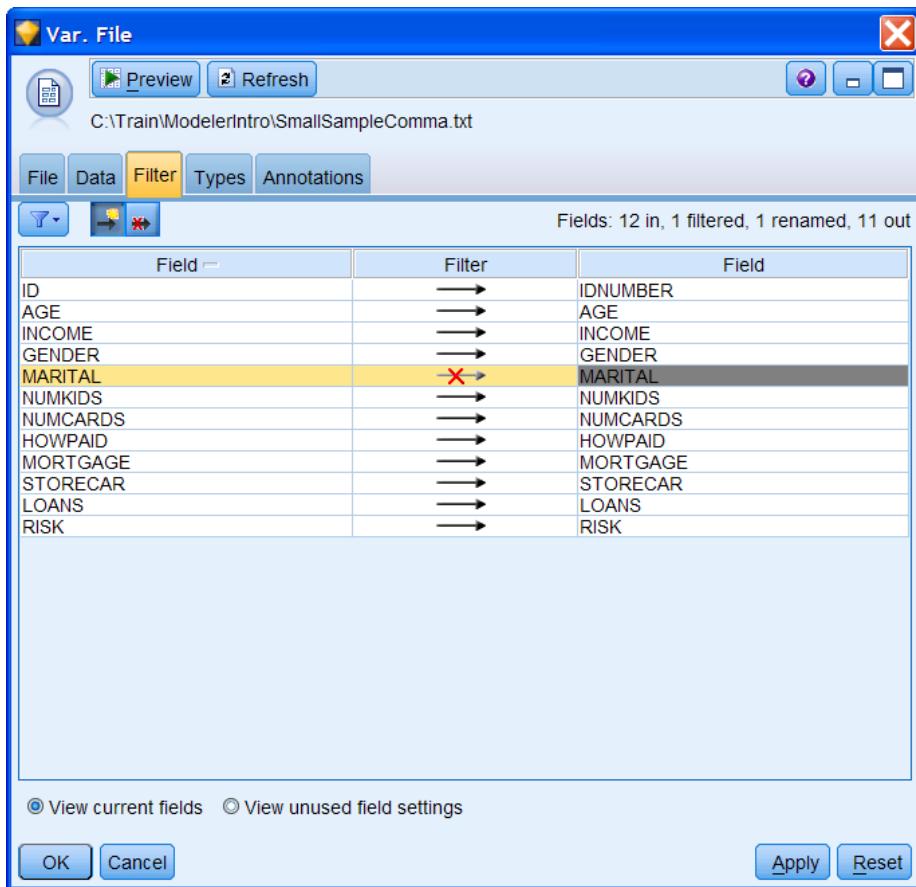
We might want to edit one or more field names or even decide that some fields should not be read into PASW Modeler. The Filter tab allows us to control this.

Click the **Filter** tab

Figure 3.5 Filter Tab in Var. File Node

The left column contains the field names as read from the data file. We can specify new names in the right column. The middle column shows an arrow that can be interpreted as “becomes.” As an example, suppose we would like to rename *ID* to *IDNUMBER* and would like to exclude *MARITAL*.

Double-click on **ID** in the right **Field** column
 Change **ID** to **IDNUMBER**
 Click once on the arrow in the **MARITAL** row

Figure 3.6 Changing Field Names and Dropping Fields

ID is renamed *IDNUMBER*. The crossed red arrow in the *MARITAL* row indicates that data for *MARITAL* won't be read into PASW Modeler.

Within the Filter tab, you can sort the fields (just click on column header Field), exclude all fields at once by clicking the button or include all fields at once by clicking the button. Furthermore, the filter menu options button gives access to numerous filter options such as: including/excluding all fields, toggling between fields, removing or renaming duplicates automatically and truncating fieldnames.

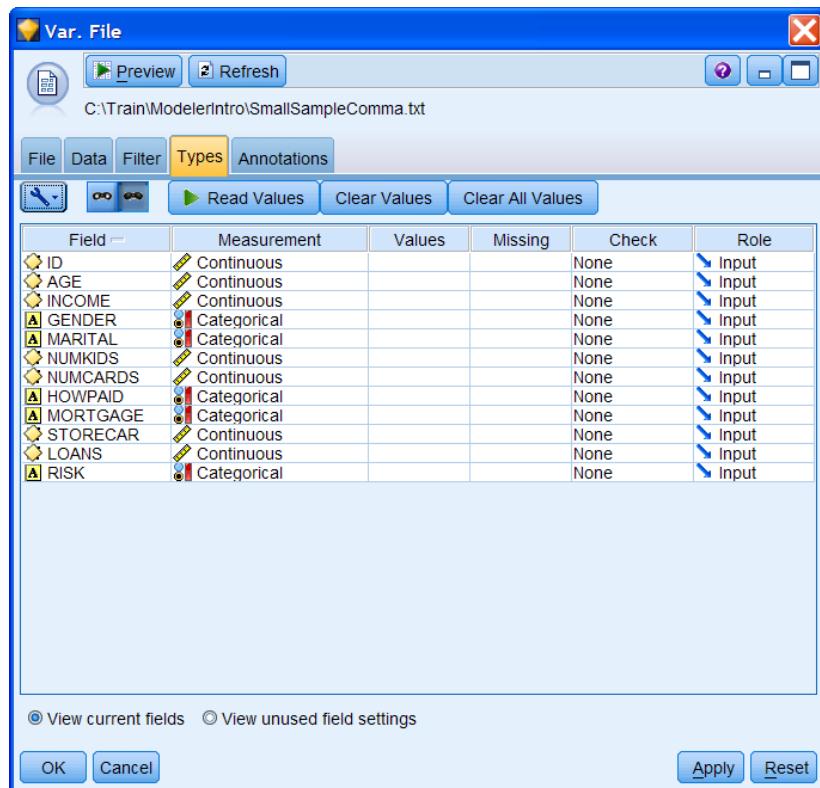
As an example we will undo the previous changes.

- Click the button and select **Include All Fields**
- Click the button and select **Use Input Field Names**

The window should be the same as it was prior to changing *ID* into *IDNUMBER* and excluding *MARITAL* (not shown).

The last tab to review in this window is the Types tab. This tab displays properties of the fields.

- Click the **Types** tab

Figure 3.7 Types Tab in Var. File Dialog

Field measurement level determines, in general, how the field will be used by PASW Modeler in data manipulation and modeling. Initially, fields with numeric values are typed as Continuous, and fields with string values are typed as Categorical. For the moment we will postpone a detailed discussion about measurement levels. Later in the lesson we will elaborate upon measurement levels and see how they can be set manually or assigned automatically by PASW Modeler as the data values are read.

The display above is based on 50 lines of data and thus presents *partially instantiated* measurement levels; categorical or continuous are partially instantiated measurement levels. Measurement levels are *fully instantiated* when all data pass through the node while the field Values are set to Read or Read+. If any field measurement is incorrect, PASW Modeler allows you to set measurement level before a full data pass is made. This can eliminate an unnecessary pass through the data, which is valuable when dealing with large files.

Click OK

By clicking OK you will return to the Stream Canvas, where the Var. File source node will have been labeled with the file name (not shown).

3.3 First Check on the Data

Once a data source node has been positioned in the Stream Canvas and linked to the data file, it is often advisable to check whether PASW Modeler is accessing the data correctly. The nodes within the Output palette display data, provide summary reports and analyses, and export data from PASW Modeler.

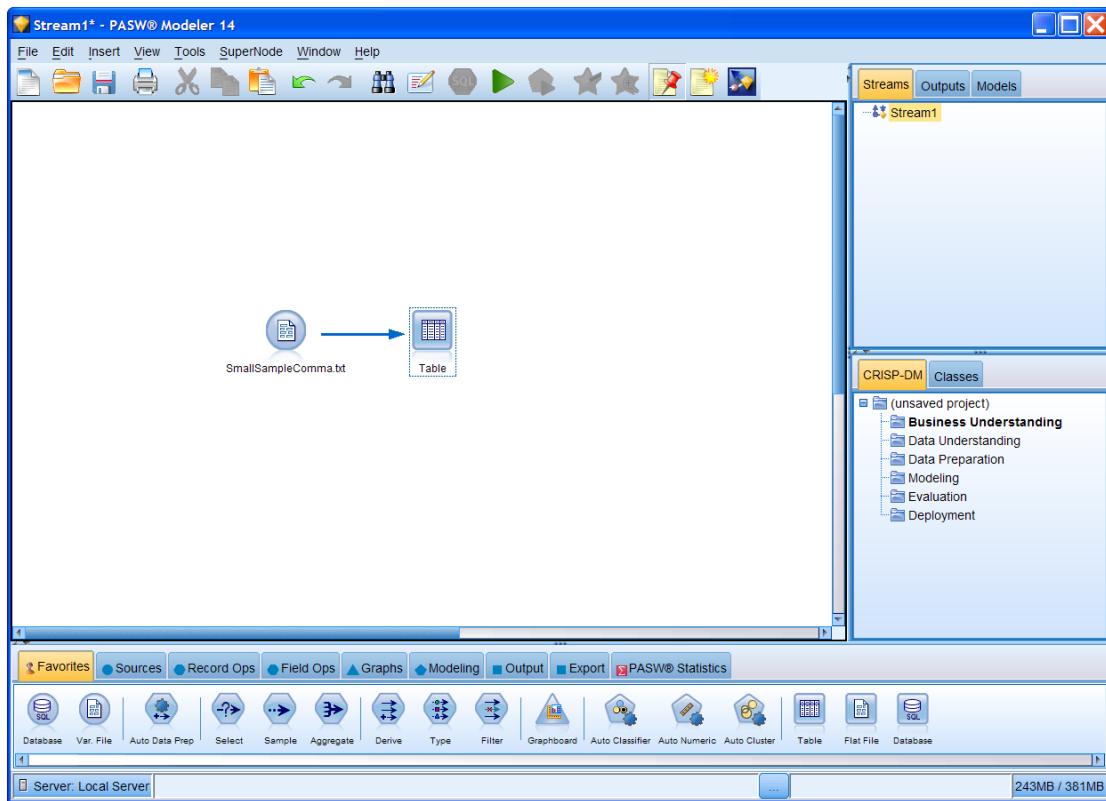
The Table node displays the data in tabular form, with one row per record and field names heading the columns. To view the data file, this node must be positioned in the data stream, downstream of the data source node that accesses the data file. Since it is an often-used node, the Table node is also present on the Favorites palette.

Click the **Favorites** palette, then click the **Table** node on the Favorites palette
Click to the **right** of the Var. File node named **SmallSampleComma.txt**

To connect the nodes:

Right-click the **Var. File** node (SmallSampleComma.txt), select **Connect** from the context pop-up menu, and then click the **Table** node in the Stream Canvas
(Alternatively, middle-click on the **Var. File** node (SmallSampleComma.txt) and drag the cursor to the **Table** node in the Stream Canvas)

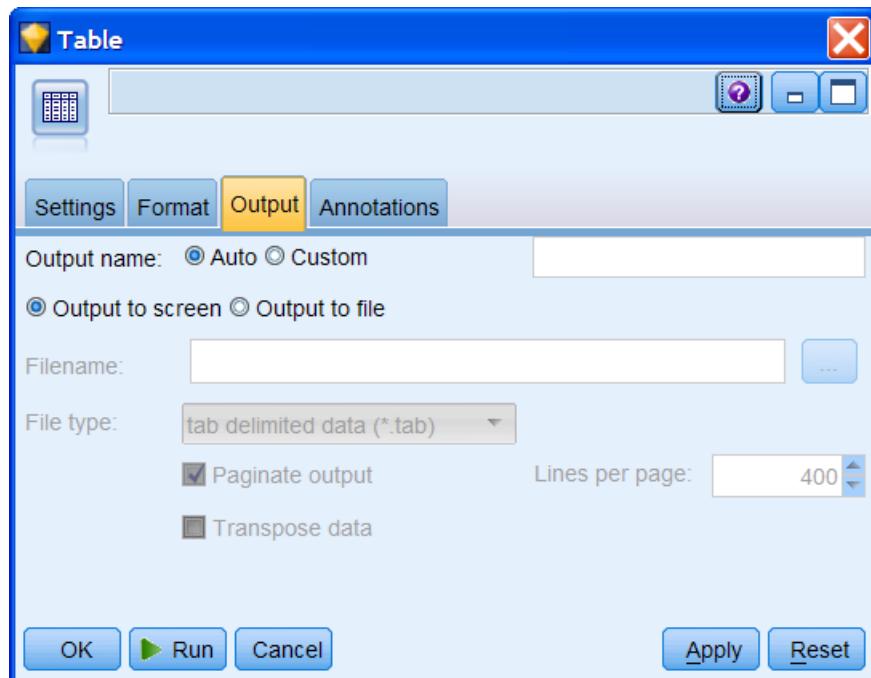
Figure 3.8 Var. File Node Connected to Table Node



An arrow appears connecting the source node to the Table node. The direction of the connecting arrow indicates the direction of data flow, passing from the source node into the Table output node.

The output style can be chosen by editing the Table node:

Double-click on the **Table** node
Click the **Output** tab

Figure 3.9 Table Node Dialog

Two output styles are available for the Table (and other display nodes) in the Output palette:

- Output to Screen: Displays the output on the screen
- Output to File: Sends output to file

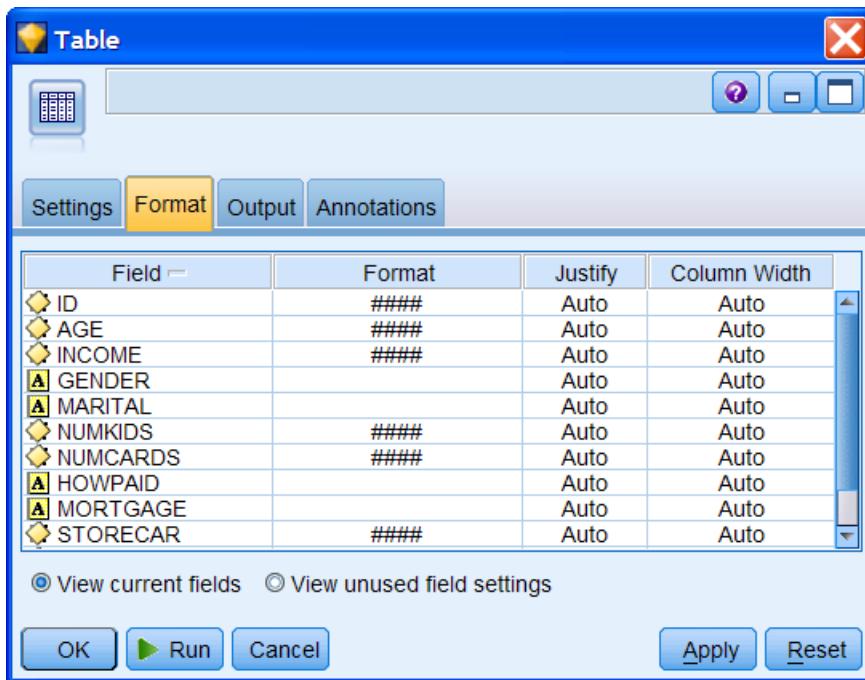
If you choose to send output to file several formats are available:

- tab delimited data (*.tab)
- comma delimited data (*.csv)
- html document (*.html)
- a PASW Modeler output object (*.cou)

We will stay with *Output to screen*, in order to see the table immediately.

Click the Format tab

Using the Format tab, the format of each individual field can be changed by selecting the field and choosing a specific format, choosing either Auto or a Manual width (which you can specify), and specifying that the field be left-, centered-, or right-justified within the column.

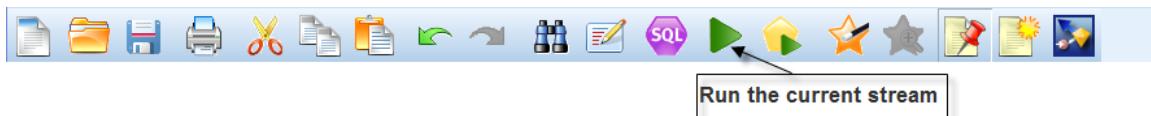
Figure 3.10 Format of Fields in Table Output

Finally, the Settings tab allows you to enter an expression in the *Highlight records where* box that identifies a set of records to be highlighted within the table itself (not shown).

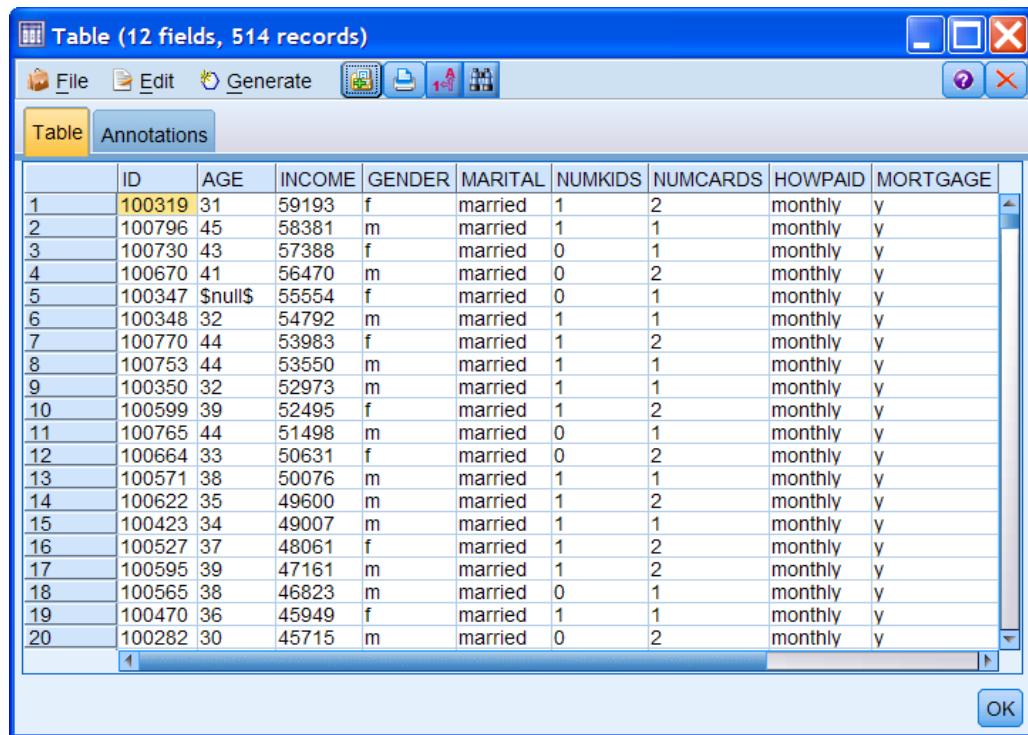
Once the dialog box has been edited, you can run the table by clicking the Run button. Alternatively, you can return to the Stream Canvas by clicking OK and then run the table by right-clicking on the Table node and selecting Run from the context menu. A third alternative, which we will use in this instance is to run the stream by using the *Run the current stream* button in the Toolbar.

Click **OK**

Click the **Run the current stream** button ➤ in the Toolbar

Figure 3.11 Run Button in the Toolbar

After having run the Table using any of the methods mentioned, the table window opens.

Figure 3.12 Table Window Showing Data from Text File


The screenshot shows the PASW Modeler Table window. The title bar reads "Table (12 fields, 514 records)". The menu bar includes File, Edit, Generate, and various icons. Below the menu is a tab bar with "Table" selected and "Annotations". The main area is a grid of data with 20 rows shown. The columns are labeled: ID, AGE, INCOME, GENDER, MARITAL, NUMKIDS, NUMCARDS, HOWPAID, and MORTGAGE. The data includes numeric values like 100319, 31, 59193, f, married, 1, 2, monthly, y, and string values like \$null\$. An "OK" button is at the bottom right.

	ID	AGE	INCOME	GENDER	MARITAL	NUMKIDS	NUMCARDS	HOWPAID	MORTGAGE
1	100319	31	59193	f	married	1	2	monthly	y
2	100796	45	58381	m	married	1	1	monthly	y
3	100730	43	57388	f	married	0	1	monthly	y
4	100670	41	56470	m	married	0	2	monthly	y
5	100347	\$null\$	55554	f	married	0	1	monthly	y
6	100348	32	54792	m	married	1	1	monthly	y
7	100770	44	53983	f	married	1	2	monthly	y
8	100753	44	53550	m	married	1	1	monthly	y
9	100350	32	52973	m	married	1	1	monthly	y
10	100599	39	52495	f	married	1	2	monthly	y
11	100765	44	51498	m	married	0	1	monthly	y
12	100664	33	50631	f	married	0	2	monthly	y
13	100571	38	50076	m	married	1	1	monthly	y
14	100622	35	49600	m	married	1	2	monthly	y
15	100423	34	49007	m	married	1	1	monthly	y
16	100527	37	48061	f	married	1	2	monthly	y
17	100595	39	47161	m	married	1	2	monthly	y
18	100565	38	46823	m	married	0	1	monthly	y
19	100470	36	45949	f	married	1	1	monthly	y
20	100282	30	45715	m	married	0	2	monthly	y

The title bar of the Table window displays the number of fields and records read into the table. If needed, scroll bars are available on the right side and bottom of the window allowing you to view hidden records and fields. Numeric fields are right-justified while string fields are left-justified.

Note the \$null\$ value for the 5th record in the AGE field (widen this column if necessary to see the full value). Looking back at the file SmallSampleComma.txt (Figure 3.1), we see that this person had no value for AGE, so PASW Modeler assigned AGE a missing value, represented by the value \$null\$. (We will discuss missing values in detail later.)

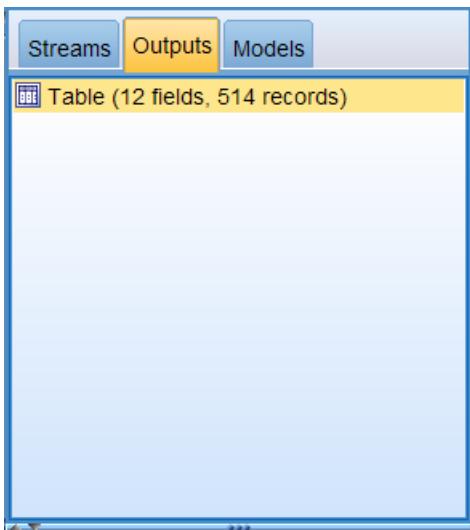
The File menu in the Table window allows you to save, print, export, or publish to the web the table. Using the Edit menu you can copy values or fields. The Generate menu allows you to automatically generate Select (data selection) and Derive (field calculation) nodes.

Now we have checked that the data file has been correctly read into PASW Modeler, we will close the window and return to the Stream Canvas.

Click **Close**  to close the Table window

Although we have closed the table, the table is still available in the Outputs manager, so we don't have to run the table again to see the data. To activate the Outputs manager:

Click the **Outputs** tab

Figure 3.13 Outputs Tab in Manager

The table is still available from the manager. In fact, each output produced (table or chart) will automatically be added as separate item in the Outputs tab and is available for later use.

For the moment we will clear this stream and take a look at reading other data sources. To clear the stream:

Click **Edit...Clear Stream**

To clear the output:

Right-click in an empty area in the **Outputs** tab
Click **Delete** (or **Delete All**) from the context menu
(Alternatively, click **Edit...Clear Outputs**)

3.4 Reading IBM® SPSS® Statistics Data Files

PASW Modeler can import and export Statistics data files. Importing is done using the Statistics File node in the Sources palette, while exporting involves the Statistics Export node in the Export palette and PASW® Statistics palette. As an example we will open the Statistics data file *SmallSample.sav*. The data are shown below in Statistics.

Typically, data in Statistics are encoded. For instance, *marital* does not have the values *divsepwid*, *married*, *single*, but the values 1, 2, 3. In Statistics we typically attach value labels to these codes to explain what the codes represent. Also when using Statistics, users often attach a label to the variable (field) name, so it's clear what the variable (field) represents. A label attached to the field is called a variable label in Statistics.

In Figure 3.14 and Figure 3.15, data and variables with and without labels are shown read into PASW Modeler.

Figure 3.14 Data File Read from Statistics: Variable Names and Data Values Displayed

	ID	AGE	INCOME	GENDER	MARITAL	NUMKIDS	NUMCARDS
1	100319	31	59193	1	2	1	2
2	100796	45	58381	2	2	1	1
3	100730	43	57388	1	2	0	1
4	100670	41	56470	2	2	0	2
5	100347	\$null\$	55554	1	2	0	1
6	100348	32	54792	2	2	1	1
7	100770	44	53983	1	2	1	2
8	100753	44	53550	2	2	1	1
9	100350	32	52973	2	2	1	1
10	100599	39	52495	1	2	1	2
11	100765	44	51498	2	2	0	1
12	100664	33	50631	1	2	0	2
13	100571	38	50076	2	2	1	1
14	100622	35	49600	2	2	1	2
15	100423	34	49007	2	2	1	1
16	100527	37	48061	1	2	1	2
17	100595	39	47161	2	2	1	2
18	100565	38	46823	2	2	0	1
19	100470	36	45949	1	2	1	1
20	100282	30	45715	2	2	0	2

Above we see the actual data values and variable names (field names) in the Statistics file.

Figure 3.15 Data File Read from Statistics: Variable and Value Labels Displayed

	ID number	Age in years	Income	Gender	Marital Status	# of dependent children
1	100319	31	59193	Female	married	1
2	100796	45	58381	Male	married	1
3	100730	43	57388	Female	married	0
4	100670	41	56470	Male	married	0
5	100347	\$null\$	55554	Female	married	0
6	100348	32	54792	Male	married	1
7	100770	44	53983	Female	married	1
8	100753	44	53550	Male	married	1
9	100350	32	52973	Male	married	1
10	100599	39	52495	Female	married	1
11	100765	44	51498	Male	married	0
12	100664	33	50631	Female	married	0
13	100571	38	50076	Male	married	1
14	100622	35	49600	Male	married	1
15	100423	34	49007	Male	married	1
16	100527	37	48061	Female	married	1
17	100595	39	47161	Male	married	1
18	100565	38	46823	Male	married	0
19	100470	36	45949	Female	married	1
20	100282	30	45715	Male	married	0

Instead of the values, we now see the value labels for some variables. We also see that *MARITAL* has the variable label *Marital Status* attached to it.

To read a Statistics data file into PASW Modeler, place the Statistics File node on the Stream Canvas.

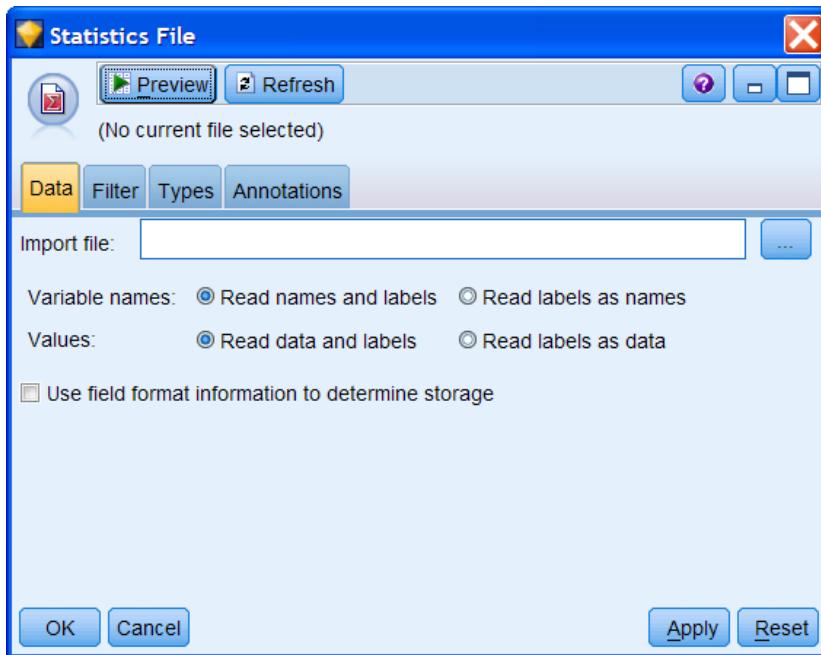
Click the **Statistics File** node from the Sources palette

Click in an open area on the left side of the Stream Canvas

To specify the Statistics data file to be read:

Double-click on the **Statistics File** node to edit it

Figure 3.16 Statistics Import Node Dialog



As in the previous examples, the file name and directory can be specified using the file list button (...) (or keyed in directly).

Select **SmallSample.sav** from C:\Train\ModelerIntro

The Statistics File dialog contains option buttons for how Variable Labels and Value labels should be imported from Statistics.

There are two choices for reading variables.

- **Read names and labels.** Select this to read both variable names and labels into PASW Modeler. By default, if this option is selected the variable names are displayed in the Type node. Variable labels may be displayed in charts, model browsers, and other types of output, depending on options specified in the Stream Properties dialog box. By default, the display of labels in output is disabled.
- **Read labels as names.** Select this to read the variable labels from the Statistics .sav file rather than the short field names, and use these labels as variable names in PASW Modeler.

There are two choices for reading data values.

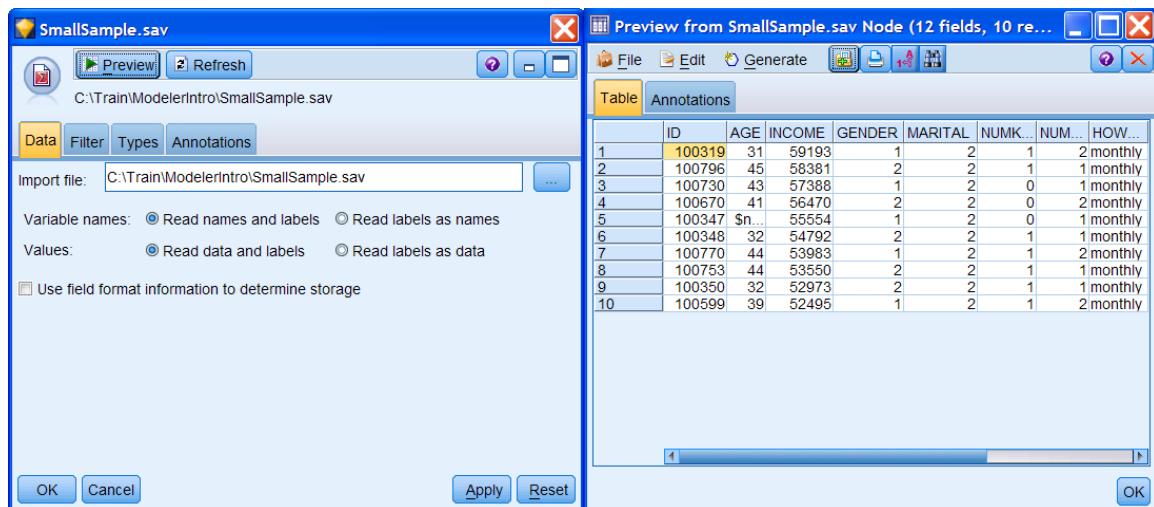
- **Read data and labels.** Select to read both actual values and value labels into PASW Modeler. By default, if this option is selected the values themselves are displayed in the Type

- node. Value labels may be displayed in the Expression Builder, charts, model browsers, and other types of output, depending on options specified in the Stream Properties dialog box.
- **Read labels as data.** Select if you want to use the value labels rather than the numeric codes used to represent the values. For example, selecting this option for data with a gender field with values labels of male and female, respectively, will convert the field to a string and import male and female as the actual values.

We will show the effect of these different choices. First, we use the default selections.

Click the **Preview** button

Figure 3.17 Statistics Import Dialog Box: Using Variable Names and Data Values



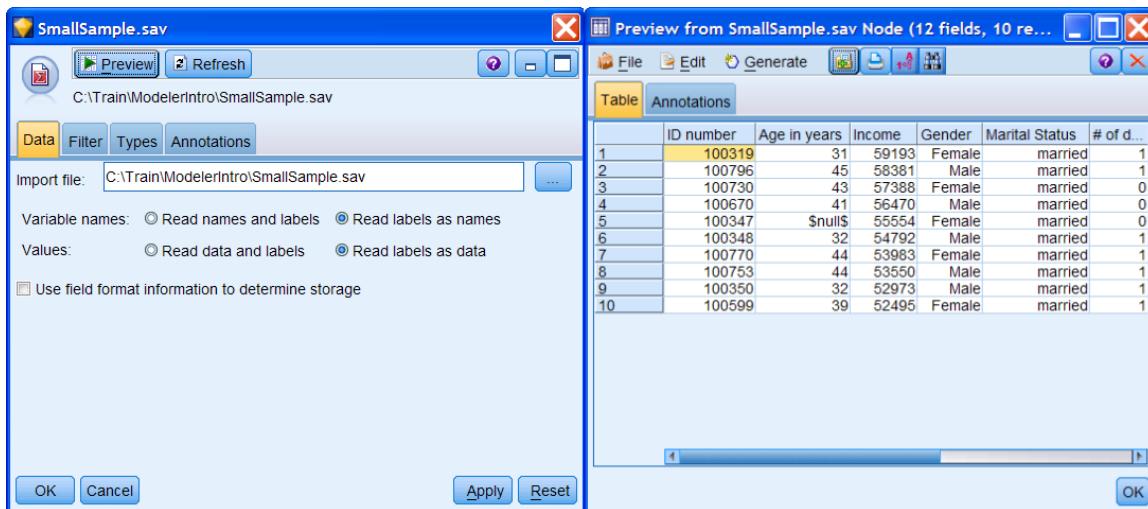
The field names in PASW Modeler are the same as the Statistics variable names, so the longer variable label is not displayed (the longer labels will be available for charts, models and some other output). This isn't a problem since the field names themselves are pretty clear. The data values in PASW Modeler are the same as the Statistics values. In general, having PASW Modeler read the values instead of the value labels is usually not recommended. The first reason is that identification information is essentially lost, though if you put the cursor on a labeled variable like marital (as shown in Figure 3.17) the value label will appear. A second reason, which perhaps is more serious, is that PASW Modeler by default will treat the field as continuous instead of categorical. If a field to be predicted is continuous but should be treated as categorical, this could well lead to potential problems later in the data-preparation and modeling phases if the modeling algorithm of choice requires a categorical outcome or, conversely, attempts to model the numeric values.

Close the **Preview** window

Go back to the **Statistics File** node

Click the **Read labels as names** and **Read labels as data** option buttons

Click the **Preview** button again

Figure 3.18 Statistics Import Dialog: Using Variable and Value Labels

When variable and values labels are imported, we have a more descriptive dataset that corresponds to the original Statistics file. It not only contains more informative values, but fields like *Gender* and *Marital Status* are correctly treated as categorical.

3.5 Reading Data Using ODBC

Using its Database source node, PASW Modeler can read data directly from databases and other data sources supporting ODBC (Open Database Connectivity protocol). Within the Database node, fields can be selected from a database table or SQL can be used to access data. Before reading data in this way, ODBC drivers must be installed (drivers for a number of databases are included in the Statistics Data Access Pack on the PASW Modeler CD) and data sources must be defined. We will illustrate how to declare a database as a data source within Windows® and then how to access that database using the Database source node.

The data tables we will access are found in the SQL Server database *Northwind.mdf*. The database contains several tables we can select from, such as Suppliers, Categories and Products.

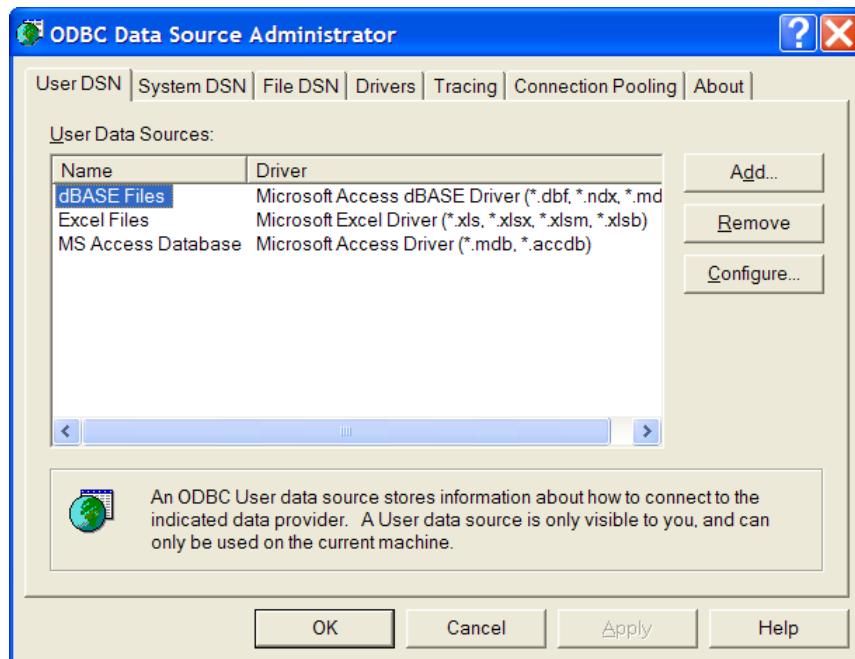
Declaring a Data Source

Before a database can be accessed via the ODBC method, it must be declared as an ODBC User Data Source using the ODBC Data Source Administrator, found within the Windows Control panel. An ODBC User Data Source stores information about how to connect to an indicated data provider. In this section we will demonstrate how to declare a database as an ODBC User Data Source using the SQL Server database *Northwind.mdf*, working with Windows XP.

Go to the **Start** menu
 Click **All Programs...Control Panel**
 Click **Administrative Tools**

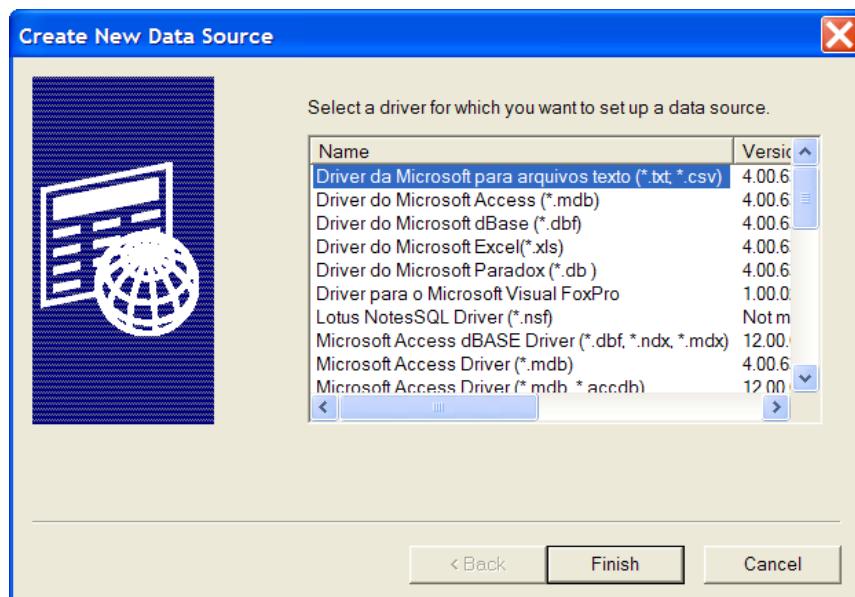
If ODBC is installed on your machine, there will be an icon labeled Data Sources (ODBC) within the Administrative Tools.

Double-click on the icon labeled **Data Sources (ODBC)**

Figure 3.19 ODBC Data Source Administrator

To declare a database as a Data Source we must add it to the User Data Sources list.

Click on the **Add** button

Figure 3.20 Create New Data Source Dialog

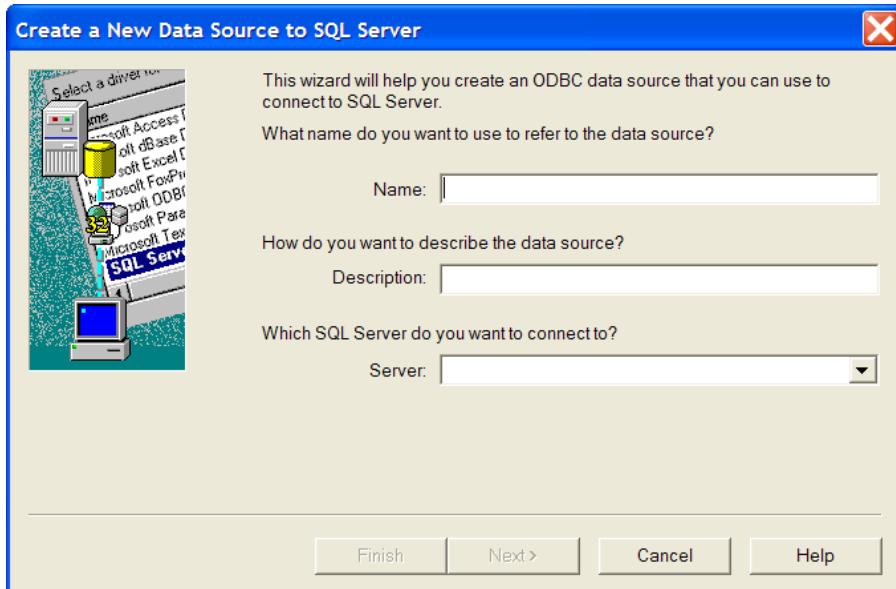
This dialog box contains a list of drivers that are installed on your machine.

From the driver list select **SQL Server** (not shown)

Click on the **Finish** button

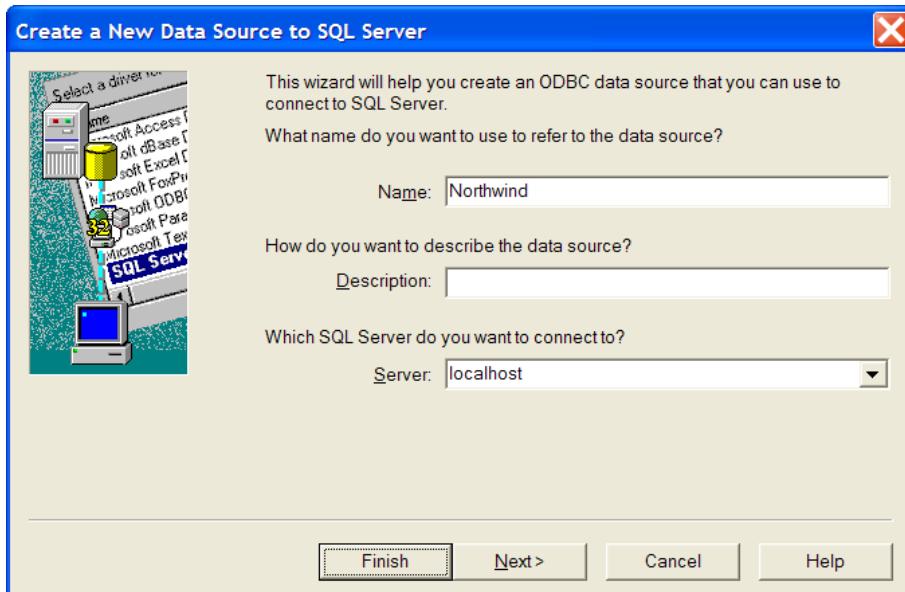
This will start up the Data Source Wizard. On the first screen, you can specify the name and description of the data source, and the name of the server running SQL Server. You can enter *localhost* in the server box if SQL Server is installed on the computer you are using. In this example, we will connect to the local copy of SQL Server.

Figure 3.21 Microsoft SQL Server Dialog Used to Set Up a Data Source # 1



Type the name **Northwind** in the **Data Source Name** text box
Type **localhost** in the **Server** box

Figure 3.22 Create a New Data Source to SQL Server dialog # 1

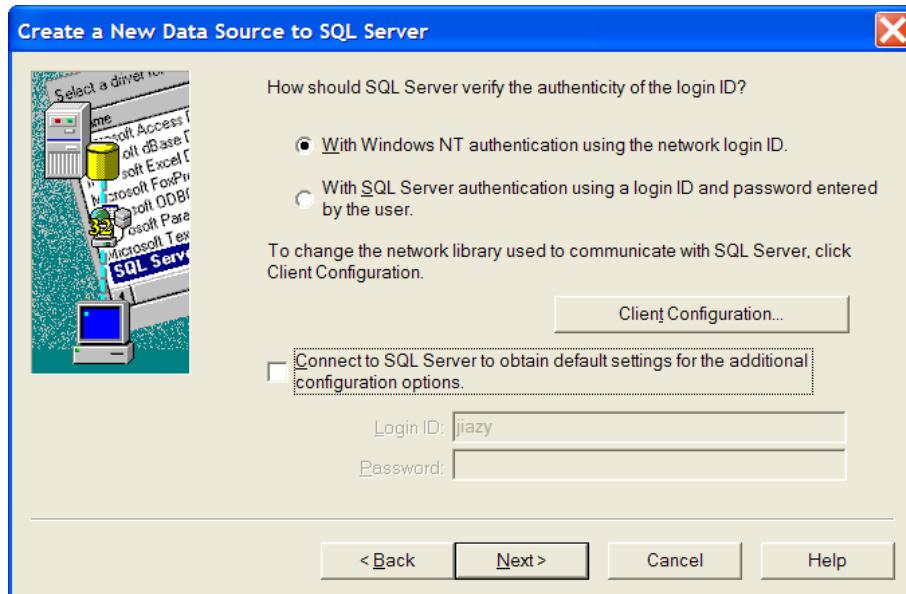


Click **Next**

On the second dialog, you can specify the method of authentication and set up SQL Server advanced-client entries and set up the login and password the SQL Server driver will use to connect to SQL Server while configuring the data source.

Uncheck the **Connect to SQL Server to obtain default settings for the additional configuration options** box

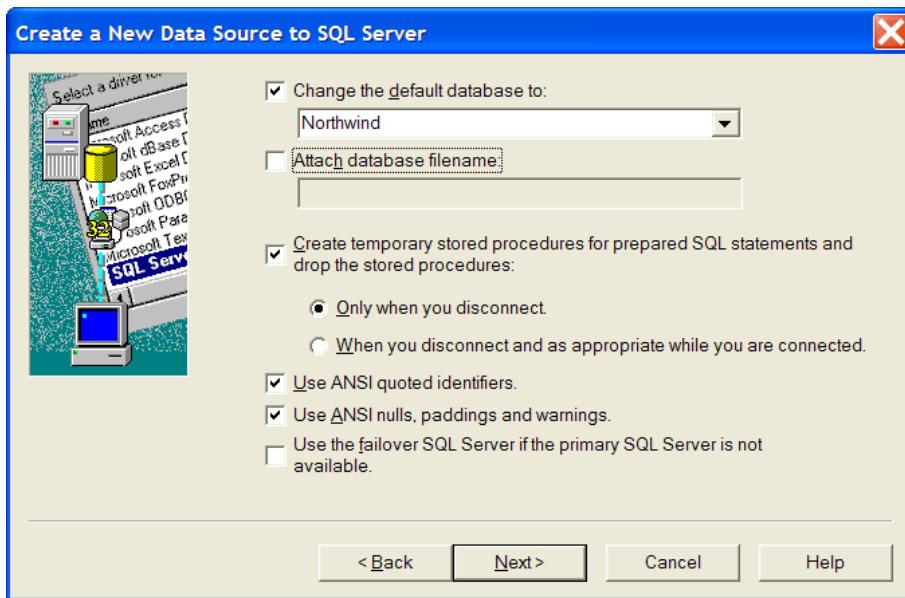
Figure 3.23 Create a New Data Source to SQL Server dialog # 2



Click Next

On the third dialog, you can specify the default database, how the driver should use stored procedures to support **SQLPrepare**, various ANSI options to be used by the driver, and whether to use a failover server.

Check the **Change the default database** to box and type **Northwind**

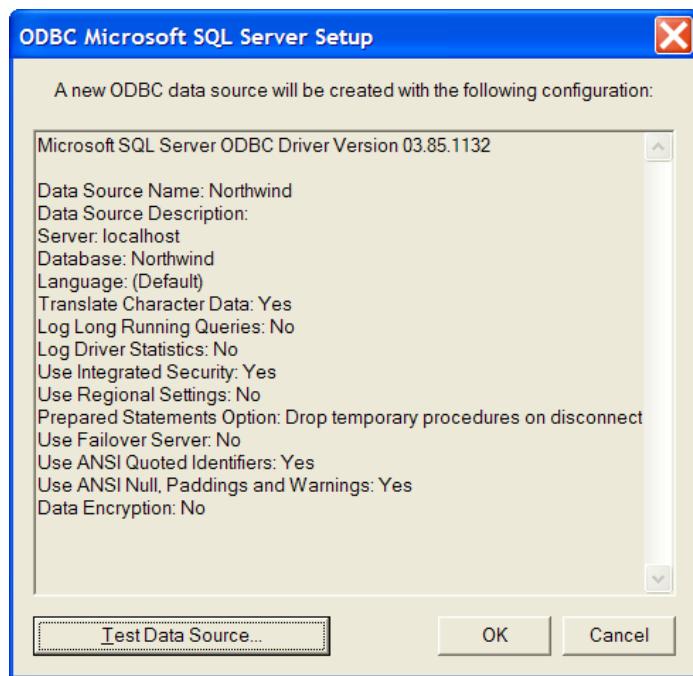
Figure 3.24 Create a New Data Source to SQL Server dialog # 3

Click Next

On the fourth screen of the wizard (not shown), you can specify the language to be used for Microsoft® SQL Server™ messages, character set translation, and whether the SQL Server driver should use regional settings. You can also control the logging of long-running queries and driver statistics settings. We will just take the defaults.

Click Finish

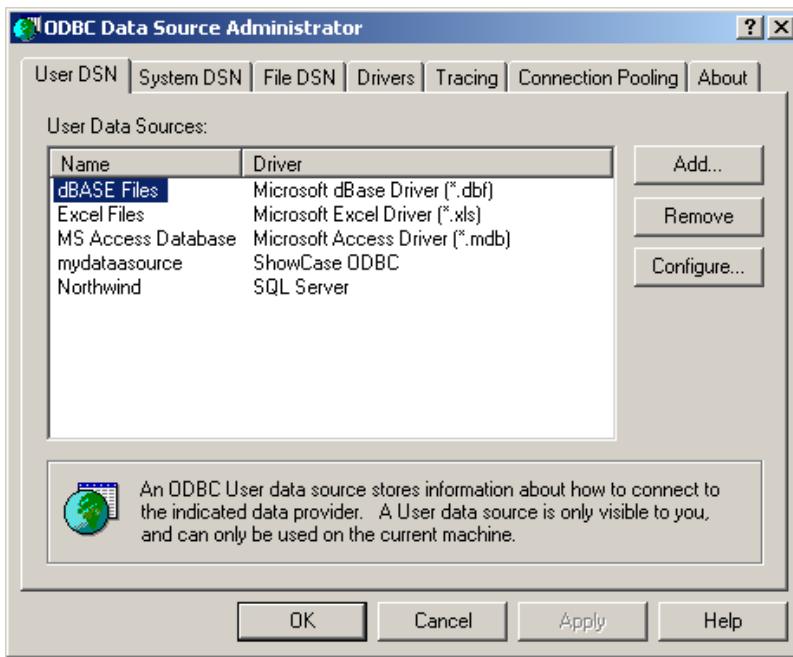
This will take you to the ODBC Microsoft SQL Server Setup screen.

Figure 3.25 ODBC Microsoft SQL Server Setup dialog

Click the **Test Data Source** button to test the connection between the driver and the SQL database.

If the tests connection is successful, you will receive a message to the tests were completed successfully.

Click **OK** to return to the ODBC Microsoft SQL Server Setup dialog
Click **OK** to return to the ODBC Data Source Administrator box

Figure 3.26 ODBC Data Source Administrator dialog

The database has now been declared as a data source and is available to PASW Modeler and other applications via ODBC.

Click **OK** to return to the Control Panel

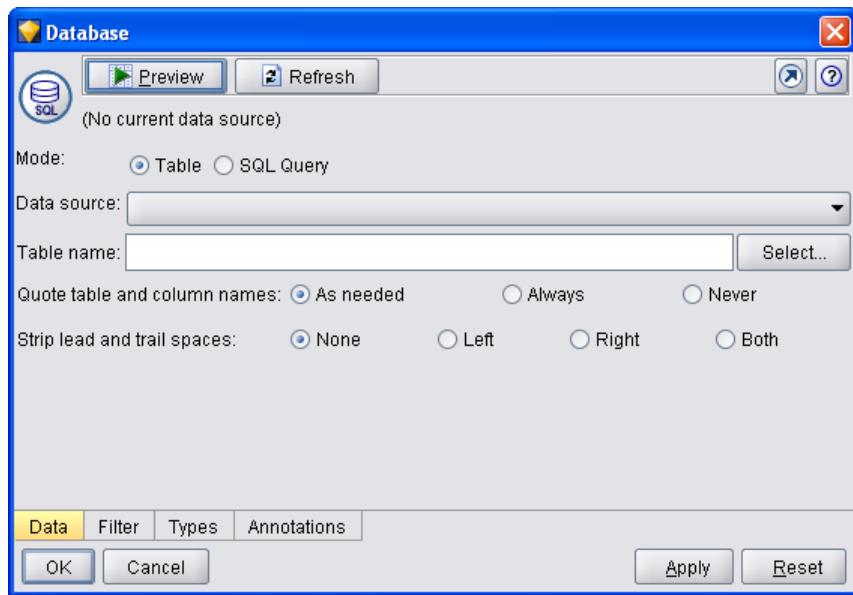
Accessing a Database Using the Database Source Node

In this section we will illustrate the use of the ODBC source node to read data from an SQL Server database from within PASW Modeler:

Clear the stream by choosing **Edit...Clear Stream**

Select the **Database** node from the Sources or Favorites palette and place it on the Stream Canvas

Edit (double-click) the **Database** source node

Figure 3.27 Database Node Dialog

The Mode options (Table or SQL Query) allow you select whether you wish to view the Tables/Views within the database, or to enter/load a SQL Query against the database. In our example we have to read a table from the Holidays database, so Table mode is the choice.

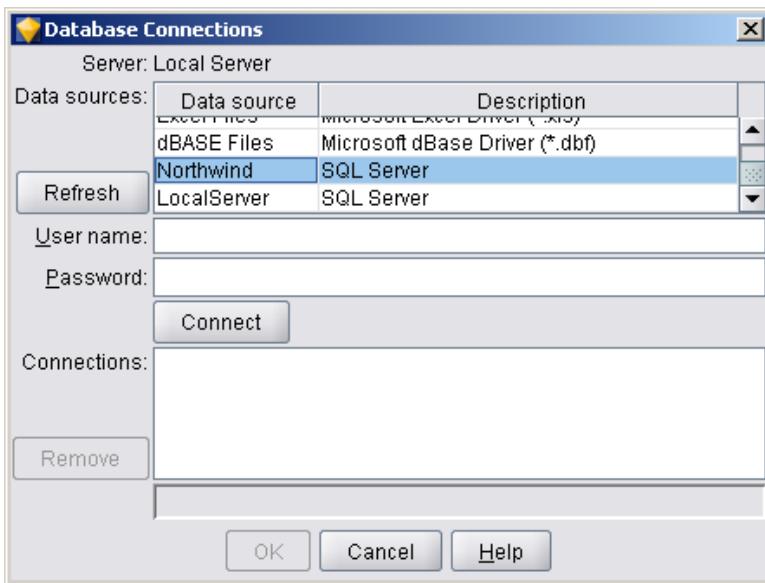
To access the data:

Click the **Data source** drop-down arrow

Since we have not connected to a database yet, we have to add a database connection.

Click **Add New Database Connection**

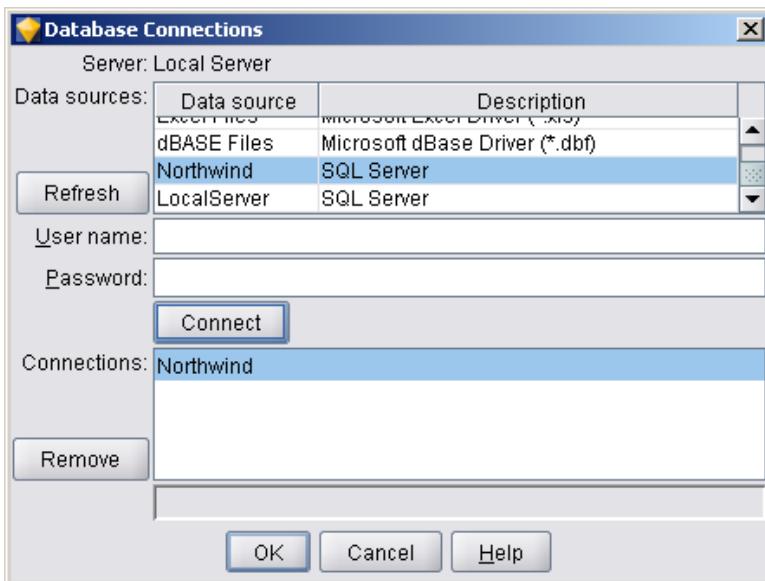
A dialog box will appear allowing you to select the database:

Figure 3.28 Connect to ODBC Data Source Dialog

The database defined as a data source in the previous section can be selected. If required, you can specify a User name and password for the database.

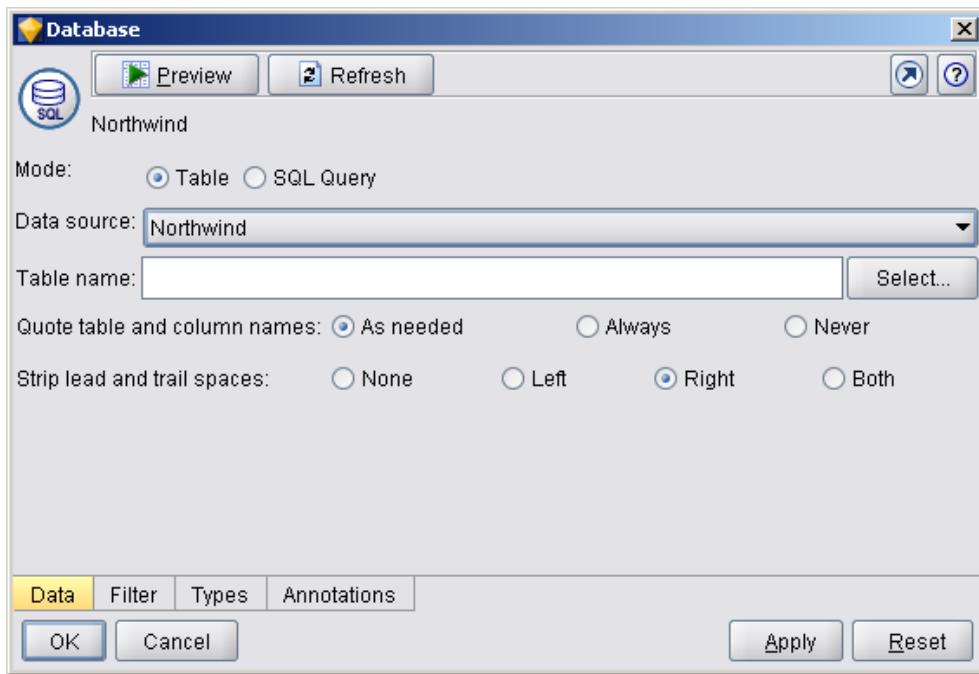
Scroll down the **Data sources:** list and select the **Northwind** data source
Click the **Connect** button

The Northwind data source now appears in the Connections box.

Figure 3.29 Database Connections Dialog (After Connection to Data Source)

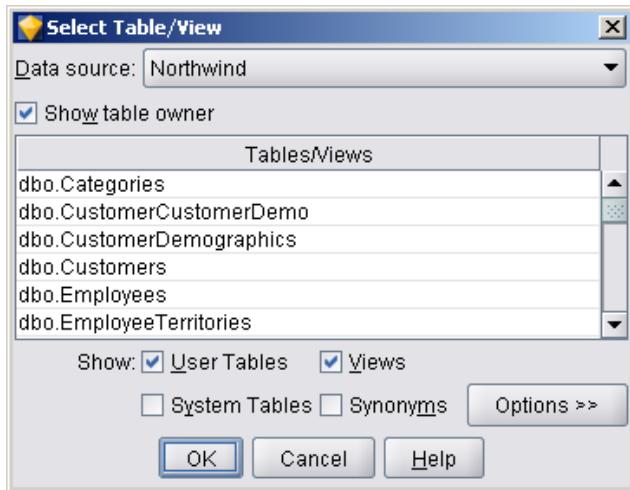
Click **OK**

Returning to the Database dialog, we see that Northwind is selected as the Data source.

Figure 3.30 Data Source Defined in Database Dialog

The next step is to select the database table.

Click on the **Select...** button

Figure 3.31 Select Table/View Dialog

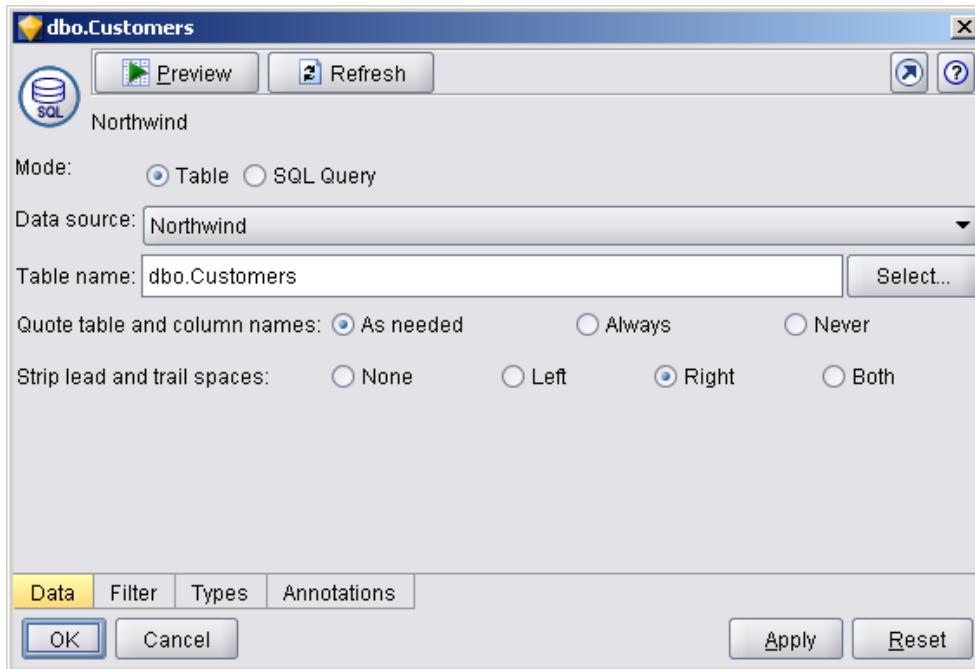
The tables within the selected database appear. When *Show table owner* is checked the owner of each table/view is shown. This will raise an error if the ODBC driver in use does not support the table owners.

Uncheck the **Show table owner** check box

The user-defined tables are shown in the Tables/Views list. You may choose whether or not to see system tables. Note that only one Table/View may be selected at any one time.

Click on **dbo.Customers** in the Tables/Views list
Click **OK** to return to the Database dialog box

Figure 3.32 Completed Database Dialog



In this dialog, options control how spaces in string fields are handled. Strings can be left and/or right trimmed. There is a further specification for quotes in relation to tables and field names. When selected, PASW Modeler will enclose table and column names in quotation marks when queries are sent to the database (if, for example, they contain spaces or punctuation). This setting can be altered to always or never quote table and column names.

If you wish to read in all of the fields from the selected table then the **OK** button can simply be clicked to return to the Stream Canvas. Alternatively you can select which fields you wish to read into PASW Modeler (use Filter tab), and then return to the Stream Canvas. This node, as a Source node, contains a Types tab that can be used to examine and set field measurement level.

Click **OK** to return to the Stream Canvas
Connect the **Database** source node to a **Table** node
Run the stream

Figure 3.33 Data Read from dbo.Customers Table in the SQL Server Database

The screenshot shows a software interface titled "Table (11 fields, 91 records)". The window has a menu bar with "File", "Edit", "Generate", and "Help". Below the menu is a toolbar with icons for file operations. The main area displays a grid of data with 20 rows and 6 columns. The columns are labeled: CustomerID, CompanyName, ContactName, ContactTitle, and Address. The data represents various customers from the SQL Server database. Row 1 contains the header information. Rows 2 through 20 contain specific customer details such as 'Alfreds Futterkiste' at 'Obere Str. 57' and 'Maria Anders' as 'Sales Representative'. The "Table" tab is selected at the bottom left, and an "OK" button is at the bottom right.

	CustomerID	CompanyName	ContactName	ContactTitle	Address
1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57
2	ANATR	Ana Trujillo Emparedados y helados	Aña Trujillo	Owner	Avda. de la Constitución 222
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312
4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.
5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8
6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57
7	BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber
8	BOLID	Bólido Comidas preparadas	Martin Sommer	Owner	C/ Araquil, 67
9	BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers
10	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.
11	BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus
12	CACTU	Cactus Comidas para llevar	Patricia Simpson	Sales Agent	Cerrito 333
13	CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993
14	CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hauptstr. 29
15	COMMI	Comércio Mineiro	Pedro Afonso	Sales Associate	Av. dos Lusíadas, 23
16	CONSH	Consolidated Holdings	Elizabeth Brown	Sales Representative	Berkeley Gardens 12 Brewster
17	DRACD	Drachenblut Delikatessen	Sven Ottlieb	Order Administrator	Walserweg 21
18	DUMON	Du monde entier	Janine Labrune	Owner	67, rue des Cinquante Otages
19	EASTC	Eastern Connection	Ann Devon	Sales Agent	35 King George
20	ERNSH	Ernst Handel	Roland Mendel	Sales Manager	Kirchgasse 6

The data table from the SQL Server database has been read into PASW Modeler.

If you wish to access a number of fields from different tables within a database, just use the following procedure:

- Use a different Database source node for each table within the database
- Link the relevant fields together in a stream containing Append and Merge nodes (examples appear in the *Preparing Data for Data Mining* training course).

3.6 Reading Data from Excel Spreadsheets

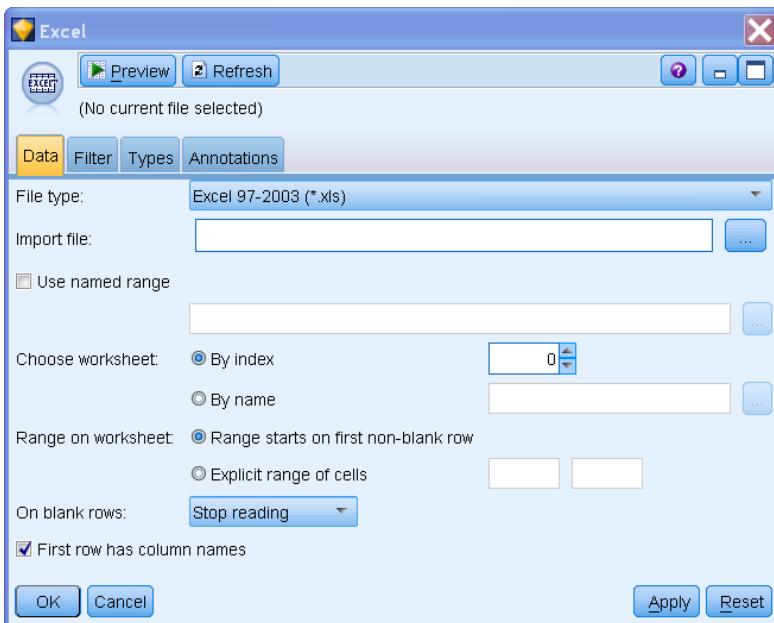
The Excel Node allows you to import data directly from any version of Microsoft Excel. Excel import is supported on Windows platforms only, not under UNIX®.

To read an Excel spreadsheet into PASW Modeler, place the Excel node on the Stream Canvas.

Close the Table window if it is still open
 Click the **Excel** node from the Sources palette
 Click in an open area on the left side of the Stream Canvas

To specify the name of the Excel file to be read:

Double-click on the **Excel** node to edit it

Figure 3.34 Excel Import Node Dialog

The file name and directory can be specified using the file list button [...] (or typed directly).

Select **SmallSample.xls** from C:\Train\ModelerIntro

The *Use Named Range* option allows you to specify a named range of cells as defined in the Excel worksheet. For a list of available ranges, click the [...] button.

The *Worksheet* option lets you specify the worksheet you want to import. Individual worksheets can be referred to either by Name or Index number. Index numbering begins with 0 for the first worksheet, 1 for the second worksheet, and so forth.

The *Data range* option lets you import an explicit range of cells in situations when you don't want the whole spreadsheet. With this option, you can either import the data beginning with the first non-blank row or specify an explicit range of cells. To import an explicit range, check the *Explicit range* option and specify the location for the upper left corner of the range in the first box and for the lower right corner in the second box. Be sure to type the locations in uppercase or they will not be recognized, e.g., A3 and G178. All the rows in that range will be returned including the blank rows.

The *First row contains field names* is used to indicate that the first row contains the names you want to use as field names. If not selected, field names will be generated automatically.

Click **OK** to close the Excel dialog box

Add a **Table** node and connect the **Excel** node to the **Table** node

Run the stream

Figure 3.35 Data Read from SmallSample.xls

ID	AGE	INCOME	GENDER	MARITAL	NUMKIDS	NUMCARDS	HOMPAID	MORTGAGE	ST
1	100319.000	31.000	59193.000	Female	married	1.000	2.000	monthly	y
2	100796.000	45.000	58381.000	Male	married	1.000	1.000	monthly	y
3	100730.000	43.000	57388.000	Female	married	0.000	1.000	monthly	y
4	100670.000	41.000	56470.000	Male	married	0.000	2.000	monthly	y
5	100347.000	\$null\$	55554.000	Female	married	0.000	1.000	monthly	y
6	100348.000	32.000	54792.000	Male	married	1.000	1.000	monthly	y
7	100770.000	44.000	53983.000	Female	married	1.000	2.000	monthly	y
8	100753.000	44.000	53550.000	Male	married	1.000	1.000	monthly	y
9	100350.000	32.000	52973.000	Male	married	1.000	1.000	monthly	y
10	100599.000	39.000	52495.000	Female	married	1.000	2.000	monthly	y
11	100765.000	44.000	51498.000	Male	married	0.000	1.000	monthly	y
12	100664.000	33.000	50831.000	Female	married	0.000	2.000	monthly	y
13	100571.000	38.000	50076.000	Male	married	1.000	1.000	monthly	y
14	100622.000	35.000	49600.000	Male	married	1.000	2.000	monthly	y
15	100423.000	34.000	49007.000	Male	married	1.000	1.000	monthly	y
16	100527.000	37.000	48061.000	Female	married	1.000	2.000	monthly	y
17	100595.000	39.000	47161.000	Male	married	1.000	2.000	monthly	y
18	100565.000	38.000	46523.000	Male	married	0.000	1.000	monthly	y
19	100470.000	36.000	45949.000	Female	married	1.000	1.000	monthly	y
20	100282.000	30.000	45715.000	Male	married	0.000	2.000	monthly	y
21	100679.000	42.000	45564.000	Female	married	1.000	1.000	monthly	y

3.7 Data from PASW Data Collection Products

The Data Collection node is used to import survey data based on the Data Collection Data Model used by many software applications from SPSS, an IBM Company. Some examples of Data Collection products are PASW® Data Collection Interviewer Web, PASW® Reports for Surveys, and PASW® Data Collection Base. This node allows you to read and mine survey data in Data Collection files directly in PASW Modeler rather than having to export the data first and then read it into PASW Modeler in a subsequent step. This node requires the Data Collection Data Model 3.0 or higher in order to run. The Data Collection Export node saves data in the format used by PASW Data Collection market research software.

3.8 SAS Software Compatible Data

The SAS File node allows you to bring SAS software compatible data into your data mining session. You can import four types of files:

- SAS for Windows/OS2 (.sd2)
- SAS for UNIX (.ssd)
- SAS Transport File (*.pt)
- SAS version 7/8/9 (.sas7bdat)

When the data are imported, all variables are kept and no field types are changed. All cases are selected. Similar to the SAS File node, the SAS Export node allows you to write data in SAS software compatible format to be read into SAS applications. You can export in three SAS software file formats: SAS for Windows/OS2, SAS for UNIX, or SAS Version 7/8/9.

3.9 XML Data

The XML source node imports data in XML format into the stream. You can import a single file, or all files in a directory. You can optionally specify a schema file from which to read the XML structure. The XML Export node enables you to output data in XML format, using UTF-8 encoding.

3.10 Defining Field Measurement Level

After specifying your data source, the next step is to define the measurement level for each of the fields within the data. The measurement level for each field must be set before the fields can be used in the Modeling, and some other, nodes.

Measurement level can be set in most source nodes (click on the Types tab) at the same time you define your data, or in the Type node (found in the Field Ops palette) if you need to define a measurement level for the field later in your stream (perhaps because a new field was created). In some earlier examples we have seen the Types tab in data source nodes. We now turn to our postponed discussion of measurement level definitions. We will define the measurement level in the Types tab in the source node, but we could also use a Type node to do this.

As a data source we will open *SmallSampleComma.txt*.

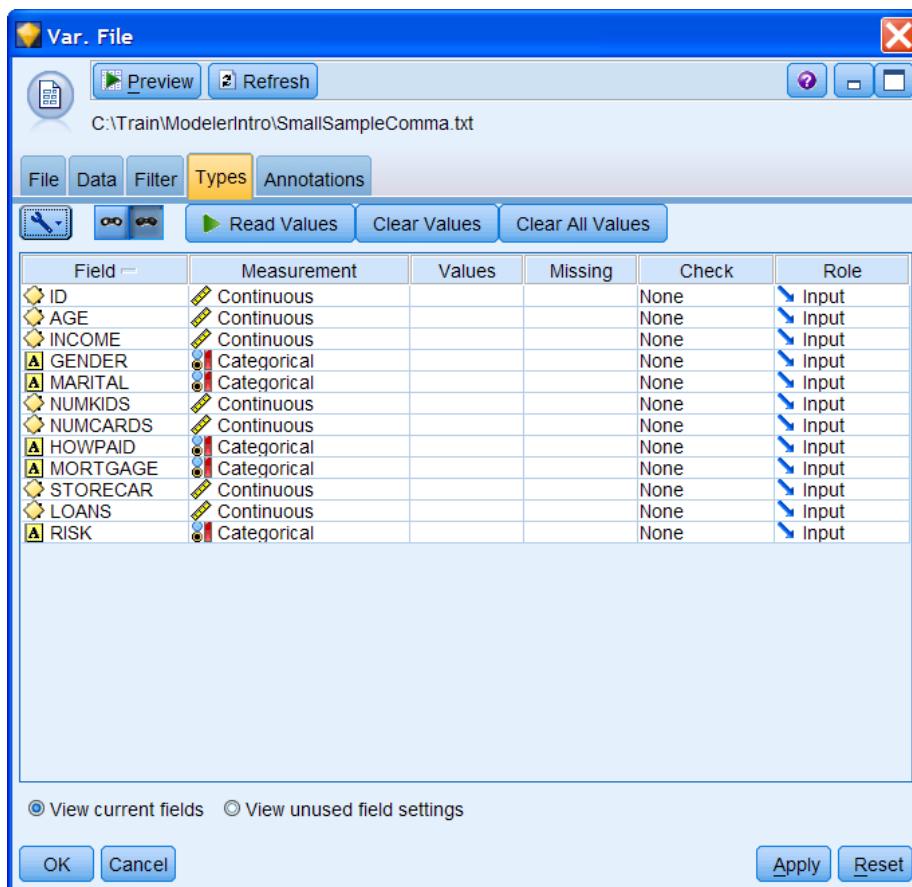
Clear the stream by choosing **Edit...Clear Stream**

Place a **Var. File** node on the Stream Canvas

Edit the node and specify **SmallSampleComma.txt** in the **c:\Train\ModelerIntro** directory
as data file

Click **Types** tab

Figure 3.40 Types Tab of Var. File Node



The Types tab in a data source node or the Type node controls the properties of each field: measurement level, data values, role, and missing value definitions. This node also has a Check

facility that, when turned on, examines fields to ensure that they conform to specified settings, such as checking whether all the values in a field are within a specified range. This option can be useful for cleaning up data sets in a single operation.

In this section we concentrate on the measurement level and role definitions. Other specifications (missing values) will be discussed in later lessons.

Measurement Level Definition

The measurement column in the Types tab of source nodes (and the Type node) describes the measurement level of the field, which determines how PASW Modeler will use the field. PASW Modeler distinguishes among:

- **Continuous.** Used to describe numeric values such as a range of 0-100 or 0.75-1.25. A continuous value may be an integer, real number, or date/time.
- **Categorical.** Used for string values when an exact number of distinct values is unknown.
- **Flag.** Used for data with two distinct values such as Yes/No or 1, 2.
- **Nominal.** Used to describe data with multiple distinct values, each treated as a member of a set, such as married, single, divorced, etc.
- **Ordinal.** Used to describe data with multiple distinct values that have an inherent order, such as *1 low*, *2 medium*, and *3 high*. Notice, the order is defined by the natural sort order of the data elements. For example, *1, 3, 5* is the default sort order for a set of integers, while *high, medium, low* (ascending alphabetically) is the order for a string values. So, make sure the categories of the field will be ordered correctly when the field is defined as ordinal.
- **Typeless.** Used for data that does not conform to any of the above measurement levels or for a categorical field with too many values. It is useful for cases in which the measurement level would otherwise be categorical with many values (such as an account number). When you select Typeless for a field's measurement level, the field role is automatically set to None (meaning the field cannot be used in modeling). The default maximum size for sets is 250 unique values. This number can be adjusted or disabled in the Stream Properties dialog.

At this stage (see Figure 3.41), the fields in *SmallSampleComma.txt* are in a partially instantiated state. Instantiation refers to the process of reading or specifying information such as measurement level and values for a field. Fields with totally unknown measurement level are considered uninstantiated. Fields are referred to as partially instantiated if the program has some information about how they are stored (string or numeric), but the details are incomplete. For example, the Categorical measurement level is temporarily assigned to a string field until it can be determined if it is either a Flag, Nominal or Ordinal measurement level. The Continuous measurement level is given to all numeric fields, whether they are fully instantiated or not. When all the details about a field are known, including the measurement level and values, it is considered fully instantiated and Flag, Nominal, Ordinal or Continuous is displayed in the Measurement column (although normally you will have to change the measurement level to Ordinal yourself after the data are read).

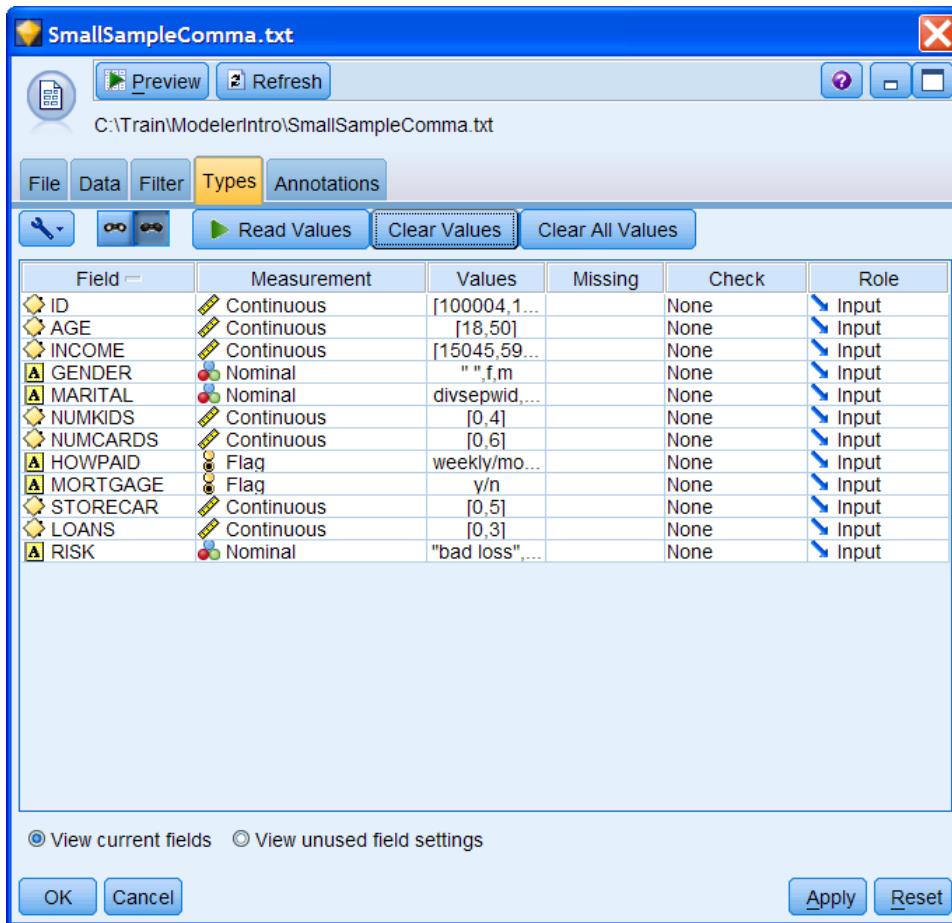
During the execution of a data stream instantiation occurs when the field Values settings in the Types tab are set to Read or Read+ (meaning that values should be read, or current values retained and new values added when the data are read). Once all of the data have passed through the data source or Type node, all fields become fully instantiated.

In reading the data values through the source node, PASW Modeler identifies the measurement level of each field (when the field's Values property is set to Read or Read+). To check the measurement level, edit the source node (or Type node) after data have passed through it. We can force data

through the source node by placing and executing a node downstream of it; alternatively we can click the Read Values button, which reads the data into the source node or Type node (if Read Values is clicked from within a Type node).

Click the **Read Values** button, and then click **OK** to the subsequent dialog box

Figure 3.41 Types Tab after Values are Read (Fully Instantiated)



Fields *ID*, *AGE*, and *INCOME* are continuous (with the lower and upper bounds in the Values column). *HOWPAID* (with values weekly/monthly) and *MORTGAGE* (with values y/n) are flags. *MARITAL* and *RISK* are nominal (*RISK* could be changed to ordinal since its values can be ranked in terms of potential monetary return). Notice that *GENDER* is nominal, due to the fact that not only f and m appear as values, but also a space " ".

If execution is interrupted, the data will remain partially instantiated. Once the measurement levels have been instantiated, the values of a field in the Values column of the Types tab are static at that point in the stream. This means that any upstream data changes will not affect the stored Values of a particular field, even if you re-run the stream. To change or update the Values based on new data or added manipulations, you need to edit them in the Types tab (or re-instantiate the field, by setting its Values column entry to Read or Read+ and passing data through the node).

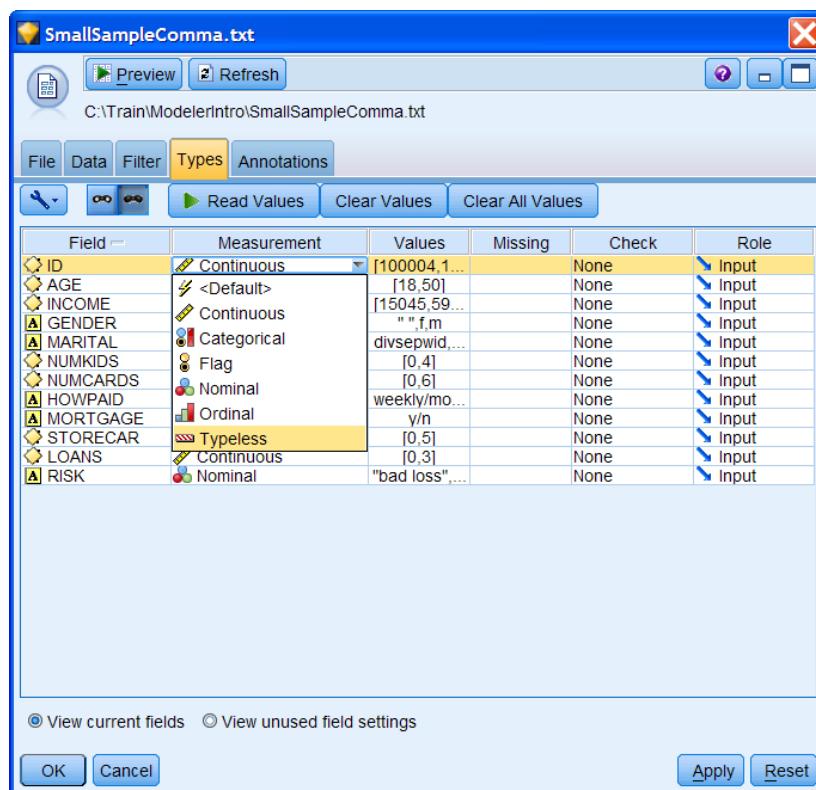
Numeric fields that represent categories must be forced to the correct field measurement level by specifying Categorical. They will then be assigned flag or nominal measurement level when fully instantiated.

The easiest way to define the measurement level of each field is to initially allow PASW Modeler to autotype by passing the data through the source node (or Type node), and then manually editing any incorrect measurement level.

As an example of changing the measurement level for a field, consider *ID*. This field contains a unique reference number and is better defined as Typeless.

Click in the **Measurement** column for the field **ID**
Choose **Typeless** from the list

Figure 3.42 Context Menu for Entries in Measurement Column



After selecting Typeless, *ID*'s measurement level will change to Typeless and its role will change to None (not shown). We discuss role next.

Click **OK**

3.11 Field Role

The role of a field is relevant only to modeling nodes. The available roles are:

Input	The field will be used as an input or predictor to a modeling technique. (i.e., a value on which predictions will be based).
Target	The field will be the target for a modeling technique. (i.e. the field to be predicted).
Both	Role suitable for the Apriori, CARMA, and Sequence modeling nodes. Allows the field to be both an input and a target in an association rule. All other modeling techniques will ignore the field.
None	The field will not be used in modeling.
Partition	Indicates a field used to partition the data into separate samples for training, testing, and (optional) validation purposes. We will discuss this option in a later lesson.
Split	Only available for categorical (flag, nominal, ordinal) fields. Specifies that a model is to be built for each possible value of the split field.
Frequency	Only available for numeric fields. Setting this role enables the field value to be used as a frequency weighting factor for the record.
Record ID	Only relevant for the Linear node, where the specified field will be used as the unique record identifier.

Setting the role for a field is done in the same way as setting the measurement level: click on the Role value for a field and choose the appropriate role from the drop-down list. Multiple fields can be selected and properties like role or measurement level changed from the context menu (right-click on any of the selected fields).

Note

Setting the role of fields may be performed later in the project if you are in the initial stages of data mining or are not planning on using any of the Modeling techniques available.

3.12 Saving a PASW Modeler Stream

To save our PASW Modeler stream for later work:

Click **File...Save Stream As**.
Navigate to the **c:\Train\ModelerIntro** directory (if necessary)
Type **SmallCommaDef** in the File name text box (not shown)
Click the **Save** button

The File menu also allows you to save (and Open) a State file (which contains the stream and any models stored in the Models palette) and a Project file (which can contain streams, graphs, reports, and generated models, thus organizing elements related to an analysis project). Also, you can add the saved stream to the current project by clicking the *Add file to project* check box.

Summary

In this lesson you have been given an introduction on how to read data into PASW Modeler, define the measurement level and role of the fields, and view the data file.

3.13 Appendix A: Reading Data from Fixed-field Text Files

Data in fixed column format can be read into PASW Modeler with the Fixed File node in the Sources palette. We see an example of such a file below (shown in Notepad).

Figure 3.43 Fixed-field Text File

SmallSampleFixed.txt - Notepad						
File	Edit	Format	View	Help		
100319	31	59193	f	married	1	2 m y 1 1 good risk
100796	45	58381	m	married	1	1 m y 1 0 good risk
100730	43	57388	f	married	0	1 m y 1 0 bad loss
100670	41	56470	m	married	0	2 m y 1 0 bad loss
100347		55554	f	married	0	1 m y 1 0 good risk
100348	32	54792	m	married	1	1 m y 2 0 good risk
100770	44	53983	f	married	1	2 m y 2 0 bad profit
100753	44	53550	m	married	1	1 m y 1 1 bad loss
100350	32	52973	m	married	1	1 m y 1 0 bad profit
100599	39	52495	f	married	1	2 m y 1 1 good risk
100765	44	51498	m	married	0	1 m y 2 1 bad loss
100664	33	50631	f	married	0	2 m y 1 0 good risk
100571	38	50076	m	married	1	1 m y 1 1 bad profit
100622	35	49600	m	married	1	2 m y 2 1 good risk
100423	34	49007	m	married	1	1 m y 1 0 bad profit
100527	37	48061	f	married	1	2 m y 1 0 good risk
100595	39	47161	m	married	1	2 m y 1 1 good risk
100565	38	46823	m	married	0	1 m y 1 1 good risk
100470	36	45949	f	married	1	1 m y 2 0 bad profit
100282	30	45715	m	married	0	2 m y 1 1 good risk
100679	42	45584	f	married	1	1 m y 1 0 good risk
100722	43	45390	f	married	0	2 m y 2 0 good risk
100449	35	45238	f	married	0	2 m y 1 1 good risk
100562	38	45103	f	married	0	1 m y 2 1 good risk

Each field is located in the same column positions on every record in the data file. When instructing PASW Modeler how to read such a file, you must know the column position(s) that each variable occupies. Typically the program or individual creating the data file can supply this information. In our example, we have the following information available.

Table 3.1 Information about Field Start Position and Length

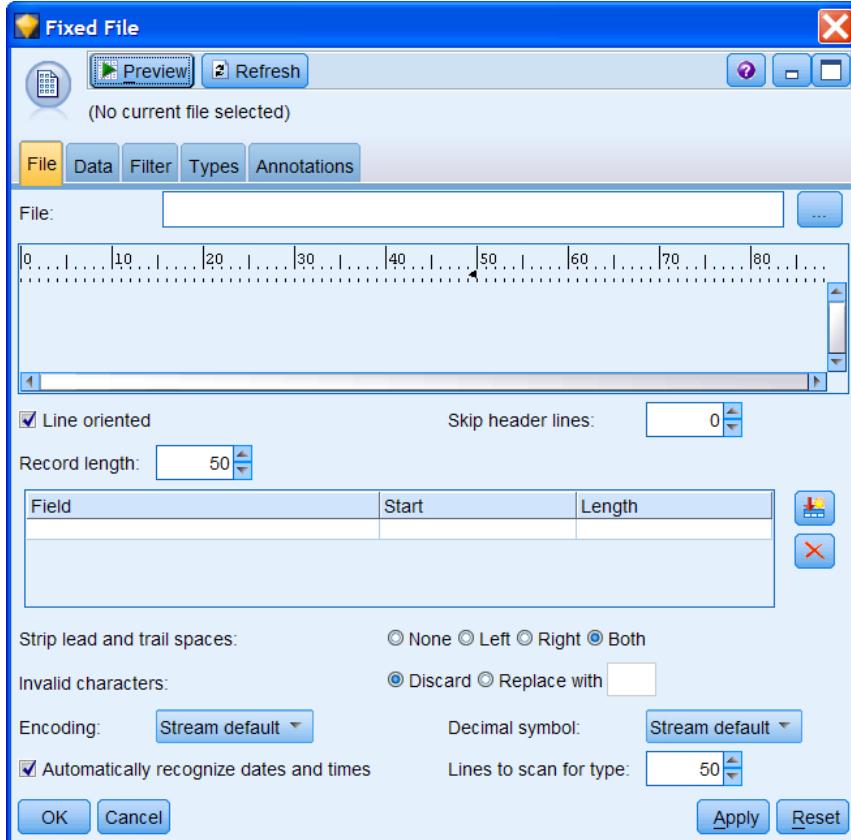
Field	Start Position	Length
ID	1	6
AGE	8	2
INCOME	11	5
GENDER	17	1
MARITAL	19	7
NUMKIDS	30	1
NUMCARDS	32	1
HOWPAID	34	1
MORTGAGE	36	1
STORECAR	38	1
LOANS	40	1
RISK	44	10

Data in fixed column format can be read into PASW Modeler using the Fixed File node in the Sources palette.

Start a new stream by choosing **File...New Stream**

Double-click the **Fixed File** node in the **Sources** palette, which places it directly on the Stream Canvas
Double-click on the **Fixed File** node in the Stream Canvas to edit it

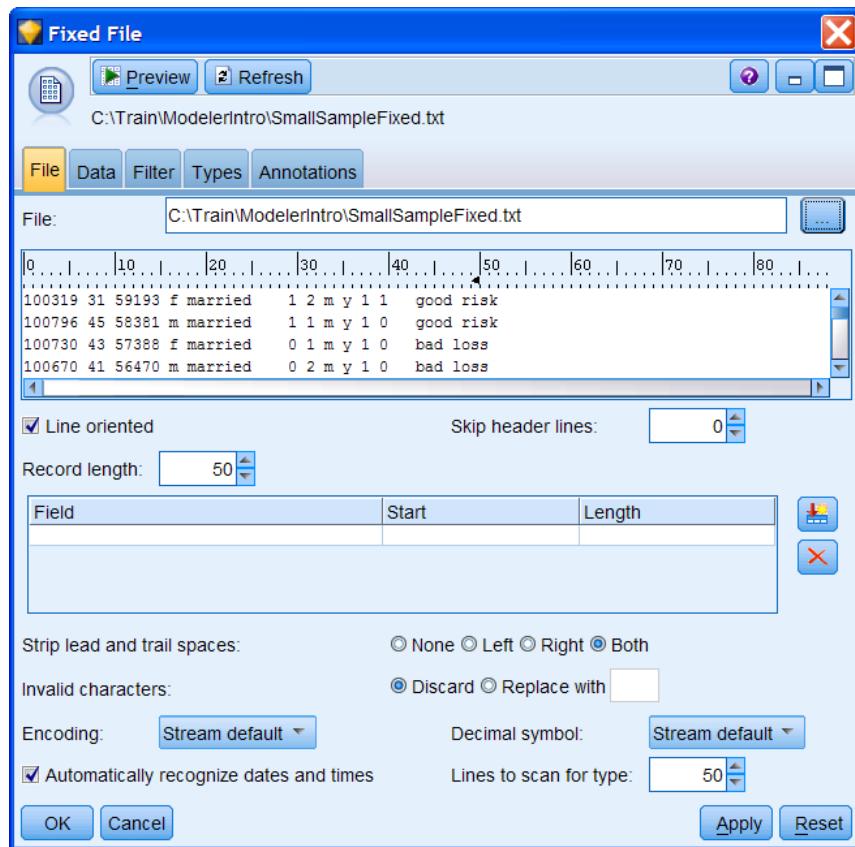
Figure 3.44 Fixed File Source Dialog



This dialog has much in common with the Var. File dialog, such as the handling of leading and trailing spaces in string fields, how many lines to scan for measurement level, skipping lines, a Data tab (to manage data storage), a Filter tab (to include/exclude fields or to rename fields) and a Types tab (to manage measurement level for a field).

To read the data, we first have to specify the data file:

Click the file list button 
Select **SmallSampleFixed.txt** in this directory, and then click **Open**

Figure 3.45 Fixed File Node Dialog: Data Preview

There are two ways to define fields:

- Interactively: You specify fields using the data preview above. The ruler at the top of the preview window helps you to measure the length of variables and specify the breakpoint between them. You can specify breakpoint lines by clicking in the ruler area above the fields. Each breakpoint line automatically adds a new field to the field table below. Start positions indicated by the arrows are automatically added to the Start column in the table below. Breakpoints can be moved by dragging and can be discarded by dragging them outside the data preview region.
- Manually: You specify fields by adding empty field rows to the table below. Double-click in a cell or click the New Field button to add new fields. Then, in the empty field row, enter a field name, a start position and a length. These options will automatically add breakpoint lines (arrows) to the data preview canvas that can be easily adjusted.

As we have information about starting positions of the fields and field lengths readily available, we choose the second alternative.

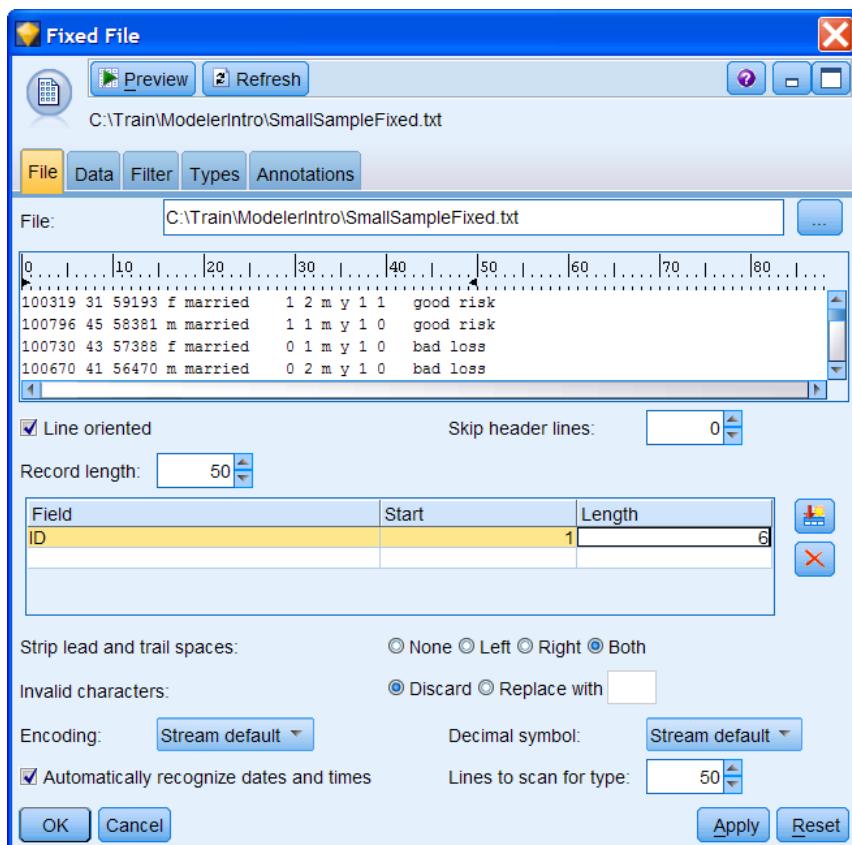
Double-click in the cell in the **Field** column

Specify **ID** as fieldname

Press the tab key to move on to the **Start** column (or double-click in the cell in the **Start** column). Note, that the start position is already specified as **1** by default, so we can move to the next specification

Press the tab key to move on to the **Length** column. Replace the default value **1** with **6**

Figure 3.46 Fixed File Dialog: Specifying Field Information



To define the next field (*AGE*, starting at position 8, length 2):

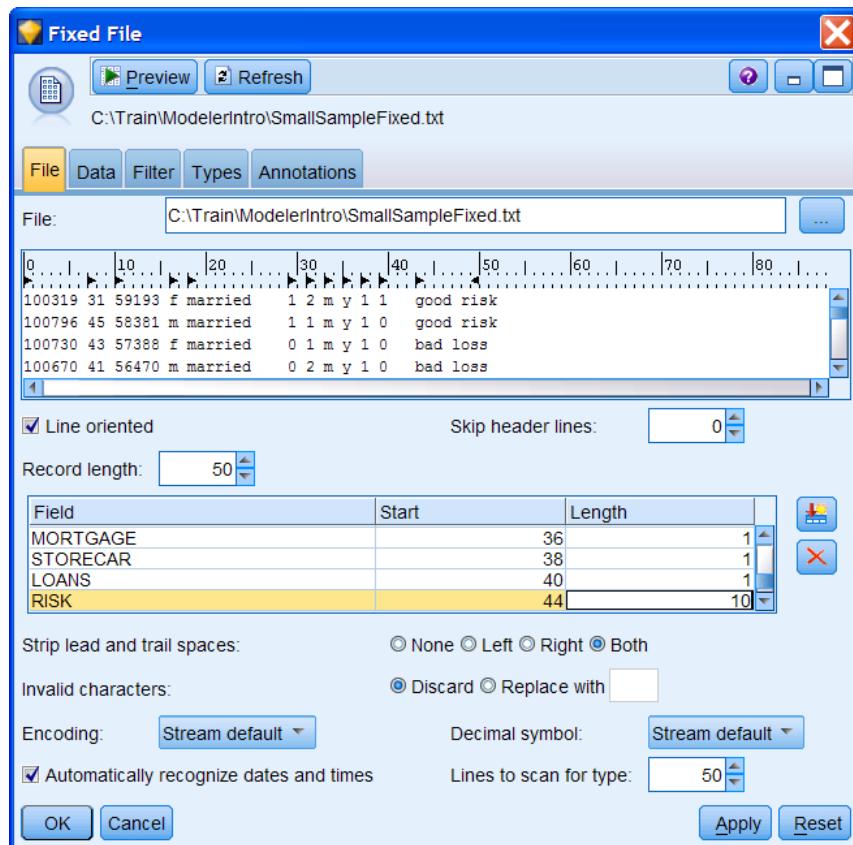
Double-click in cell in the **Field** column below **ID**

Specify **AGE** as fieldname

Move on to the **Start** column and type **8**

Move on to the **Length** column and type **2**

The rest of the fields are defined in the same way. The final result is shown below.

Figure 3.47 Fixed File: All Fields Defined

Any fields that are not defined are skipped when the file is read.

Although the definitions look correct, there is still a detail remaining. Notice that at position 50 in the preview pane, there is an end-of-line indicator (`\n`). This is a result of the default record length of 50. Unless we make a change, the last characters of *RISK* won't be read. To correct this:

Set the record length to, say, 60, either by moving the end-of-line character `\n` or by typing **60** in the **Record length:** text box

Now that our definitions are complete, we can move on and check to see if the data are read correctly.

Click the **Preview** button

Figure 3.48 Preview Table Showing Data from Fixed Text File

	ID	AGE	INCOME	GENDER	MARITAL	NUMKIDS	NUMCARDS	HOWPAID	MORTGAGE	STORECAR	LOANS	RISK
1	100319	31	59193	f	married	1	2	m	y	1	1	good risk
2	100796	45	58381	m	married	1	1	m	y	1	0	good risk
3	100730	43	57388	f	married	0	1	m	y	1	0	bad loss
4	100670	41	56470	m	married	0	2	m	y	1	0	bad loss
5	100347	\$null\$	55554	f	married	0	1	m	y	1	0	good risk
6	100348	32	54792	m	married	1	1	m	y	2	0	good risk
7	100770	44	53983	f	married	1	2	m	y	2	0	bad profit
8	100753	44	53550	m	married	1	1	m	y	1	1	bad loss
9	100350	32	52973	m	married	1	1	m	y	1	0	bad profit
10	100599	39	52495	f	married	1	2	m	y	1	1	good risk

Again, note the \$null\$ value for *AGE* in the 5th record. Looking back at the file *SmallSampleFixed.txt* we see that this person had no value for *AGE*, so a missing value, represented by the value \$null\$, was assigned for this case.

3.14 Appendix B: Working with Dates

Data mining will often include working with fields that represent dates. This can range from simply sorting data into a chronological order, to calculating elapsed time, to performing some form of complex time series analysis.

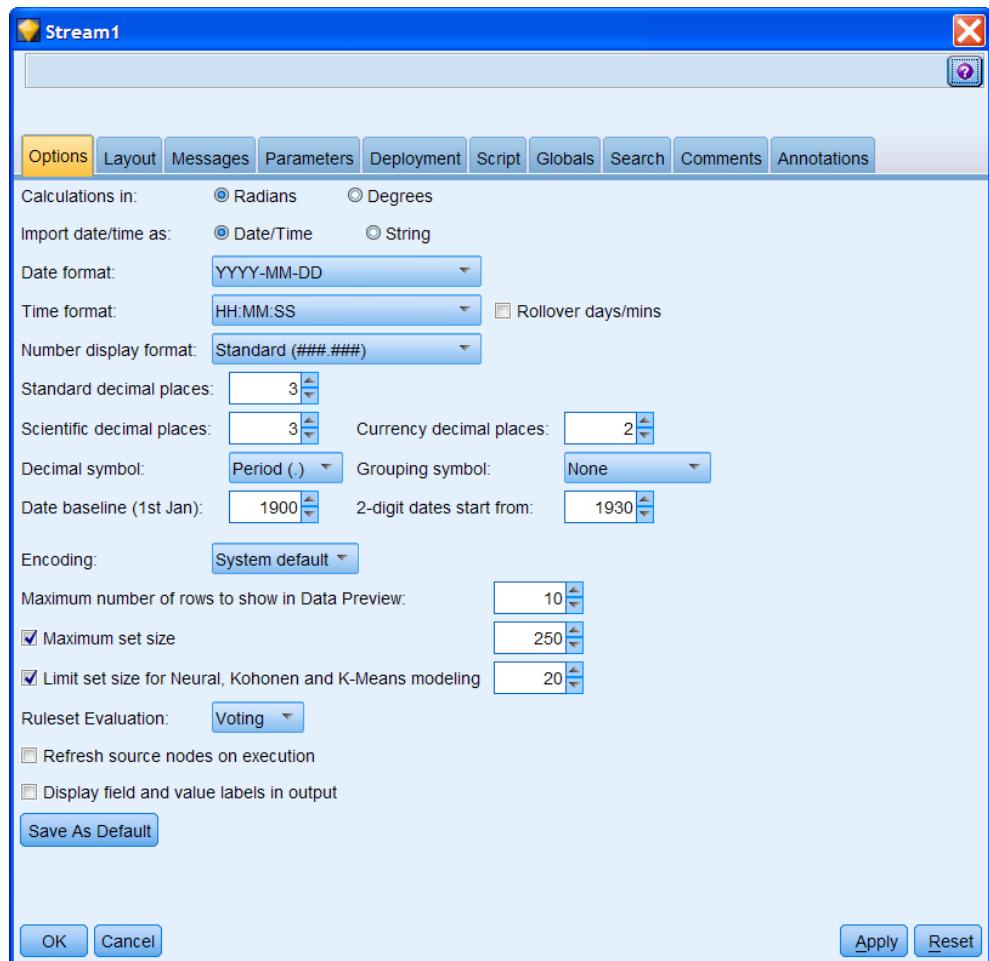
It is not unusual for dates to be stored in a variety of formats within a data file, especially if separate sources have been merged. In this appendix we discuss how to define date formats in PASW Modeler and review the different formats available.

3.15 Declaring Date Formats in PASW Modeler

PASW Modeler can read date fields that are stored either as true dates (date storage), or with the values in string format. The storage type is determined at the source when reading the data into PASW Modeler. Since formats may vary for different databases, there are a number of different representations available.

The formats of date and time fields are specified within the Options tab of the Stream Properties dialog, reached by clicking Tools...Stream Properties...Options. These settings are specific to each stream file and are saved with the relevant stream (or state) file. One limitation of this is that if different date/time formats exist within one data file, the user must select only *one* format for the entire stream and manipulate the other formats into the chosen one. There are a number of PASW Modeler expressions that can be used for such manipulation.

If the Stream Canvas is not empty, click **File...New Stream**
Click **Tools...Stream Properties...Options**

Figure 3.49 Stream Options Dialog: Options Tab

A number of the options refer to time and date settings.

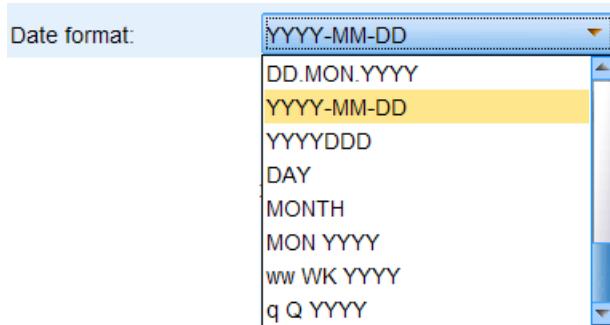
Date baseline (1st Jan) is used as a reference value when using many of the CLEM date functions. (CLEM --- Control Language for Expression Manipulation --- is a powerful language for data manipulation.) For example, the expression `date_in_years(datevar)` returns the time in years from the baseline date (January 1 of the listed year) to the date represented by the string `datevar`.

With *2-digit dates start from* you specify the cutoff year to add century digits for years denoted with only two digits. For example, specifying 1930 as the cutoff year will roll over 05/11/04 to the year 2004. The same setting will use the 20th century (1900s) for dates after 30, such as 05/11/73.

Rollover days/mins check box is appropriate when calculating the difference between two dates/times. If it is checked and PASW Modeler is asked to return the difference between two times, for example 23:45 and 00:23, it will assume that 23:45 is occurring on the previous day and will return the difference of 38 minutes. Alternatively, if it is left unchecked and calculates the same time difference it will return the value of -1392 minutes.

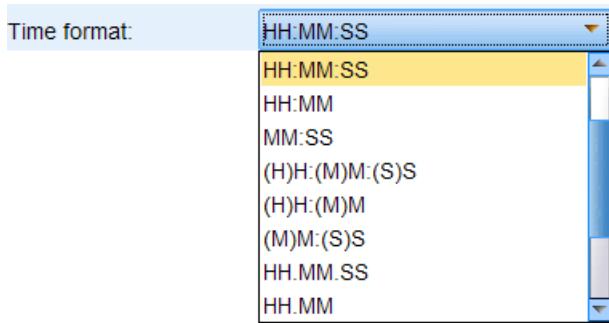
Date format contains a number of different formats that can be used when strings are interpreted as dates by CLEM date functions.

Click the **Date format** drop-down list box

Figure 3.50 PASW Modeler Date Formats

When you wish to interpret string values as times using CLEM time expressions, set *Time format* to the format that matches your data.

Click the **Time format:** drop-down list box

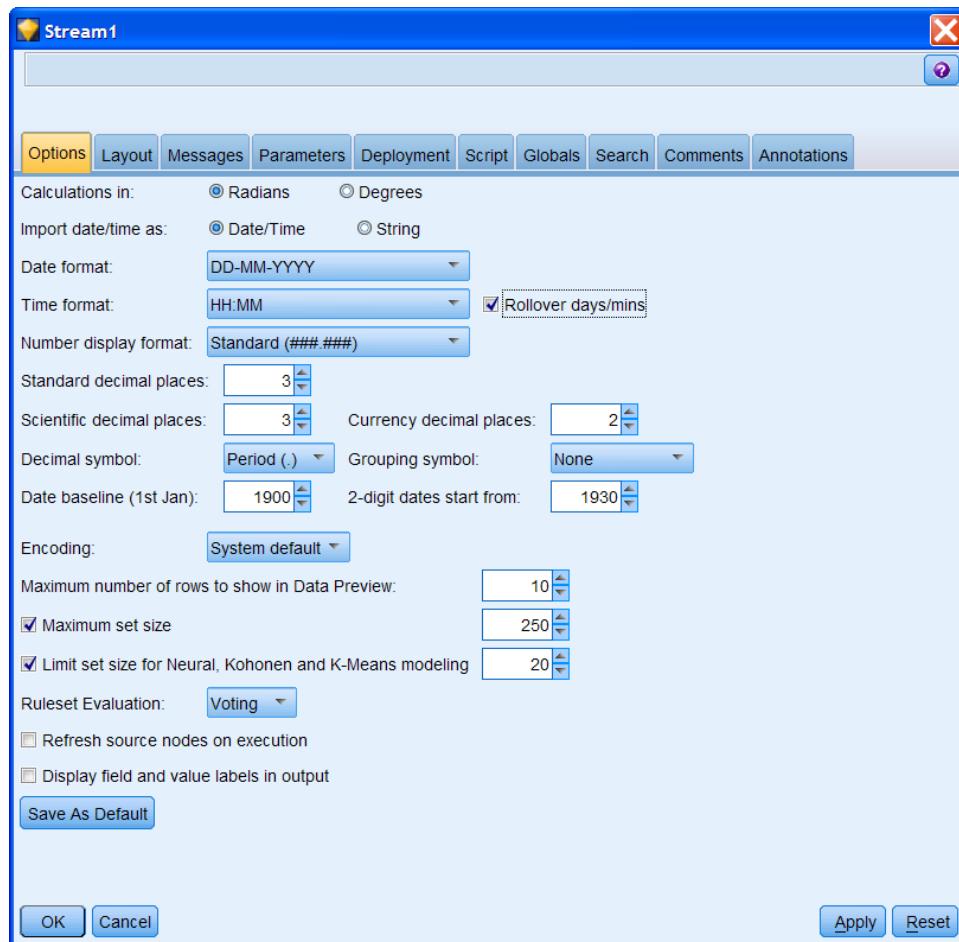
Figure 3.51 PASW Modeler Time Formats

In this example we will change the date and time format. In the data file to be used, *fulldata.txt*, dates are given in the format DD/MM/YYYY and times are given as HH:MM. We set the formats accordingly.

Set the Date format: to **DD-MM-YYYY**

Set the Time format: to **HH:MM**

Click the **Rollover days/mins** check box

Figure 3.52 Setting Date and Time Formats

Click **OK** to return to the Stream Canvas

If the dates being read are in string format, PASW Modeler will be able to interpret them as dates or times in PASW Modeler expressions if the *Import date/time as:* option is set to String, and if they appear in the selected date or time format. Alternatively, if times or dates are read in with date, time or datetime storage, then the *visual display* of the field will be set according to these options.

Reading Data Which Includes Dates

We'll begin by reading data from the file *fulldata.txt*.

Select a **Var. File** node from the Sources Palette and place it on the left-hand side of the canvas.

Edit the **Var. File** node

Browse and select the file **c:\Train\Modeler\Intro\fulldata.txt**

Ensure that the **Read field names from file** is checked

PASW Modeler is now able to access the data stored in this file; however, let's look at how PASW Modeler has set the Storage format for each of the fields within the file.

Click the **Types** tab within the **Var. File** source node

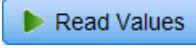
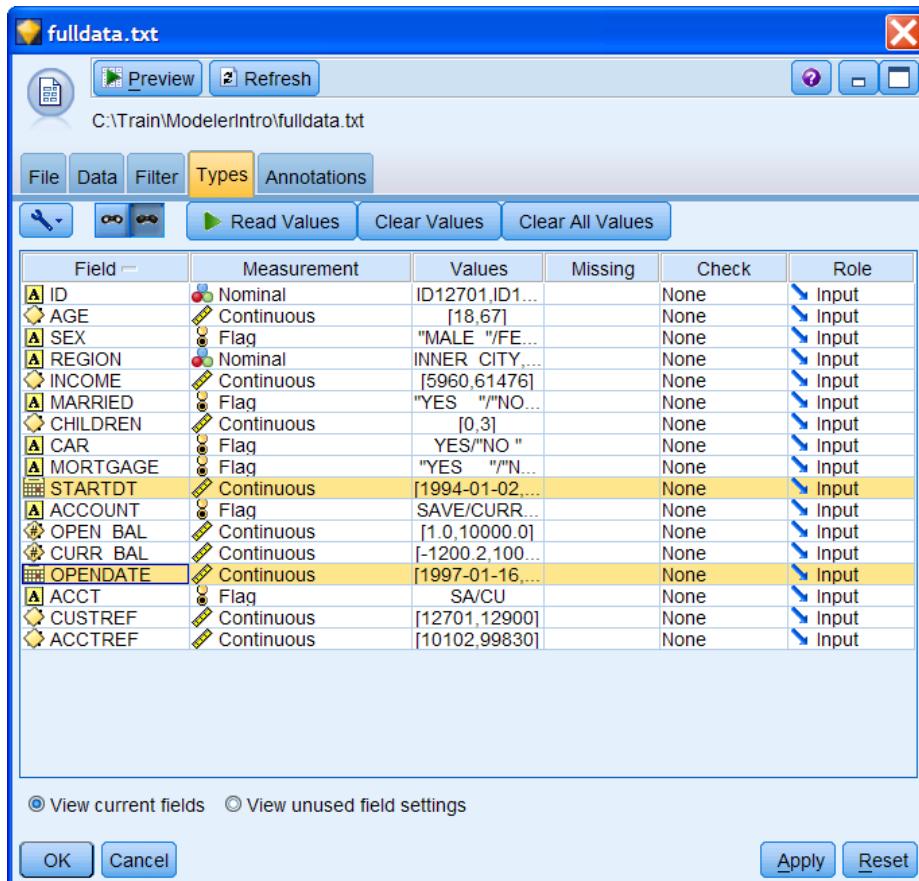
Click the  option within the **Types** tab
 Click **OK** to the Read Values warning message box

Figure 3.53 Initial Storage and Settings for Fields in FullData.txt



Notice that the two fields that should be date fields, *OPENDATE* and *STARTDT*, have been given measurement level continuous, with their intervals displayed within square brackets mark.

Click the **Preview** button

Figure 3.54 Table Showing the Date Fields

Preview from fulldata.txt Node (17 fields, 10 records)

	MORTGAGE	STARTDT	ACCOUNT	OPEN_BAL	CURR_BAL	OPENDATE	ACCT	C
1	YES	05-01-1995	SAVE	1000.000	1005.320	11-02-1997	SA	1
2	YES	13-12-1994	SAVE	100.000	144.510	20-05-1997	SA	1
3	NO	18-02-1994	SAVE	300.000	321.200	20-07-1997	SA	1
4	NO	18-02-1994	CURRENT	150.000	-204.510	23-05-1997	CU	1
5	YES	09-09-1994	SAVE	2000.000	2022.020	07-03-1997	SA	1
6	NO	05-03-1995	SAVE	190.000	287.800	06-09-1997	SA	1
7	NO	05-03-1995	CURRENT	2742.000	2762.990	14-05-1997	CU	1
8	NO	05-03-1995	CURRENT	150.000	191.090	12-06-1997	CU	1
9	YES	21-02-1994	SAVE	300.000	353.690	31-01-1997	SA	1
10	YES	21-02-1994	CURRENT	1412.000	1490.110	04-02-1997	CU	1

Notice that both date fields now follow the date format that was selected within the Stream Properties dialog box. PASW Modeler understands that both of these fields are Date fields and calculations can now be performed on these fields using the functions specifically designed for Date fields.

Data Obtained from Non-Text Files

If data are accessed from a source other than a text file, it is not possible to override the storage within the source node.

If data come from a Statistics file or via ODBC, the original host application determines the storage format of the data, not PASW Modeler. If the date field(s) within the database, or Statistics, has String storage then PASW Modeler will keep the existing format when importing the data.

While it is not possible to override the storage within the source node under these situations, it is possible to convert the storage type downstream of the source node. Using a Derive node after the source node, you can convert data from String storage to Date storage using the function `to_date(ITEM)`. Conversions can also be performed on other storage formats in the same manner.

Exercises

In these exercises we will practice using the source nodes demonstrated in this lesson. The exercise data file is to be used throughout the course and exists in three formats; comma delimited (*charity.csv*), Excel (*charity.xls*), and Statistics data file (*charity.sav*). If possible, all are to be used in this session. The file originates from a charity and contains information on individuals who were mailed a promotion. The file contains details including whether the individuals responded to the campaign, their spending behavior with the charity and basic demographics such as age, gender and mosaic (demographic) group.

1. Start PASW Modeler, if you haven't done so already, and clear the Stream canvas.
2. Select a Var. File node from the Sources palette and place it on the Stream canvas
3. Edit this node and set the file to *c:\Train\ModelerIntro\charity.csv*. The file contains the field names in the first row, so check the option that instructs PASW Modeler to read these from the file.
4. Return to the Stream canvas by clicking the OK button.
5. Select the Excel node from the Sources palette and place it on the Stream canvas.
6. Edit this node and set the file to *c:\Train\ModelerIntro\charity.xls*. The Excel file contains one worksheet with the field names in the first row. You might want to first review this file in Excel to see the format.
7. Return to the Stream canvas by clicking the OK button.
8. Select the Statistics File node from the Sources palette and place it also on the Stream canvas.
9. Edit this node and set the file to *c:\Train\ModelerIntro\charity.sav*. The Statistics data file contains variable and value labels. Check the options to read both variable names and labels, and both data values and value labels.
10. Return to the Stream canvas by clicking the OK button.
11. To check that all three source nodes are working correctly, connect a Table node to each.
12. Run each stream individually using the Run option in the Context (pop-up menu following a right click) menu.
13. Scroll across the tables and familiarize yourself with the fields in the data set. Is the measurement level for *sex* in the Statistics data source the same as the measurement level for the corresponding field in the other data sources?
When you have finished close the Table windows (click File...Close).
14. We will use one of the data sources in the following exercises. Delete all but one data source. Go through the fields, check their measurement level and change it if necessary.
15. Save this stream in the *c:\Train\ModelerIntro* directory under the name *ExerLesson3.str*.

Lesson 4: Data Understanding

Objectives

- Missing data in PASW Modeler
- Using the Data Audit Node to determine data quality
- Selecting valid records and automatically checking data quality
- Examining the distributions of categorical and continuous fields

Data

To illustrate how PASW Modeler deals with missing information, we use a small data file containing missing values, *SmallSampleMissing.txt*. This file has one record per account held by customers of a financial organization. It contains demographic details on the customer, such as income, gender, and marital status.

For other examples, we will use a version of the data file introduced in the previous lesson, *Risk.txt*. The file contains information concerning the credit rating and financial position of individuals, along with basic demographic information such as marital status and gender.

4.1 Introduction

As we have noted, the first steps in a data-mining project involve gaining an understanding of the business question(s) and objectives to be answered, and formulating a concrete plan of how to proceed throughout the data mining process. One of the next steps after this will be to collect one or more data files for use in the project, including relevant information about the data source, how the data were gathered, etc.

Once you have some data in hand, you are ready for the Data Understanding phase in the CRISP-DM process. In this phase, you are concerned with exploring and becoming thoroughly familiar with the characteristics of your data. You should review the distribution of each field, its range (for continuous fields), outliers, anomalies, and missing values (type and amount). You can also begin looking for interesting simple patterns in the data, especially relationships between a predictor and a target field.

In this lesson, we will consider issues of data quality and postpone the search for relationships until a later lesson.

Data sets always contain problems or errors such as missing information and/or spurious values. Therefore, before data mining can begin, the quality of the data must be assessed. As a general point, the higher the quality of the data used in data mining, the more accurate the predictions and the more useful the results.

PASW Modeler provides several nodes that can be used to investigate the integrity of data. In the following sections we will introduce the Data Audit node to study many characteristics of each field. Then we will look at other nodes that can be helpful in data exploration.

4.2 Missing Data in PASW Modeler

Missing data is ubiquitous in almost every data file. It arises for a variety of reasons, but it must be considered carefully for any data mining project. Although we might expect that missing data should be discarded for modeling, that is not always the case. Many analysts have found that missing data

can be useful information. For example, in database marketing, not knowing something about a potential customer (e.g., income) may still be predictive of behavior.

Whether we find missing data to be helpful or not, the first step is to assess the type and amount of missing data for each field. Only then can we decide how to handle it (thus, if there is little missing data, perhaps it will be simpler to discard those records).

In PASW Modeler, missing data is generically labeled *blanks*. It is important to distinguish between the normal use of the word “blank” and PASW Modeler’s labeling of missing values with that same term.

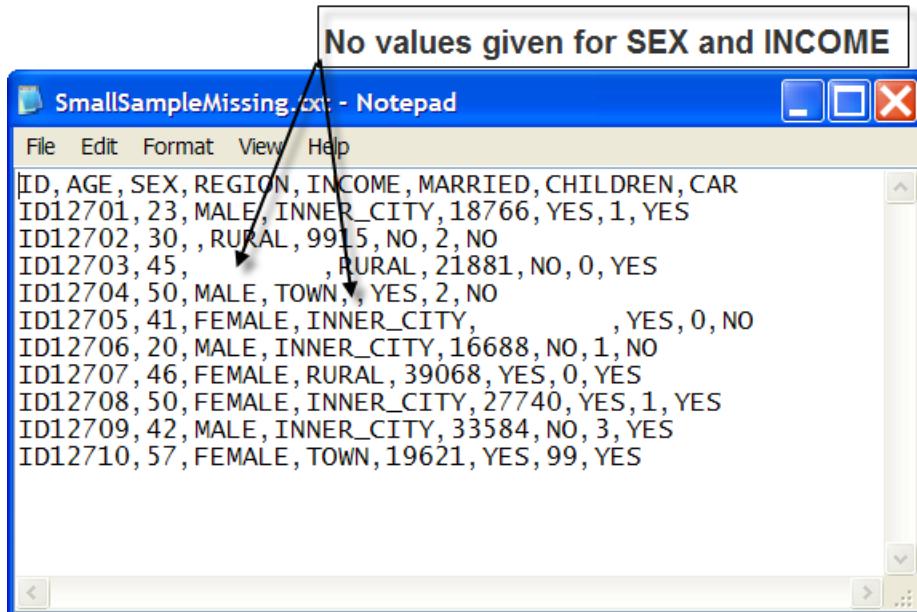
In PASW Modeler there are a number of different types of missing data. First, a field may be blank, i.e., contain no information. PASW Modeler calls such missing information *white space* if the field is string and *null value* (non-numeric) if the field is numeric. In addition, if a non-numeric character appears in a numeric field, PASW Modeler also treats this as a null value since no valid value can be created.

Second, a string field may be empty, which means that it contains nothing (this is common in databases). This type of missing is called an *empty string*.

Finally, predefined codes may be used to represent missing or invalid information. PASW Modeler refers to such codes as *value blanks*.

The file *SmallSampleMissing.txt* (shown in **Figure 4.1**) contains examples of each kind of missing data.

Figure 4.1 Data with Missing Values



A screenshot of a Windows Notepad window titled "SmallSampleMissing.txt - Notepad". The window displays a list of 10 data rows, each consisting of 9 fields separated by commas. The fields are: ID, AGE, SEX, REGION, INCOME, MARRIED, CHILDREN, CAR, and YES/NO. Arrows point from the text "No values given for SEX and INCOME" at the top to the empty "SEX" and "INCOME" fields of the second row (ID12702). The "SEX" field of the third row (ID12703) is also highlighted with a red arrow.

ID	AGE	SEX	REGION	INCOME	MARRIED	CHILDREN	CAR	YES/NO
ID12701	23	MALE	INNER_CITY	18766	YES	1	YES	
ID12702	30		RURAL	9915	NO	2	NO	
ID12703	45			21881	NO	0	YES	
ID12704	50	MALE	TOWN	,	YES	2	NO	
ID12705	41	FEMALE	INNER_CITY	,	YES	0	NO	
ID12706	20	MALE	INNER_CITY	16688	NO	1	NO	
ID12707	46	FEMALE	RURAL	39068	YES	0	YES	
ID12708	50	FEMALE	INNER_CITY	27740	YES	1	YES	
ID12709	42	MALE	INNER_CITY	33584	NO	3	YES	
ID12710	57	FEMALE	TOWN	19621	YES	99	YES	

Note that a value for *SEX* is not given for ID12702 (it is an empty string) and *SEX* has the value “ ” for ID12703. Similarly, a value for *INCOME* is not given for ID 12704 and ID12705 (null values). Furthermore, ID12710 has the value 99 for *CHILDREN* (number of children) because the number of

children was unknown and the database administrator decided to put the value 99 in this field to represent unknown (a value blank).

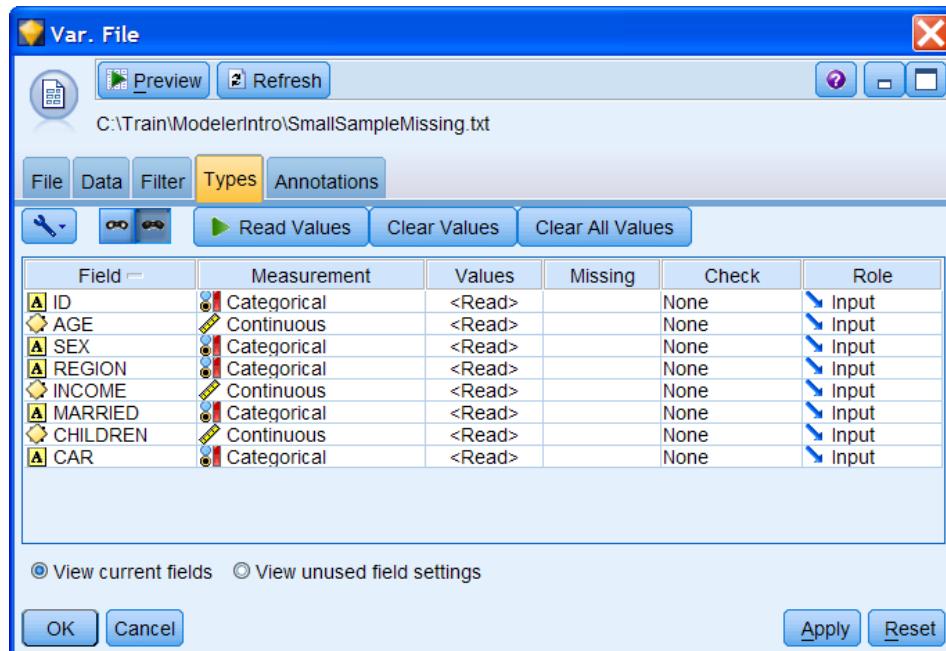
So in this file we have different kinds of missing information. In the next section we will see how PASW Modeler deals with this.

4.3 Assessing Missing Data

To illustrate the handling of missing data we will open *SmallSampleMissing.txt* and assess the quality of the data. We will have PASW Modeler identify the measurement levels while creating a Table.

If the Stream Canvas is not empty, start a new stream by clicking **File...New Stream**
 Select the **Var. File** node and place it on the Stream Canvas
 Double-click to **Edit** the node and set the file to **SmallSampleMissing.txt** held in the
c:\Train\ModelerIntro directory
 Click the **Types** tab, then right-click any field and click **Select All** from the context menu
 Right-click any field and then click **Set Values...<Read>** from the context menu

Figure 4.2 Types Tab with Values Set to Read for All Fields



Click the **Preview** button

Figure 4.3 Data Table Showing Blanks and Missing Values

Preview from SmallSampleMissing.txt Node (8 fields, 10 records)

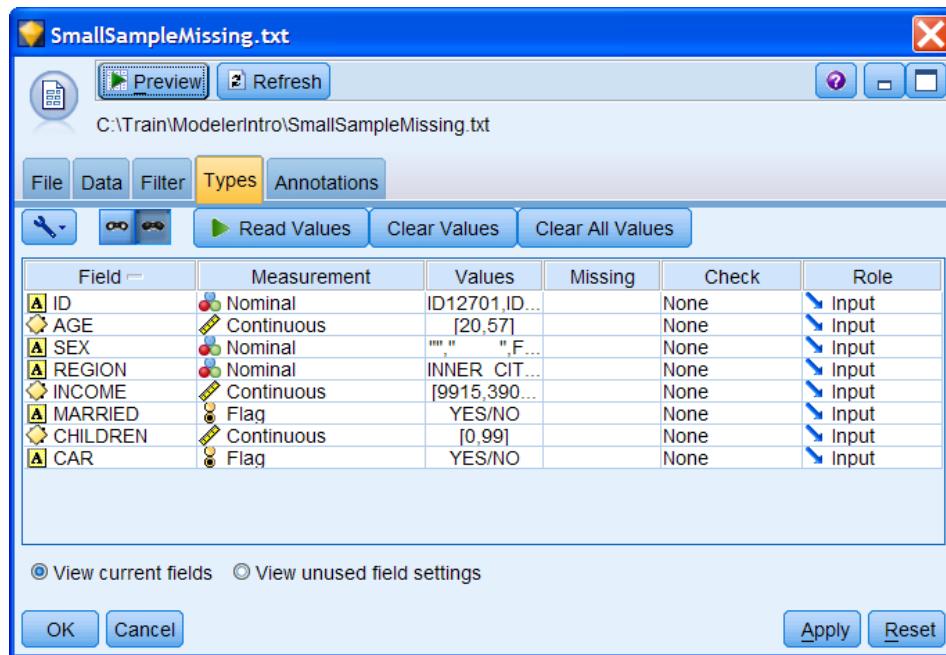
The table shows three examples of missing information.

ID	AGE	SEX	REGION	INCOME	MARRIED	CHILDREN	CAR	
1	ID12701	23	MALE	INNER CITY	18766	YES	1	YES
2	ID12702	30		RURAL	9915	NO	2	NO
3	ID12703	45		RURAL	21881	NO	0	YES
4	ID12704	50	MALE	TOWN	\$null\$	YES	2	NO
5	ID12705	41	FEMALE	INNER CITY	\$null\$	YES	0	NO
6	ID12706	20	MALE	INNER CITY	16688	NO	1	NO
7	ID12707	46	FEMALE	RURAL	39068	YES	0	YES
8	ID12708	50	FEMALE	INNER CITY	27740	YES	1	YES
9	ID12709	42	MALE	INNER CITY	33584	NO	3	YES
10	ID12710	57	FEMALE	TOWN	19621	YES	99	YES

The table shows three examples of missing information.

- *SEX* has been left blank for the records with ID12702 and ID12703 (see the text file: ID12702 has no value for *SEX* in the text file, ID12703 has spaces " " as the value in the text file).
- *INCOME* has a non-numeric value, appearing as *\$null\$* in the table, for records ID12704 and ID12705. The value *\$null\$* is assigned by PASW Modeler in case a numeric value is undefined, and it is considered by PASW Modeler as missing information. The reason that PASW Modeler assigned the *\$null\$* value, instead of leaving it empty as with *SEX*, is that *INCOME* is numeric (as opposed to the string type of *SEX*).
- *CHILDREN* has a user input missing value of 99 for record ID12710. It is not showing as missing (a blank) because we haven't yet defined it as such in PASW Modeler.

Click to close the Preview table
Go back to the **Types** tab of that **Var. File** node

Figure 4.4 Types Tab: File with Missing Data

Notice that *SEX* is defined as Nominal because four discrete values were found, as displayed in the Values column (FEMALE, MALE, a set of space characters (white space), and an empty string), and that *CHILDREN* has a range of 0 through 99. PASW Modeler will not automatically make value blanks missing, and it will not automatically treat white space or empty strings as missing either for string fields. Thus, when appropriate, you need to specify missing values.

The Data Audit node reports on such missing values even if they are not declared as missing in the Types tab of a source node (or a Type node). However, if you know that such values should be identified as missing values, then there is an advantage in declaring them before data are read, since they then will not appear on the Values list for the field. In that case, *SEX* would be properly identified as Flag with values FEMALE and MALE. We will return to this point later.

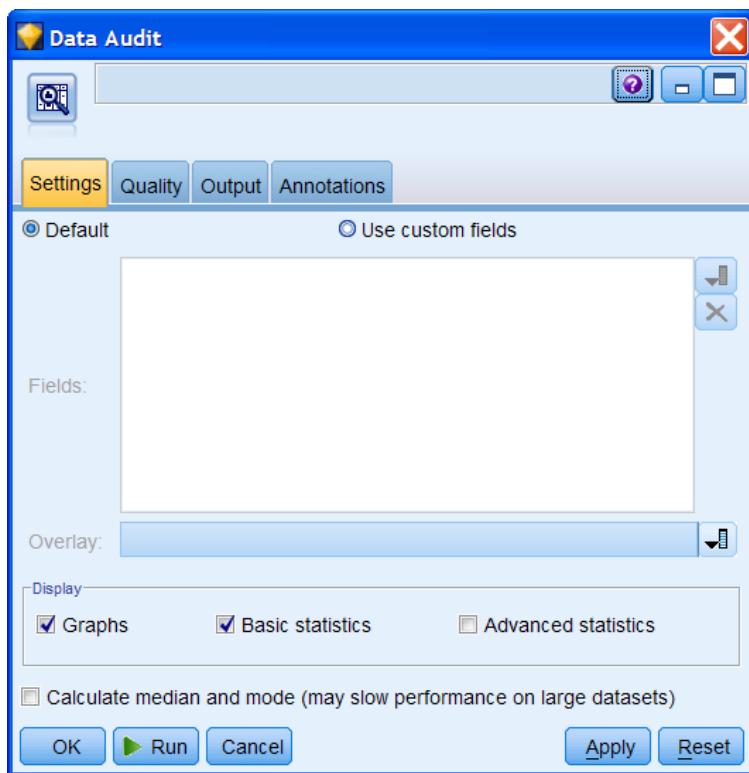
4.4 Using the Data Audit Node for Missing Data

The Data Audit node provides a report about the missing values in a data stream, plus lots of other information on a field's distribution. It checks for missing values or blanks and is located in the Output palette. It is a terminal node (no connections can lead from a terminal node). The Data Audit node can take into account all of the missing value definitions previously mentioned.

In this section we will focus on only the missing value reports from the Data Audit node.

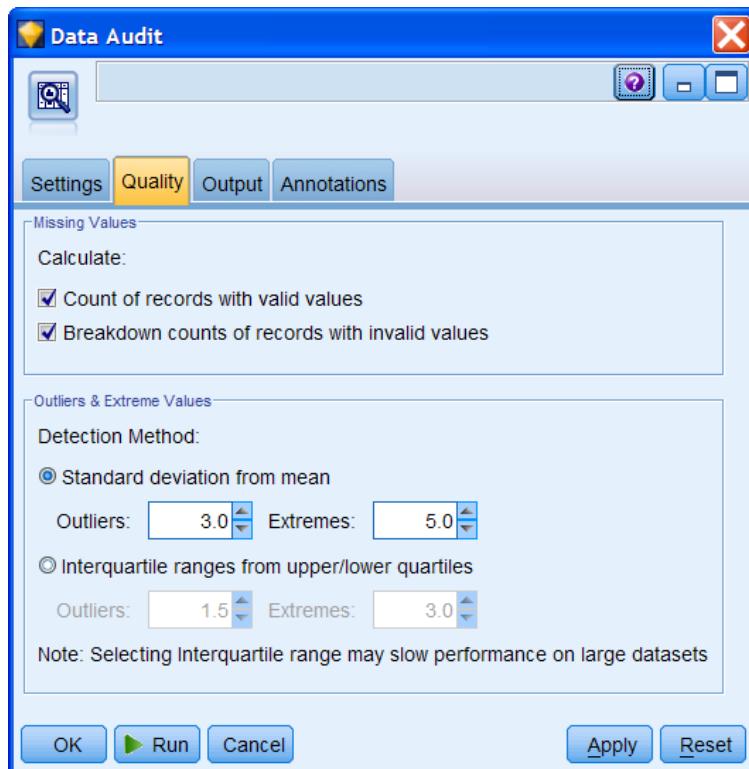
- Click **OK** to close the Var. File node
- Place a **Data Audit** node from the Output palette into the Stream Canvas and connect the **Var. File** node to it
- Edit the **Data Audit** node

The default view shows the usual box in which to select fields for analysis, plus check boxes to control the statistics and graphs produced. We will use only the selections in the Quality tab at this point.

Figure 4.5 Data Audit Node Dialog

Click the **Quality** tab

The choices in the Quality tab control checking for missing data and also what values are considered to be outliers and extreme for data checking.

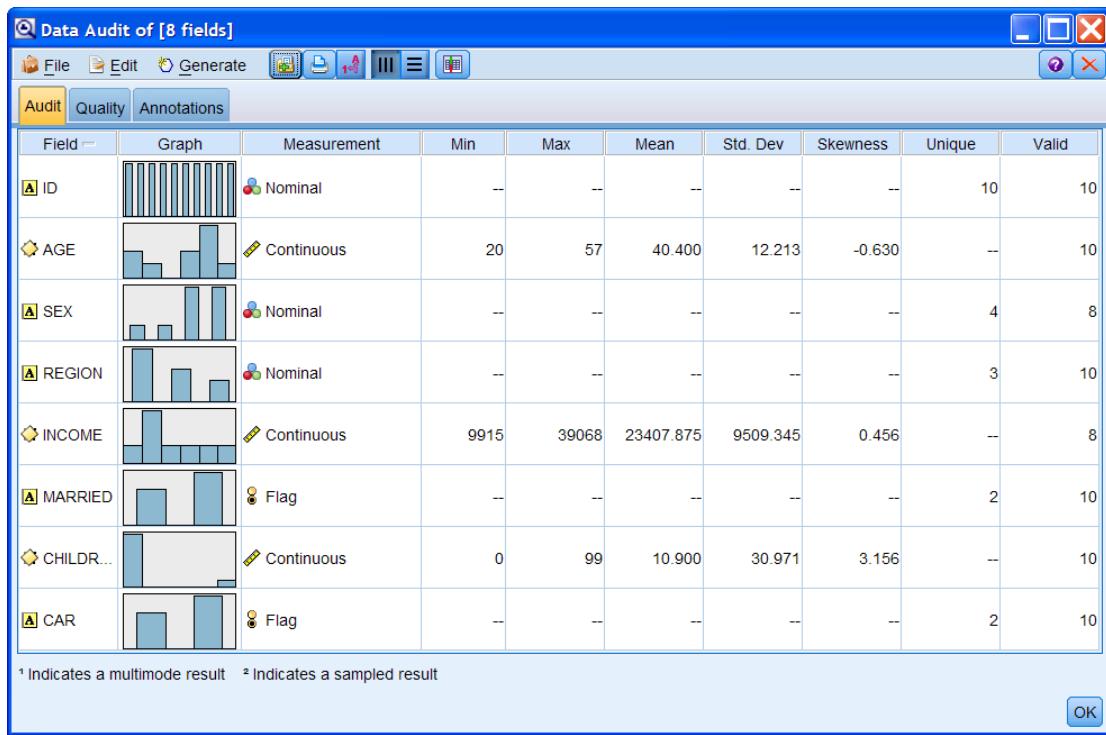
Figure 4.6 Quality Tab in Data Audit Node

Check boxes control what the Data Audit node will report as missing. These include:

- **Count of records with valid values.** This option shows the total number of records with valid values for each selected field. Null values, value blanks, white space, and empty strings are always treated as invalid values.
- **Breakdown counts of records with invalid values.** This option shows the number of records with each type of invalid value for each field.

By default, both missing values check boxes are turned on. We will accept the defaults, which also include running the audit on all the fields.

Click the **Run** button

Figure 4.7 Basic Data Audit Output

The Data Audit node produces several types of information for each field, plus an appropriate graph of the field's distribution. The last column displays the number of valid records for each field, i.e., those with no missing data. We can see that, in this small file of 10 cases, all the fields have 10 valid records except *SEX* and *INCOME*. These are the two fields with various types of missing data (except for the value blank of 99 for *CHILDREN*), and the Data Audit node correctly found this missing data and reported that *SEX* and *INCOME* each have 8 valid records.

Since the value blank of 99 is not defined for *CHILDREN*, we can observe that the statistics produced for that field are incorrect, such as the mean (a very unlikely value of 10.9 children).

Further information is available in the Quality tab.

Click the **Quality** tab

The resulting window contains many columns, and so Figure 4.8 displays the window in two parts. If we focus first on the *% Complete* column, we see a summary of the number of valid values for each field, expressed as a percentage of the total number of records.

This is followed by columns for the four types of missing values in PASW Modeler: Null Value, Empty String, White Space, and Blank Value. There are two null values for *INCOME* because of the spaces in the raw data in this numeric field. There are two white space values for *SEX* and one empty string. This is because an empty string is also considered to be white space by PASW Modeler for purposes of classifying missing values.

Figure 4.8 Quality Tab Output from Data Audit Node

Data Audit of [8 fields]

Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method
ID	Nominal	--	--	Never	Fixed	
AGE	Continuous	0	0 None	Never	Fixed	
SEX	Nominal	--	--	Never	Fixed	
REGION	Nominal	--	--	Never	Fixed	
INCOME	Continuous	0	0 None	Never	Fixed	
MARRIED	Flag	--	--	Never	Fixed	
CHILDREN	Continuous	0	0 None	Never	Fixed	
CAR	Flag	--	--	Never	Fixed	

Data Audit of [8 fields]

Impute Missing	Method	% Complete	Valid Records	Null Value	Empty String	White Space	Blank Value
ver	Fixed	100	10	0	0	0	0
ver	Fixed	100	10	0	0	0	0
ver	Fixed	80	8	0	1	2	0
ver	Fixed	100	10	0	0	0	0
ver	Fixed	80	8	2	0	0	0
ver	Fixed	100	10	0	0	0	0
ver	Fixed	100	10	0	0	0	0
ver	Fixed	100	10	0	0	0	0

Note

This report can be sorted by values within any of the columns by clicking on the column header; choices will cycle through ascending order, original order, and descending order. When sorted by ascending or descending order, an upward or downward pointing icon indicates the sort order, while a dash indicates the original order.

Two other statistics are provided here. PASW Modeler reports that 75% of the fields are complete and have no missing data; second, that 60% of the records are complete and have valid values for every field. This latter number can be quite critical for modeling. If we don't somehow adjust for missing data—by imputing or estimating missing values for some fields—and we use all these fields for modeling, we will have only 60% of the cases available. This is quite a reduction in number of records and so could be of concern.

In data mining, we often have many thousands of records, so our total number of cases may still be adequate. For example, imagine that we initially had 100,000 records and now would have only 60,000. That number would likely still be adequate for modeling.

The more crucial concern is whether there is a pattern to the missing data such that there is a difference between the records with missing data and those with valid data. If there is, then our model can be misestimated and not apply to the full population of interest.

Note

There is an additional, perhaps fortunate, complication concerning missing data. Several of the modeling algorithms either use missing data directly (such as C5.0), or adjust for it automatically before constructing a model (such as Neural Nets). As a consequence, you may not feel the need to handle missing data yourself in some situations. Further discussion on this issue is reviewed in the modeling courses.

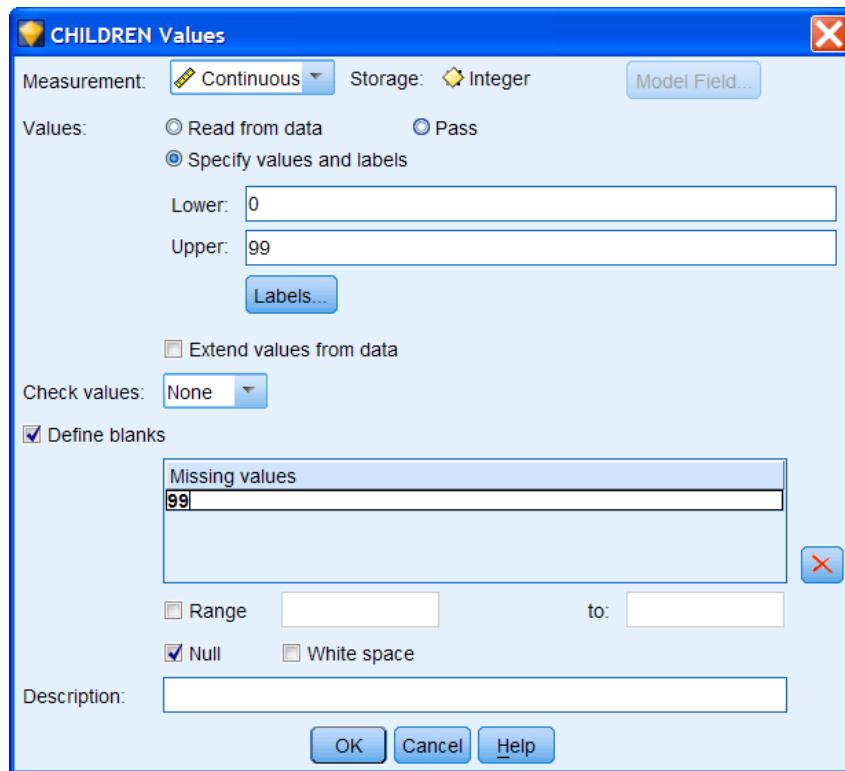
Defining Value Blanks

We turn now to the matter of how to define the missing value 99 (the value blank) for *CHILDREN*. To do so we return to the Types tab in the source node (or we could add a Type node to the stream).

Close the Data Audit window
Edit the **Var. File** node (double-click the Var. File node)

The Missing column in the Types tabs controls whether some data values within a field will be defined as missing.

Click the cell in the **Missing** column and **CHILDREN** row
Select **Specify** from the drop-down menu
Click the **Define blanks** check box
Click in the cell under **Missing values** and type **99**

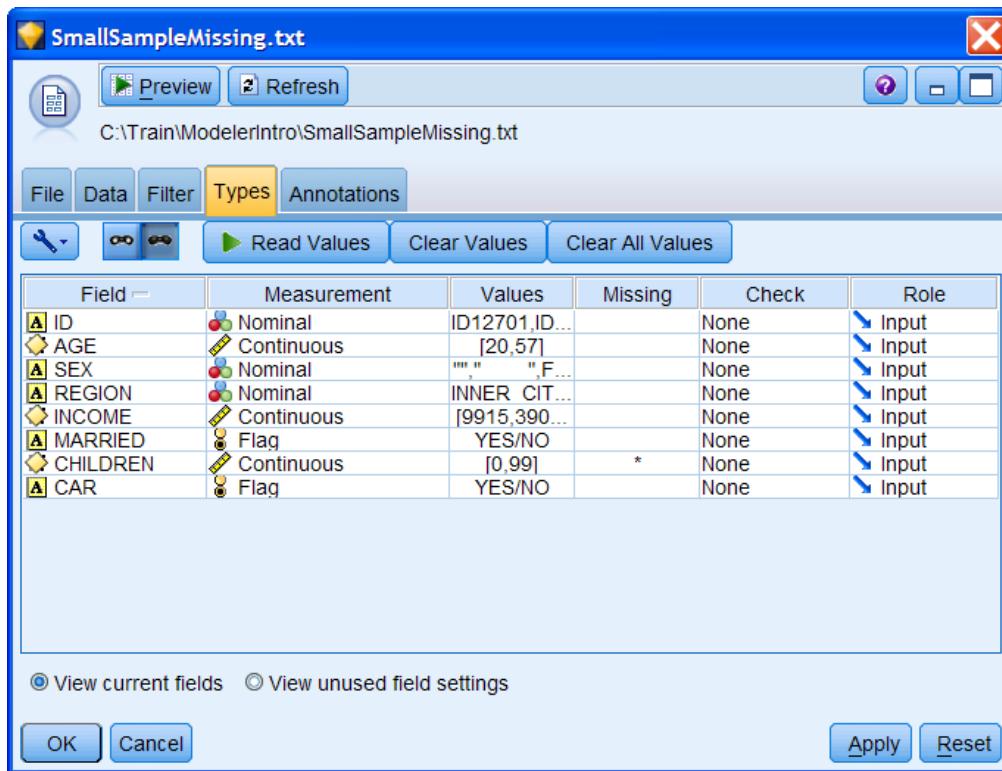
Figure 4.9 Defining a Value Blank

Notice that the *Null* check box is automatically checked, so non-numeric values will be considered missing for the *CHILDREN* field. The *White space* check box, while not relevant for a numeric field, would serve to define white space (no visible characters, including empty strings) as missing for string fields (for example, *SEX*).

Click **OK** to return to the Var. File node dialog box

The cell in the Missing column for *CHILDREN* has an asterisk. This indicates that missing values have been defined for this field.

Figure 4.10 Missing Values Defined for CHILDREN

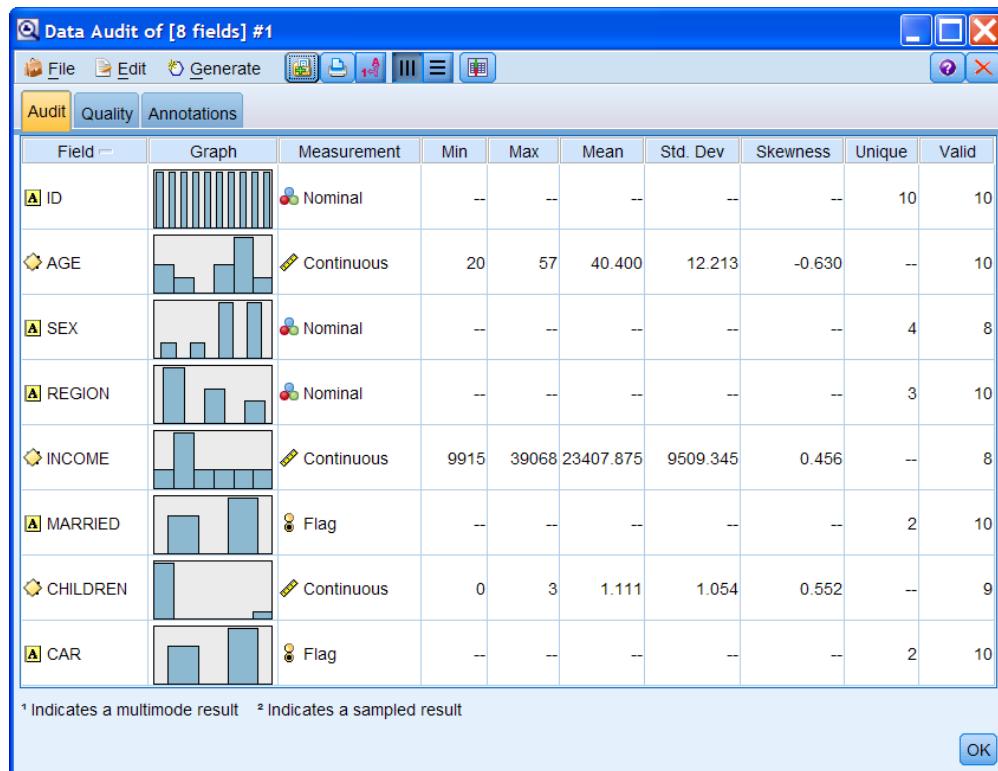


Having defined 99 as missing for *CHILDREN*, let's run the Data Audit node again.

Click **OK** to close the Var. File dialog box
Right-click the **Data Audit** node and select **Run**

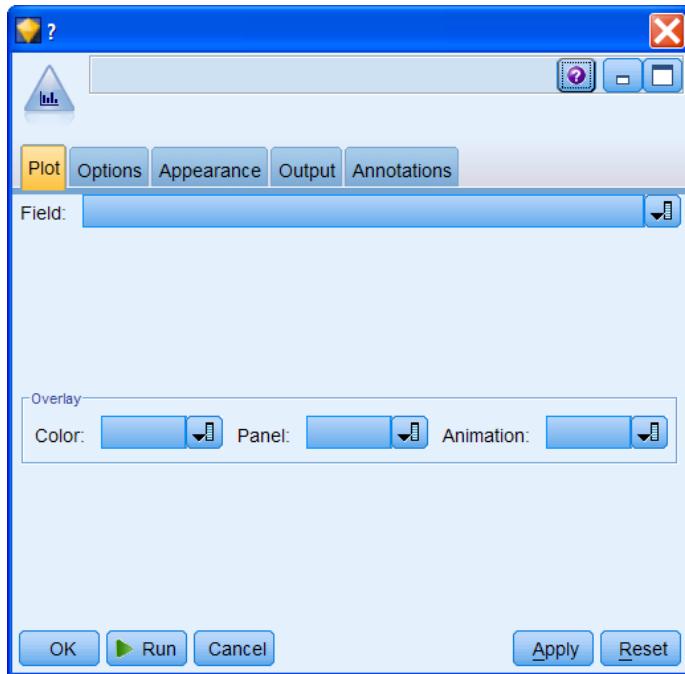
In the Audit tab, notice that the number of valid records for *CHILDREN* is now 9, so the Data Audit node recognizes the missing value specification we made.

You can also click on the Quality tab and see that the missing value for *CHILDREN* is counted as a Blank Value (not shown).

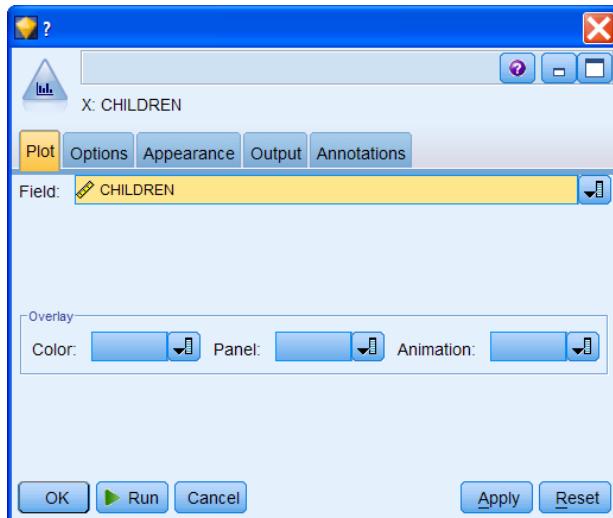
Figure 4.11 Data Audit Output Window with Missing Value for CHILDREN

It is important to understand that defining a missing value in a Source node or Type node doesn't necessarily have a direct effect on how PASW Modeler treats that value for a field in all nodes. We can easily see this by requesting a histogram of *CHILDREN* (although a histogram is available within the Data Audit node, we create one separately to make this point explicit).

- Close the Data Audit window
- Click on the **Graphs** palette
- Click on the **Histogram** icon and then click in the stream to the right of the Source node
- Connect the **Var. File** node to the **Histogram** node
- Double-click on the **Histogram** node to edit it
- Click the Field Chooser button

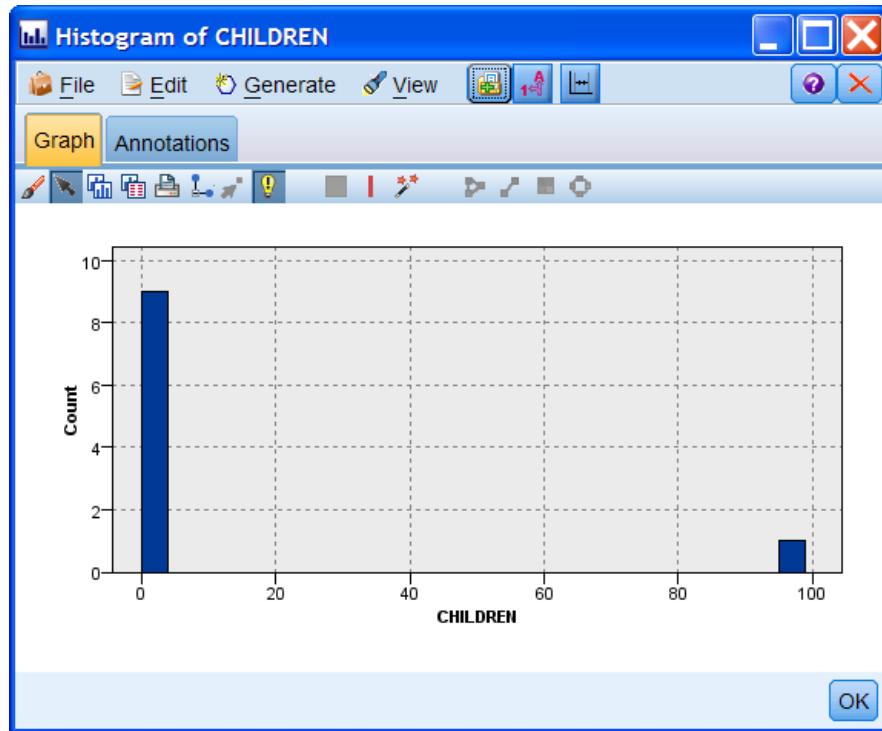
Figure 4.12 Selecting a Field for the Histogram

Select **CHILDREN**

Figure 4.13 Requesting Histogram for CHILDREN

Click **Run**

Something surprising occurs. The histogram includes the value of 99, even though we defined it as missing! PASW Modeler certainly knows this value is missing, but for some nodes—the Graph nodes, nodes that do field transformations, and some Modeling nodes—the missing value definition is either ignored, as here, or the value is adjusted (as noted above in some modeling nodes).

Figure 4.14 Histogram of CHILDREN

Given this behavior, often a more direct method of handling missing data is to remove it from the stream with a Select node, which can be generated automatically from the Data Audit output window, as we demonstrate below.

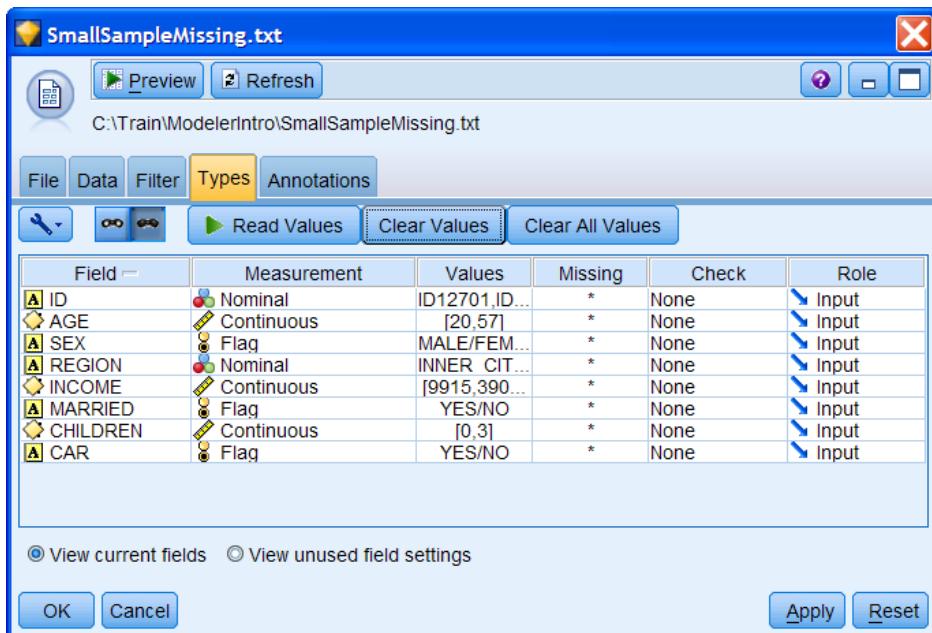
In any case, if we want PASW Modeler to recognize the missing data values for *SEX* and *INCOME*, we should declare them as missing in the Types tab of the Source node. The Data Audit node reports on missing values but it doesn't define them. There is a very easy way to accomplish this.

- Close the Histogram window
- Edit the **Var. File** node
- Click **Clear Values** button on the Types tab
- Right-click any field, and then click **Select All** on the context menu
- Right-click any field, and then click **Set Missing...On** (not shown)

An asterisk appears in the Missing column for each field. This indicates that missing values have been declared. As we will see, by setting *Missing On*, all selected numeric fields will have the null value declared as missing and all selected string fields will have white space and the null value declared as missing. Thus you can quickly declare missing values for many fields. Blank fields (user-defined missing values) need to be defined manually. To view the result:

- Click **Read Values** button

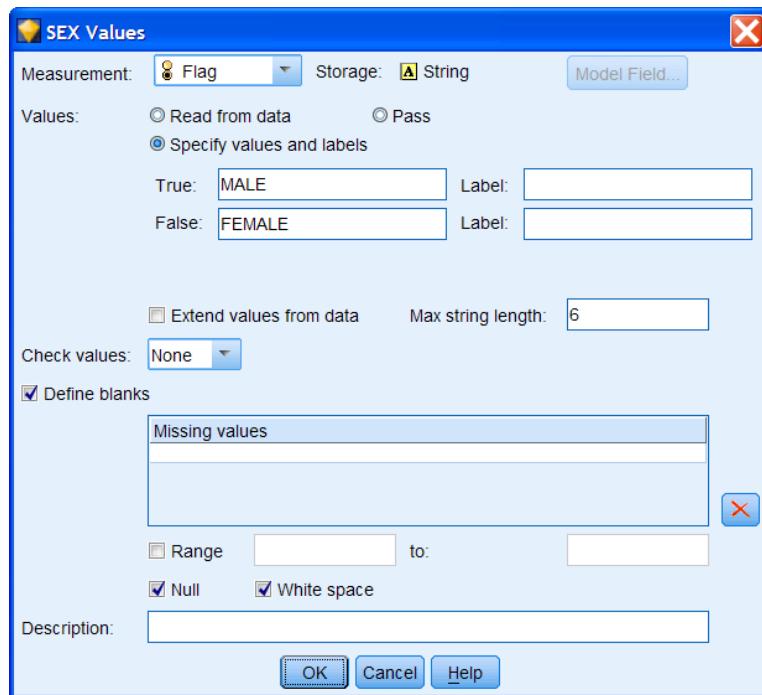
Figure 4.15 Types Tab with Missing Values Declared



Compared to the original Types tab (Figure 4.4), there are differences for the *SEX* and *CHILDREN* fields. Since white space is considered missing for *SEX*, the measurement level for *SEX* is now correctly identified as Flag with values FEMALE and MALE. Also, the range for *CHILDREN* is now 0 through 3 because 99 is declared as a blank value *and* the data have been reread. The point to remember is that declaration of missing values can influence the autotyping of string fields and the range values for continuous fields, so it is advantageous to do this early in the data mining process (that is why we placed the discussion of missing values in this lesson just after reading in data to PASW Modeler). To verify that missing values were, in fact, declared for all fields:

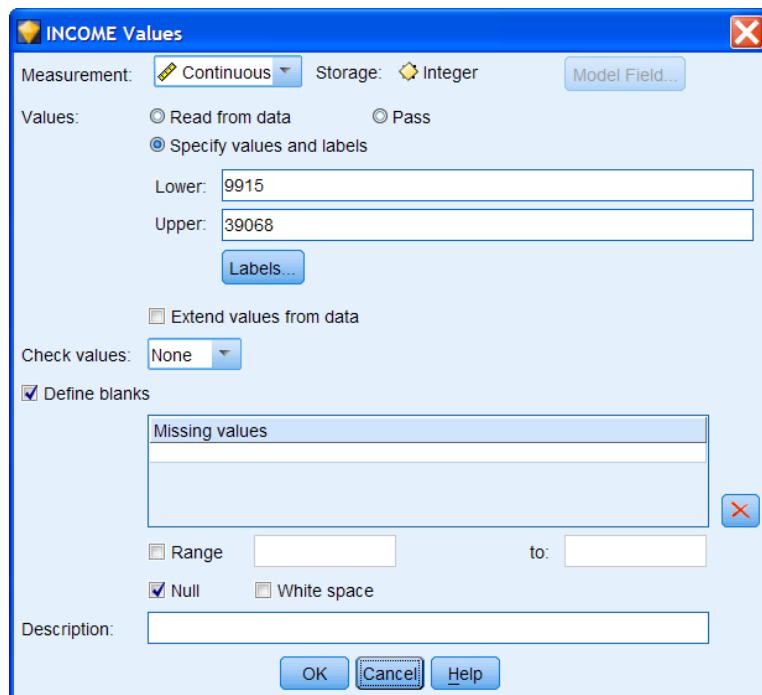
Click the cell in the **Missing** column and **SEX** row, and then click **Specify**

The *Define blanks* check box is checked, along with the *Null* and *White space* check boxes. Turning missing values on for a string field automatically declares null values and white space as missing. It may seem odd that null values are declared as missing for a string field, but some databases code empty string fields as null, and so Null is checked to accommodate this. And recall that white space also includes empty string.

Figure 4.16 Missing Values Defined for a String Field (SEX)

Click the **Cancel** button

Click the cell in the **Missing** column and **INCOME** row, and then click **Specify**

Figure 4.17 Missing Values for a Numeric Field (INCOME)

After setting **Missing On**, the **Define blanks** and **Null** check boxes are checked for the selected numeric fields. In addition, user-defined missing values can be declared (as we did for **CHILDREN**).

In summary, if you want null values, white space, and empty strings to be treated as missing within PASW Modeler, then selecting *Set Missing... On* from the right-click context menu is a convenient way to accomplish it.

Generating a Select Node for Valid Records

If you have a very large data file, you may be able to delete all records with missing data from the stream and develop models on only complete records. The Data Audit browser provides a direct method to generate a Select node to accomplish this task. Similarly, you can generate a Filter node that removes fields with a large amount of missing data.

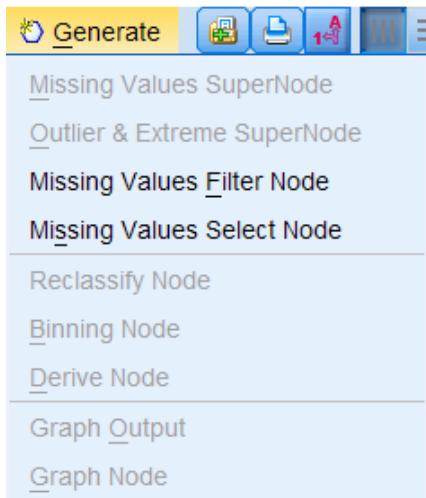
Note

Before we demonstrate this feature, it should be emphasized that how you handle missing data when you are developing a model has some complications. Let's suppose that you use only valid records to develop a model to predict an outcome. This means that you can't use the model successfully on new data unless all the fields have non-missing data, but that isn't very likely, since the training data had missing data before you deleted them. Unless the number of records with missing data is negligible, following this scenario would prevent you from making predictions on an appreciable fraction of records in new data, which is not desirable.

We need to rerun the Data Audit node to use the changes we just made to missing definitions.

- Click **OK** to close the INCOME values window
- Close the Var. File source node
- Right-click on the **Data Audit** node and select **Run**
- In the Data Audit window, click the **Quality** tab
- Click the **Generate** menu

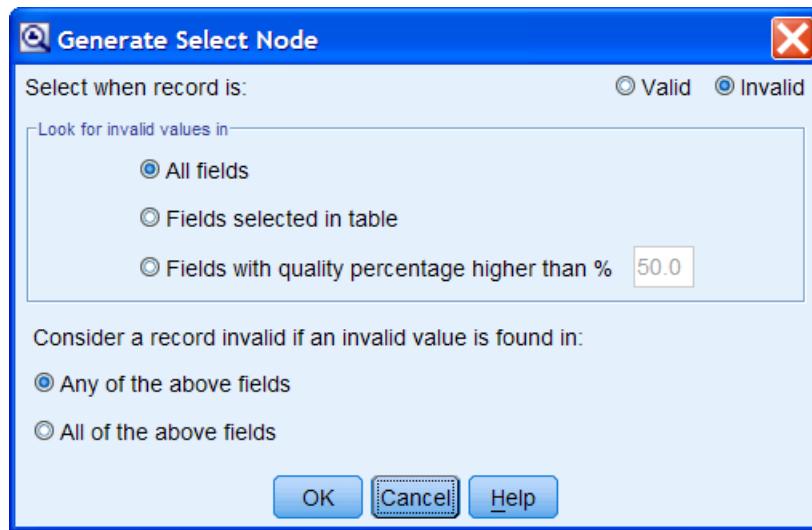
Figure 4.18 Generate Menu within the Data Audit Output Window



There are two active choices, *Missing Values Filter Node* and *Missing Values Select Node*. The first will filter out fields, the second records. We will use the second option to generate a Select node. Suppose we want to view a table of all records with no missing values, of any kind, on any field.

- Click **Missing Values Select Node** on the Generate menu

Figure 4.19 Generate Select Node Dialog (from Quality Tab Output)



Select when record is: specifies whether records should be kept when they are *Valid* or when they are *Invalid*. In our case, we are interested in viewing the records without missing values in a table, so we use the first option.

The setting for *Look for invalid values in:* specifies where to check for invalid values:

- *All fields*: the Select node will check all fields for missing values.
- *Fields selected in table*: the Select node will check only the fields currently selected in the Quality output table.
- *Fields with quality percentage higher than %*: the Select node will check fields for which the percentage of non-missing records is greater than the specified threshold. The default threshold is 50%.

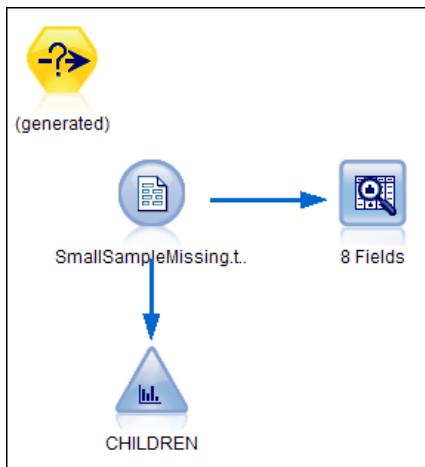
The setting for *Consider a record invalid if an invalid value is found in:* specifies the condition for identifying a record as invalid:

- *Any of the above fields*: the Select node will consider a record invalid if any of the fields specified above contains a missing value for that record.
- *All of the above fields*: the Select node will consider a record invalid only if all of the fields specified above contain missing values for that record.

In our example we want to include a record in our table if it has *no* missing values on any field, so all fields must be checked for any missing values. We can use the default settings, with one change to request valid records.

Click **Valid** in Select when record is: area
 Click **OK**
 Close the Data Audit Output window

PASW Modeler generates a Select node and puts it in the upper left corner on the Stream Canvas.

Figure 4.20 Generated Select Node from Data Audit Output Window

To preview the data:

Connect the generated **Select** node to the **Source** node, and then double-click the **Select** node
Click the **Preview** button

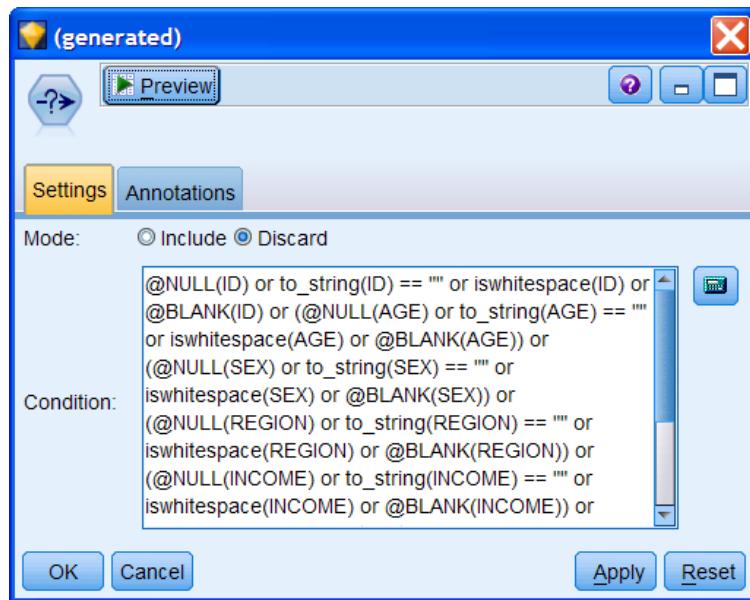
The Table (shown in Figure 4.21) lists 5 records, so there are only five records, or 50% of this small file, with no missing values. Specifically, we see that there are no missing values for *SEX*, *INCOME*, or *CHILDREN*. This data stream could now be used for modeling without a concern about how to handle missing values since they have been removed from the data.

Figure 4.21 Records with no Missing Data

ID	AGE	SEX	REGION	INCOME	MARRIED	CHILDREN	CAR	
1	ID12701	23	MALE	INNER CITY	18766	YES	1	YES
2	ID12706	20	MALE	INNER CITY	16688	NO	1	NO
3	ID12707	46	FEMALE	RURAL	39068	YES	0	YES
4	ID12708	50	FEMALE	INNER CITY	27740	YES	1	YES
5	ID12709	42	MALE	INNER CITY	33584	NO	3	YES

If you are interested in how the selection has been done by PASW Modeler, just go back to the generated Select node.

Close the Preview window
Go back to the generated **Select** node

Figure 4.22 Generated Select Node

PASW Modeler generated this lengthy expression for us. Note that we can easily produce a table with only records containing at least one missing value by choosing the *Include* mode in the Select node dialog. We learn more about data selection in Lesson 10.

Click **OK** to close the Select node

4.5 Auto Checking for Missing and Out-of-Bounds Values

In the above sections we demonstrated how to specify missing values and select valid records. In many cases the number of fields and records in a data file are vast and this can be a time consuming process. To address this, the Types tab (and Type node) contains an automatic checking process that examines the values in fields to determine whether they comply with the current type and bounds settings.

One point to be aware of is that if a value is specified as a blank (user-defined missing value) by the Type node, checking will ignore it. Blank values within a field's range are regarded as part of the field's range and will therefore not trigger this checking process.

When an out-of-bounds or illegal value is found, the checking process can be set to perform one of five possible actions (in addition to the default of doing nothing):

Table 4.1 Effects of Various Type Node Data Check Settings

Setting	Effect
NONE	The default option. Records are passed through the node without checking.
NULLIFY	Values falling outside those specified in the Type node will be converted to the system null value (\$null\$).
COERCE	Values falling outside those specified in the Type node will be converted to a legal value. Legal values are defined by the measurement level of the field and are given in Table 4.2.
DISCARD	The entire record is discarded if an illegal value is found.
WARN	The number of illegal values encountered is reported in a message window.
ABORT	The first illegal value encountered will result in an error and stream execution will be aborted.

Table 4.2 indicates how the Coerce setting operates for different levels of measurement.

Table 4.2 Result of Coerce Setting in the Check Column of the Type Node

Field Measurement Level	Illegal Value	Illegal Value Coerced to:
Flags	Anything other than True or False	False
Nominal, Ordinal	Any unknown value	First member of set's values
Continuous	Greater than upper bound	Replaced by upper bound
Continuous	Less than lower bound	Replaced by lower bound
Continuous	Non-numeric value	Midpoint of range

In this example we will use checking in the Source node to coerce illegal values for income. Say, for demonstration purposes, that income should be between 10,000 and 100,000.

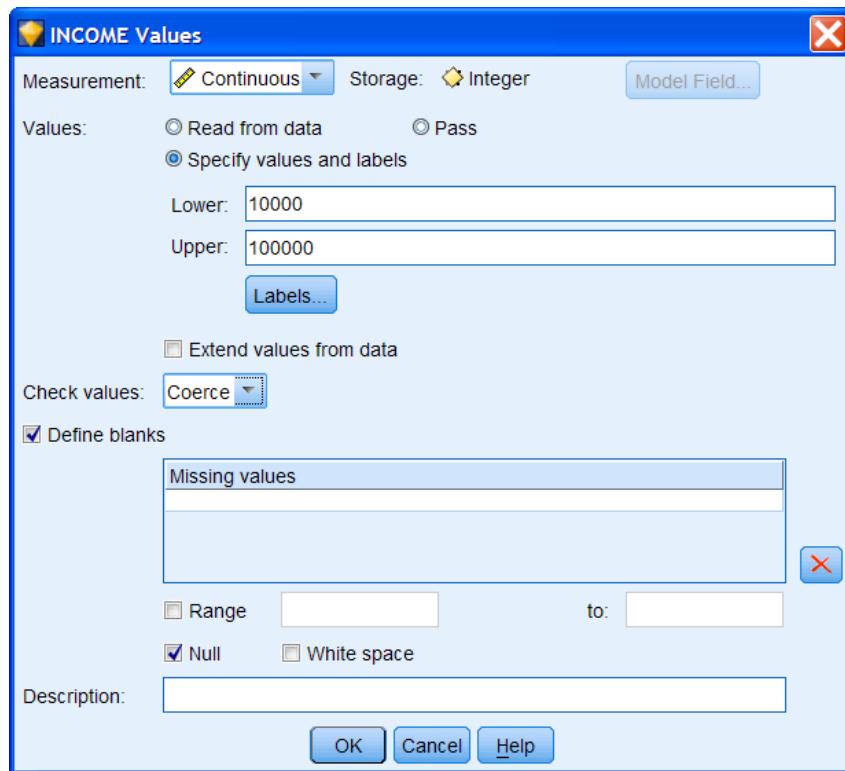
Edit the **Var. File** node

Click in the **Missing** column for **INCOME** and choose **Specify** from the drop-down menu

Click **Specify values and labels** button (if necessary)

Specify **10000** as **Lower**, **100000** as **Upper** values (no commas)

Select **Coerce** from the **Check values** drop-down list

Figure 4.23 Coercing Values for INCOME

Click **OK**

In the Types tab, the Check column now lists *Coerce* for *INCOME* (not shown).

We can now preview the result.

Click the **Preview** button

Figure 4.24 Table Node after Coercing Income Values

	ID	AGE	SEX	REGION	INCOME	MARRIED	CHILDREN	CAR
1	ID12701	23	MALE	INNER CITY	18766	YES	1	YES
2	ID12702	30		RURAL	10000	NO	2	NO
3	ID12703	45		RURAL	21881	NO	0	YES
4	ID12704	50	MALE	TOWN	\$null\$	YES	2	NO
5	ID12705	41	FEMALE	INNER CITY	\$null\$	YES	0	NO
6	ID12706	20	MALE	INNER CITY	16688	NO	1	NO
7	ID12707	46	FEMALE	RURAL	39068	YES	0	YES
8	ID12708	50	FEMALE	INNER CITY	27740	YES	1	YES
9	ID12709	42	MALE	INNER CITY	33584	NO	3	YES
10	ID12710	57	FEMALE	TOWN	19621	YES	99	YES

Coercing has made a replacement for *INCOME* for ID 12702, which before had a value of 9915. Values formerly below 10,000 are now exactly 10,000. Perhaps counterintuitively, no replacement is made for records with income values of \$null\$. This is because the Define blanks check box was selected for *INCOME*, along with the Null check box (by default). This means that PASW Modeler recognizes the value of \$null\$ as a defined value and thus doesn't see it as an illegal value. To see this, you can try turning off Define blanks and then executing the table again.

The auto-checking feature of the Type node is an alternative to use when you want to automatically remove missing data from a stream, or you want to make extreme and outlying values missing or coerce them to a preset value.

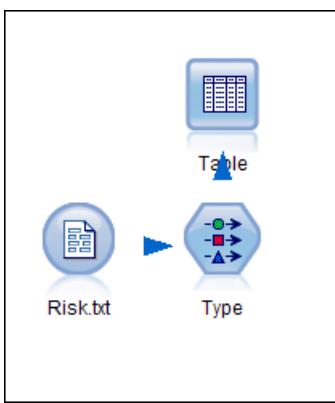
Having seen the different types of missing values and how they are declared and reported, we now must examine the general distribution and other values for the data fields; we will use a complete data set for this purpose (see the Appendix to this lesson for some general advice on handling missing data).

Opening a Stream File

We now switch to a larger and richer version of the data file with the name *Risk.txt*. In addition to having more records (4117) and no missing values, it is a tab-delimited file. Rather than modifying the current Var. File node to read this file or building a new stream, we will open a previously saved stream. First we clear the current stream.

Close the Preview table
Right-click on the stream canvas and select **Close Stream**
Click **No** to save this stream
Click **File...Open Stream**
Navigate to the **c:\Train\ModelerIntro** directory (if necessary)
Double-click on **Riskdef.str**

Figure 4.25 Stream to Read Risk.txt Data



Notice that this stream contains a Type node, which is an alternative to using the Types tab in a Source node. The Type node is not necessary here but would be needed to properly type fields modified or added in the course of a PASW Modeler stream.

4.6 Field Distributions and Summary Statistics

A data set could contain 100% complete data but still have inaccurate entries or outliers (see Lesson 5). It is therefore important before modeling takes place to see how records are distributed for the fields in the data set. This can identify values which, on the surface, appear to be valid, but when compared to the rest of the data are either out of range or inappropriate.

Furthermore, you will want to review the general distribution of a field. When a data field is of categorical (flag, set, or ordinal), it is of primary interest to see how many unique values there are and how the records are distributed among the categories of that field. You look for categories with either very few, or very many, records. In either instance, this can be a problem for modeling. For numeric fields there is usually interest in the distribution of the data values (histogram) and summary statistics (mean, minimum, maximum, and standard deviation). You look for odd distributions (such as those that are highly skewed). Odd distributions don't hinder some data mining algorithms (such as decision trees) but can be a problem for the more statistically based techniques (such as linear regression).

PASW Modeler provides a number of ways of examining the distribution of data fields. In this section we continue to use the Data Audit node.

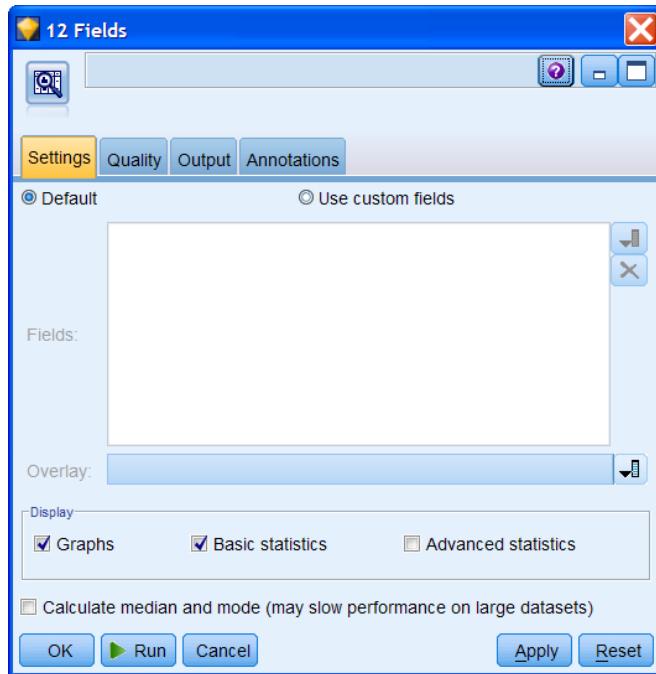
Run the **Table** node (right-click Table node, then click **Run**)

Figure 4.26 Data from Risk.txt

ID	AGE	INCOME	GENDER	MARITAL	NUMKIDS	NUMCARDS	HOWPAID
1	100756	44	59444	m	married	1	2
2	100668	35	59692	m	married	1	1
3	100418	34	59508	m	married	1	1
4	100416	34	59463	m	married	0	2
5	100590	39	59393	f	married	0	2
6	100657	41	59276	m	married	1	2
7	100702	42	59201	m	married	0	1
8	100319	31	59193	f	married	1	2
9	100666	28	59179	m	married	1	1
10	100389	30	59036	m	married	1	1
11	100758	38	58914	m	married	0	1
12	100695	36	58878	f	married	1	1
13	100698	42	58785	f	married	0	2
14	100769	44	58529	m	married	0	1
15	100376	33	58505	f	married	0	2
16	100796	45	58381	m	married	1	1

There are several string fields (*GENDER*, *MARITAL*, etc.) and several numeric fields (*AGE*, *INCOME*, etc.).

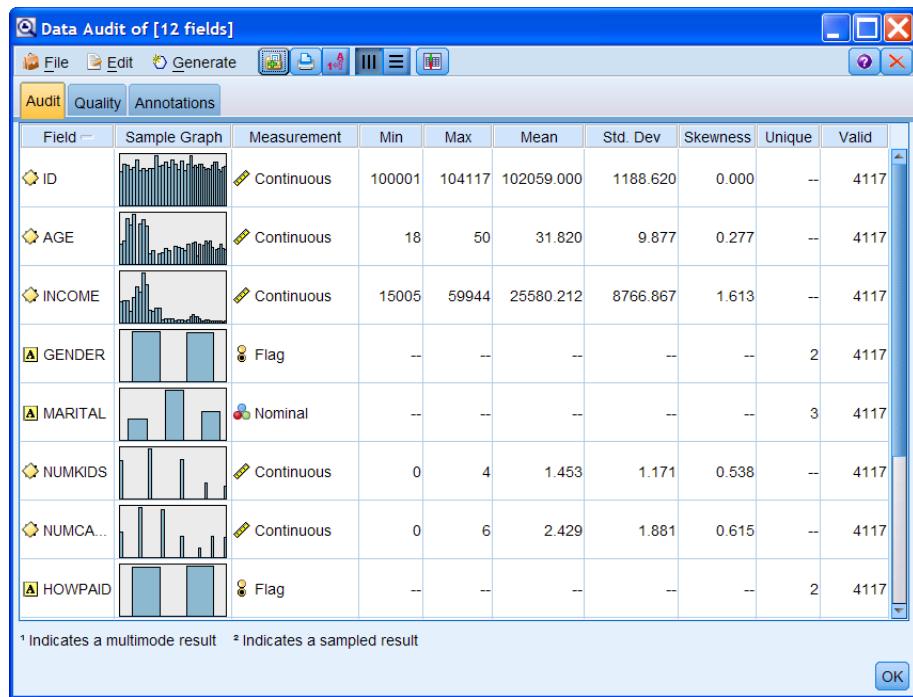
- Close the Table output window
- Click the **Data Audit** node in the **Output** palette
- Click in the Stream Canvas to the **right** of the **Type** node
- Connect the **Type** node to the **Data Audit** node
- Double-click the **Data Audit** node

Figure 4.27 Data Audit Settings Dialog

By default, all fields are included in the data audit analysis. However, the *Field Chooser* button  can be used to select specific fields for analysis when the *Use custom fields* option is chosen. If custom fields are selected, the *Overlay* field list allows the distribution of a categorical field to appear over the distribution of the fields selected in the *Field* list. For example, a distribution of marital status with credit risk status as an overlay would yield a graph showing the number of records within each category of marital status broken down by credit risk category.

The Display group controls whether graphs are created and which summary statistics will be calculated. Since median and mode statistics require more computational resources than the basic statistics, they constitute a separate option.

Click **Run** button 

Figure 4.28 Data Audit Output

Each row of the Data Audit output represents a field and the columns contain graphs, type information, and statistical summaries. Under default settings in the Data Audit node, every field will have a graph, type information, and a summary of the number of records with valid values for that field (*Valid* column), as we saw in the missing values discussion. For continuous fields, the graph in the *Graph* column is a histogram, while categorical (flag, nominal, ordinal) fields are graphed using bar charts (in PASW Modeler they are called distribution charts). Graphs are displayed as thumbnails in the initial report, but full-sized graphs and graph nodes can also be generated by double-clicking on the thumbnails.

The summary statistics for a continuous field are minimum (*Min*), maximum (*Max*), mean, standard deviation (*Std. Dev*), skewness, and number of valid values (*Valid*). Skewness is a measure of symmetry in a distribution; a perfectly symmetric distribution would have a skewness value of 0, while distributions with long tails to the right (see *INCOME*) would have positive skewness, and vice versa. The *AGE* field has an observed range of 18 to 50 with a mean of 31.82, based on 4,117 records. Interest would be attracted by unexpectedly low or high values or odd distributions. For example, since this is a credit risk data file, an *AGE* value of 11 would suggest a data error. Similarly, a concentration of high-income values would suggest data errors or a sample not representative of the population at large (although this could be the target population).

Summary statistics are not calculated for flag, nominal, or ordinal fields; instead, the *Unique* column displays the number of unique categories found for the field. As expected, *GENDER* has two unique values and the distribution plot suggests the file has roughly equal numbers of males and females.

Examine the graphs and summaries for the other fields appearing in the Data Audit output window. Do the values look reasonable? Which field has missing data?

The graphs in the Data Audit output clearly display the general distribution of the fields, but are too small to present the scale and category identifiers. However, more detailed versions of the graphs can be readily produced.

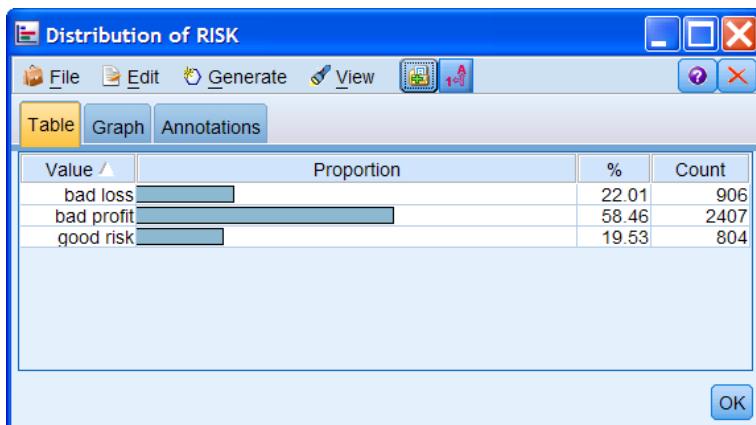
Distribution Plots

We can see distribution plots of a field's distribution directly from the Audit node output window.

Double-click on the **graph** for **RISK** in the Data Audit output window

Double-clicking on a graph in the Data Audit output window creates the graph (distribution plot or histogram) in its own window. For distribution plots, category labels appear along with count and percent summaries. This graph is added as a new object in the Outputs manager.

Figure 4.29 Distribution Table for Risk



The screenshot shows a Windows-style dialog box titled "Distribution of RISK". The title bar includes standard window controls (minimize, maximize, close). Below the title bar is a menu bar with "File", "Edit", "Generate", "View", and a toolbar with icons for "Table", "Graph", and "Annotations". The main area contains three tabs: "Table" (selected), "Graph", and "Annotations". The "Table" tab displays a distribution table:

Value	Proportion	%	Count
bad loss		22.01	906
bad profit		58.46	2407
good risk		19.53	804

At the bottom right of the dialog box is an "OK" button.

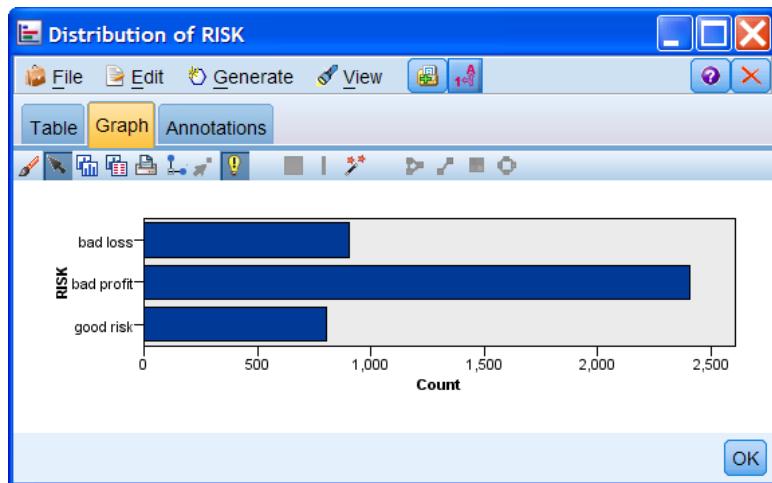
We examine the distribution of the field *RISK* that we are going to try to model in later lessons. This field contains three categories: *good risk*, *bad profit* and *bad loss*; these categories represent the view of a credit card company that an individual may be a good credit risk, a bad credit risk but be profitable, or a bad credit risk and cause a loss.

The largest group within the data contains 2407 individuals, or 58.46% of the data, and consists of those who are considered bad credit risks but profitable to the organization. The other two groups, bad loss and good risk, are roughly proportional, with 22.01% and 19.53% of the records, respectively.

If labels have been defined for the fields (variables) and value labels (or read from a Statistics data file), they can be displayed in the graph by clicking the *Display field and value labels* button .

The Graph tab in the Distribution window lets you view and edit just the bar chart for the displayed field.

Click **Graph** tab
Expand the window if necessary to view the full graph

Figure 4.30 Bar Chart for RISK

We can now view just the graph without the associated statistics. This graph can be edited to add a title, change the bar color, or the font characteristics for labels and text, among other things (see Lesson 7).

Once you have created a distribution table and graph, you can use options from the menus to group values, copy values, and generate a number of nodes for data preparation. In addition, you can copy or export the graph and table information for use in other applications.

Outside of the Data Audit node, a distribution plot for a single categorical field can be created from the Distribution node located in the Graphs palette.

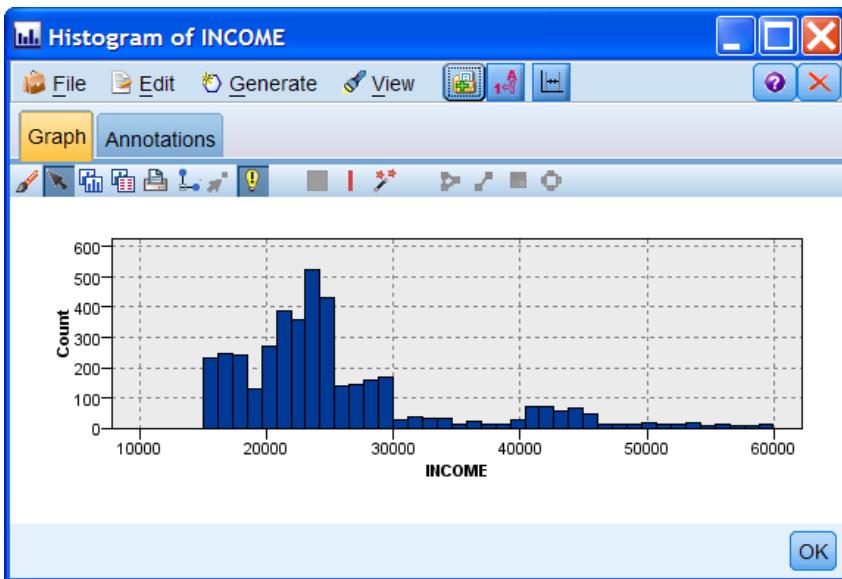
Click **File...Close** to close the Distribution graph window

Histograms

Now we'll look at a histogram for *INCOME*

Double-click on the **graph** for **INCOME** in the Data Audit output window
 Increase the size of the Histogram window

The Histogram shows the frequency of occurrence of values for numeric fields. In a histogram, the range of data values is split into bands (buckets) and bars representing the number of records falling into each band are displayed.

Figure 4.31 Histogram of INCOME

Income values range between approximately 15,000 and 60,000 with a large proportion of cases between 20,000 and 25,000. The distribution is also concentrated at the lower end of the income scale. The distribution looks generally as we would anticipate, as those with lower incomes normally are more frequent than those with higher incomes. This data exploration can be repeated for all numeric fields in the data.

When the Data Audit node generates histograms, the range displayed and binning of data values are determined automatically. You can control these properties for individual histograms by using the Histogram node and its Options tab.

Close the Histogram window, then close the Data Audit browser window

Notes

As we have seen, the Data Audit node automatically produces thumbnail distribution bar charts and histograms for all fields included in the analysis, which is a great convenience. However, you have greater control over chart options if the graph is individually produced using the Distribution or Histogram node (or the Graphboard node—see Lesson 7).

Summary

In this lesson you have been given an introduction to a number of methods that can be used to explore the quality of your data.

You should now be able to:

- Use the Data Audit node to assess the amount and type of missing data
- Select valid or invalid records using the Data Audit node browser
- Use the Types tab to define user-missing values or check for invalid values
- Visualize the distribution of fields and examine their summary statistics using the Data Audit node
- Examine Data Audit distribution bar charts and histograms in more detail

4.7 Appendix: Advice on Handling Missing Values

Some modeling techniques within PASW Modeler handle missing data better than others, specifically C5.0, C&R Tree, CHAID, QUEST, Decision List, Apriori, and Carma. However, others (for example Neural Net) use substitution rules for missing values (similar to what is done under the Coerce option; see the modeling course guides or the *PASW Modeler Algorithms Guide* for details) and the presence of missing values within a data set can have the effect of increasing the complexity of the data and the modeling process.

Eliminating missing values is often desirable but it must be kept in mind that such values can also be good indicators in the modeling process. For example, refusing to state income could be a factor in determining how likely a customer is to buy a particular product.

Standard methods to handle missing values include these:

- Omit records with missing data using Select nodes
- Omit fields with excessive missing data using Filter nodes
- Use auto-checking to coerce or discard illegal or missing values
- Impute (estimate) missing values using the Data Audit node, or modeling nodes

When considering which technique is appropriate it is important to keep in mind the following three situations.

Records with a Large Proportion of Missing Fields. If a small number of records contain a large proportion of missing fields, it is usually advisable to remove these records from the data. This can be achieved using either Select nodes or the Discard setting in the Check column of the Type node.

Fields with a Large Proportion of Missing Values. If a field contains a large proportion of missing values, the information that can be gained from the field is limited. In the majority of instances the field should be removed using a Filter node. In some cases the lack of data can provide information and the blanks can be filled with appropriate values.

Fields with a Small Number of Missing Values. If a field contains a small proportion of missing values it is not realistic to completely remove this field from the data mining process. A more sensible solution is to replace the blanks with legal values. There are a number of options available when replacing blanks with values.

Exercises

In this exercise, we will first use the Data Audit node to assess a smaller version of the charity data in which we have introduced some missing data. We will then use the stream created in the previous exercise and the complete charity data file to explore in more detail using the Data Audit node.

1. First, we first read the *CharityMissing.sav* Statistics data file. We have inserted some missing data in this file. Start a new stream; select the Statistics File node from the Sources palette and place it on the canvas. Check the measurement level and make changes if necessary as you did in the Lesson 3 exercises.
2. Add a Table node and connect the Statistics File node to the Table node. Then Run the Table node. Review the output table for null and missing data values. In which fields do you see \$null\$ values? Also, note that a couple of cases have the value 9 for SEX.
3. Add a Data Audit node and connect it to the Statistics File node; run the Data Audit node. Review the Quality table. Why are the null values reported as blank values for YOB, but not for the other fields (if you read the Statistics data file)? Note the range of values reported for SEX in the Audit table.
4. Define the value 9 as a blank (missing) value for SEX. (Hint: Use the Types tab in the Statistics File node.) Rerun the Data Audit node and observe the difference in the Quality table.
5. Clear this stream using Edit...Clear Stream.

Next, we will use Data Audit node to explore the complete charity data file.

6. Load the stream you created and saved in the last exercise, *ExerLesson3.str*, using the File...Open Stream menu choice. Or use the backup file *Backup_ExerLesson03.str*.
7. Edit the Types tab in the source node and click Read Values to force instantiation. Check that you agree with the chosen measurement levels. Change the measurement level if necessary. Which fields have Blanks defined as missing?
8. Attach a Data Audit node to the source node and examine the results for odd distributions and values. Give extra attention to fields related to pre- and post-campaign expenditure and visits.
9. Select one of the categorical (flag, nominal, ordinal) fields and examine its distribution in more detail (double-click on the graph in the Data Audit output window).
10. Select one of the continuous fields and examine its distribution in more detail.
11. Save an updated copy of the stream named *ExerLesson4.str*.

Lesson 5: Outliers and Anomalous Data

Objectives

- Use the Data Audit node to search for unusual values on categorical fields
- Use the Data Audit node to search for anomalies and outliers on continuous fields
- Use the Plot node to search for anomalies on two fields
- Use the Anomalies node to search for anomalies on many fields at once
- Provide advice on handling outliers and anomalous values

Data

We use the PASW Statistics data file *customer_offers.sav*, which contains customer data from a telecommunications company. It includes demographic data, information on the use of telecommunications services, lifestyle information, and response to three offers from the firm.

5.1 Introduction

In any data file, there may be values of one or more fields that can be considered to be unusual, odd, or extreme. Income values that are very high, or very low, total sales values that are very far from the mean or median, or telephone usage values that are close to zero minutes per month can all be viewed as anomalous or atypical. In and of themselves, unusual data values are not necessarily a cause for concern, and they are not necessarily erroneous. But, anomalous data can be a problem for models in small data files, they can be an indication of something interesting (such as a special subset of customers) to pursue further, or they can be errors in data coding.

Thus, although data mining normally uses very large data files, with many thousands or even millions of records, outliers and unusual data should be explored when preparing data for modeling.

This lesson will provide examples of several techniques to identify such data, including summary statistics, graphical techniques, and the Anomaly node. What to do with outliers and unusual values will also be discussed.

5.2 What is Anomalous Data?

What do we mean when we say that a data value is anomalous, or that it is an outlier? There are some statistically based definitions of such terms, but the general sense is that the data value is not like other data values for that field because it is either:

1. Far from the center of the distribution, measured by the mean or median and using the standard deviation as a measure of spread;
2. Far from other values, whether close to the center of the distribution, or not.

Of course, just because a value meets one of these criteria doesn't imply that it is a problem or shouldn't be used in modeling. For example, in a sample from the general public, the number of people aged 85 and above will be much smaller than those aged 35 or below, but this doesn't mean that there is something odd about people of that age. Due to the normal lifecycle, the proportion of the population who are very old decreases rapidly beyond about ages 80 to 85. If we are using age in a model, we wouldn't *a priori* consider people aged 85 and above to be outliers.

However, whether we include them or not in a model is a different proposition. Because of their generally reduced buying power, as most are living on retirement and social security or the equivalent, they may not be our target market for a product. Thus, we may set a cut-off age value and only use customers below that age to develop models. In that case, older customers are not classified as anomalous or outliers.

In comparison, suppose that we work for a telecommunications firm that sells wireless and landline services to customers. Some customers may use many, many minutes of various services each month, several orders of magnitude beyond the typical customer. These customers are potentially quite valuable to the firm, so it would be good to identify them and study them further. They are truly outliers and anomalies, but valuable ones.

If we are to develop models to predict whether a typical customer renews her service, though, including these unusual customers may not be the best strategy. Their large values may distort models developed with classical statistical techniques, such as regression-based models or discriminant analysis. Outliers can affect even neural networks.

Anomalies can also be identified by their values on two or more fields. For example, a customer could use many minutes of long distance service but no local minutes. There may be relatively few customers with such a pattern, so these customers are anomalous in the joint distribution of the two fields. Finding such patterns can be tedious, although rewarding, work, which is why data preparation is allocated such a large portion of the total project time in the CRISP-DM methodology.

The ultimate in anomaly detection is to use many fields to identify odd records. The Anomalies node provides this capability by doing a cluster analysis and then isolating records that are far from the cluster centers, or norms.

Categorical fields do not typically have outliers or anomalous values. For example, a customer's marital status, the state/province in which she lives, the type of credit cards she owns, or the type of dwelling in which she lives, will not be anomalous. Many other customers will have the same values on these fields. However, it is true that in a small data file, perhaps only a few customers will live in the same province or state. This may mean that we don't want to use the state/province field directly in modeling without recoding/reclassifying it into larger regional groupings. Otherwise, there will be only a few records in some categories, and this can lead to model instability. Categories with only a few records can be identified with a Distribution graph and table.

Classic anomalies and outliers are more likely to be found in continuous fields. Here, many different data values are possible, and even in a large data file, many values can be unique, with only one record having that value for a field. The potential problem with anomalies in scale fields is that they may distort the model by affecting parameters (such as the weights in a neural network). It is also possible that a record can be an outlier in the joint distribution of two fields. For example, a person might have a low income but have large debt. While this is certainly possible, customers of this type might require additional scrutiny, and they are definitely anomalous.

Finally, in some instances, outliers can be errors in data. In general, errors can only be found when the data value is so anomalous that it is very unlikely or impossible (such as having a million minutes of long distance time in one month).

5.3 Outliers in Categorical Fields

We'll begin our examination of outliers by using the Data Audit node to look at categorical fields.

If the Stream Canvas is not empty, click **File...New Stream**

Place a **Statistics File** node on the Stream Canvas

Edit the node and set the file to **customer_offers.sav** in the c:\Train\ModelerIntro directory

Click **Read labels as data** in the Values: area

Click **OK** to return to the Stream Canvas

Add a **Type** node to the stream

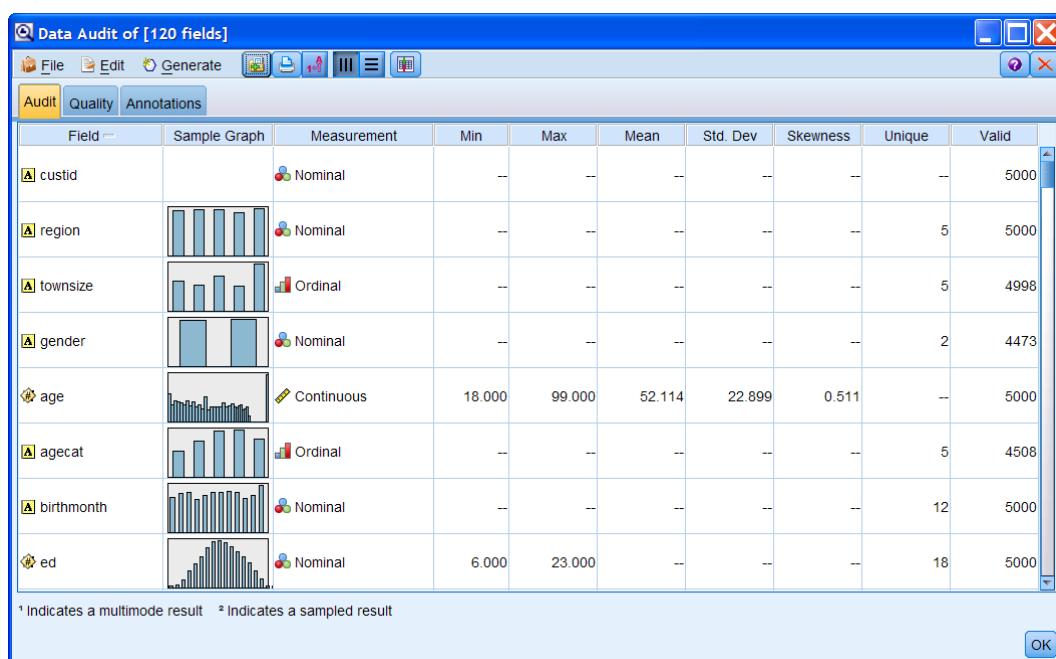
Connect the **Statistics File** node to a **Type** node

Add a **Data Audit** node to the stream and connect it to the **Type** node

Run the **Data Audit** node

Notice in the Data Audit window Audit tab that flag, nominal, and ordinal fields have no statistics except for the number of Unique values and Valid cases.

Figure 5.1 Data Audit Window Statistics



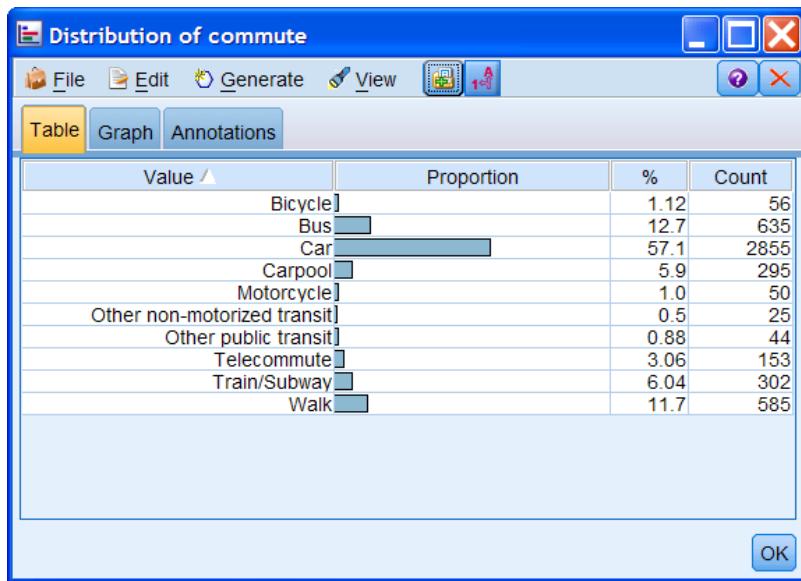
Tip

If you wish, you can click on the Label button to see the variable labels for the fields. You can also sort this table and the Quality table on field names by clicking on the Field header. This often makes it easier to locate the field of interest.

One field, *commute*, records the type of transportation used to commute to work. Let's examine that field.

Scroll down to the **commute** field

Double-click on the **Sample Graph** for **commute**

Figure 5.2 Distribution of Commute to Work

As we would expect, over half of the customers commute to work by car. None of the other categories are close in percentage to that, with bus being the second most frequent commuting method. Some methods of commuting, even in this data file of 5,000 customers, are mentioned infrequently, including motorcycle, bicycle, and the two “other” categories. There may not be enough records in these categories to use them in analysis and/or modeling.

Riding a bike or motorcycle to work is odd behavior *statistically* because so few people do so (it may be odd behavior otherwise, such as riding a bike to work in northern Alaska, but that is a different matter). But these categories aren’t an anomaly in the same way that earning a million dollars a year is for income: that value is an outlier because it so large. Riding a bike is not “far away” from other categories, since commuting method is a nominal field. Instead, riding a bike to work is simply infrequent behavior.

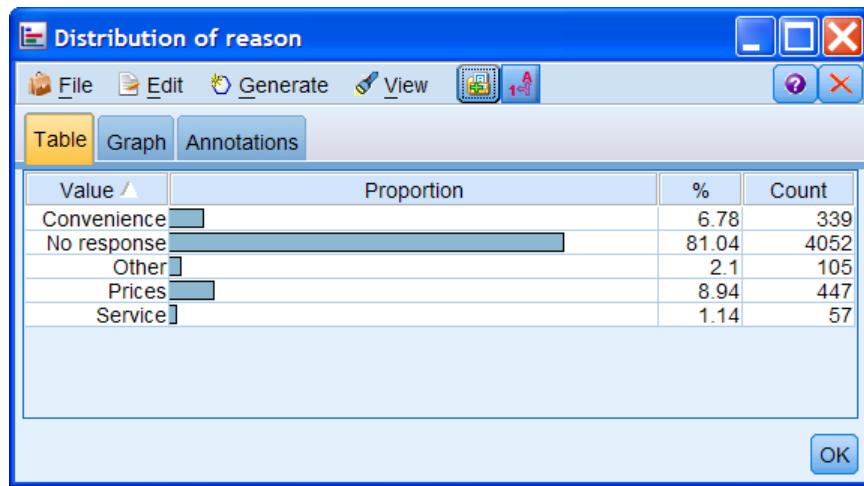
Adjusting for this type of outlier means either:

1. Grouping categories together so that there are fewer categories with a small number of records. Thus, we could group bike and motorcycle since they are similar, i.e., two wheels, not-enclosed, not public transportation. Or we could group all the small categories into a larger “Other” group.
2. Remove these categories from the stream. We might decide that they are so infrequent that we can ignore them for the bulk of customers.

We’ll examine one more categorical field.

Close the Distribution of commute window

From the Data Audit output window, double-click on the **Sample Graph for reason**

Figure 5.3 Distribution of Primary Reason for Being a Customer

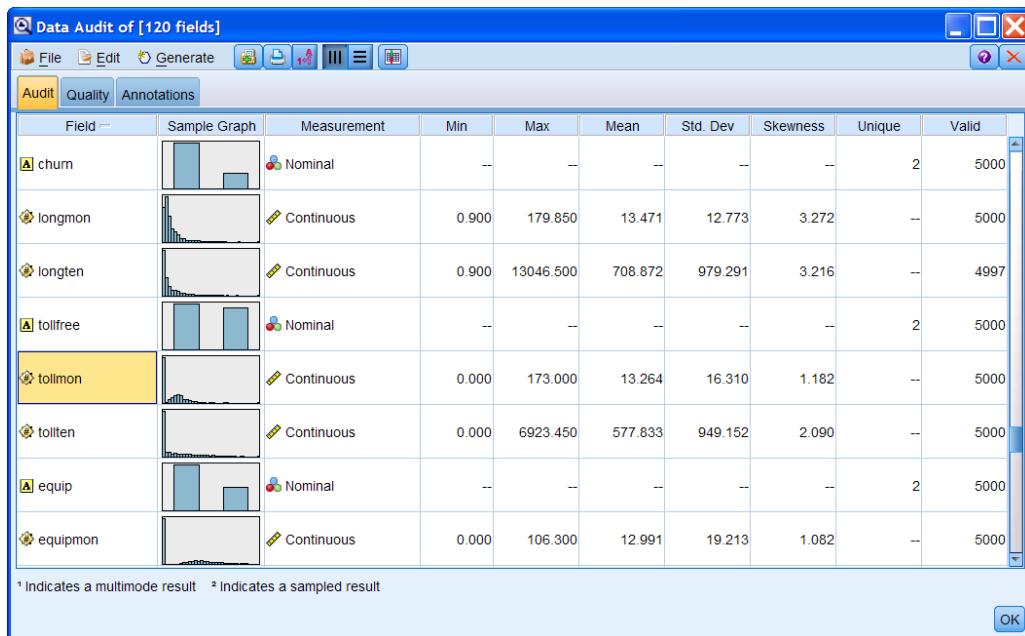
The field *reason* records the answer to the question about the primary reason someone is a customer of the telecommunications firm. The three proffered choices were convenience, prices, and service. However, over 81% of the customers have no response on this question. A response that occurs that frequently can't be an outlier, but it should make us wonder about why there are so many non-responses. Is it possibly true that so many customers refused to answer this question, or don't have an opinion? Or is it more likely that, for some currently unknown reason, only a subset of customers were asked this question?

The reason(s) for the non-response (which can be viewed as missing data, per our discussion in Lesson 4), has everything to do with how you treat it. If the non-response is equivalent to “don’t know,” then you might want to use it in modeling, on the supposition that customers who don’t know the primary reason they became customers may act differently than others. But if the non-response was caused by not being asked the question, then it is truly missing data, and this field should not be used in modeling because there is too much missing data.

5.4 Outliers in Continuous Fields

We next turn our attention to outliers and anomalous data in continuous fields. There are many fields of this type in the customer data and we focus on only a few. The Data Audit node provides summary statistics on continuous fields by default. We'll begin by examining the statistics for *tollmon*, which measures the number of toll-free minutes used last month.

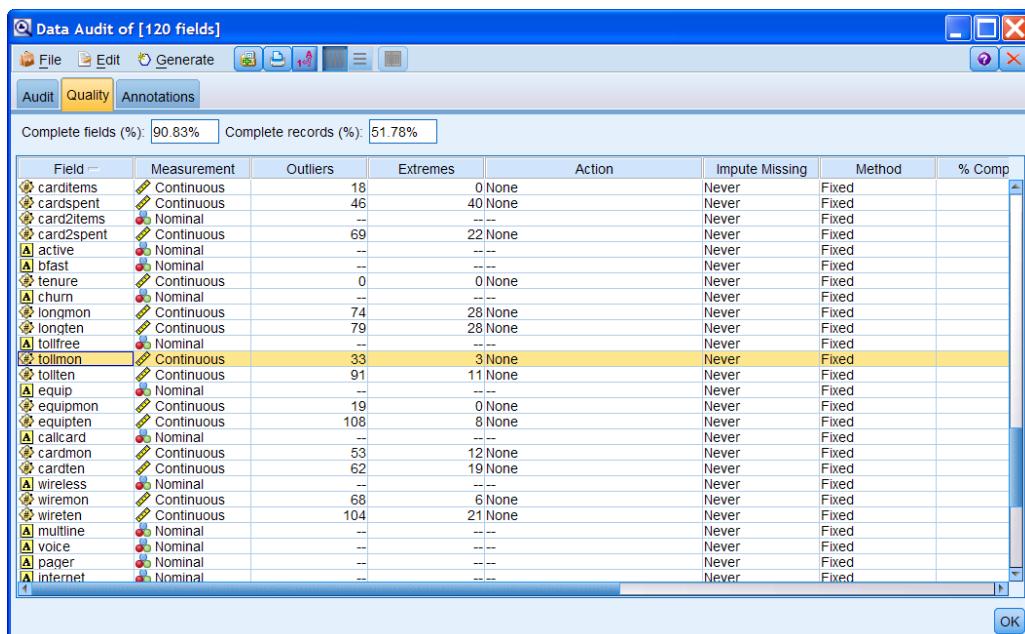
Close the Distribution of reason window
In the Data Audit output window, scroll down until you see the field **tollmon**

Figure 5.4 Summary Statistics for Tollmon

The minimum value for *tollmon* is 0, and the maximum is 173. This doesn't seem too great a range, but the mean is about 13.26 with a standard deviation of 16.31. This implies that customers who used over 100 minutes are quite statistically unusual because they are several standard deviations above the mean.

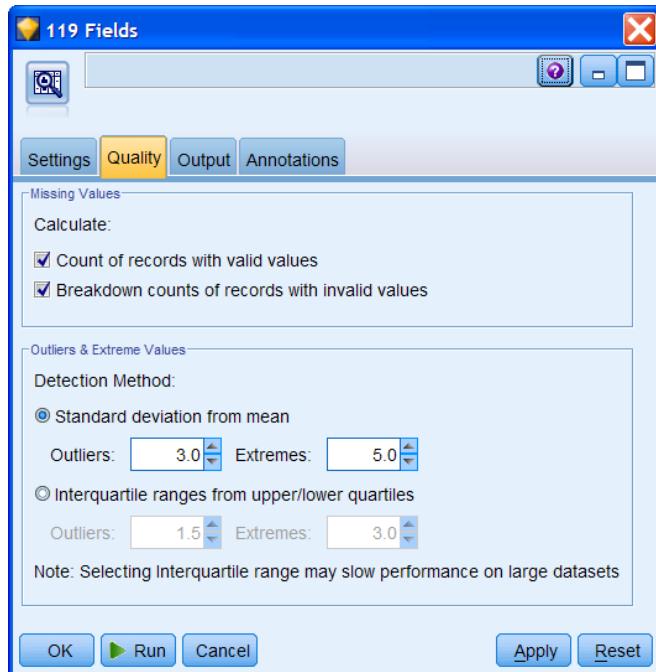
Before we look at the histogram for *tollmon*, we can view the number of outliers and extreme values.

Click the **Quality** tab
Scroll down to **tollmon**

Figure 5.5 Outliers and Extreme Values for Tollmon

There are 33 outliers and 3 extreme records for *tollmon*. The number in each category is determined by settings in the Data Audit node Quality tab, displayed for easy reference in Figure 5.6.

Figure 5.6 Data Audit Node Settings for Outliers and Extreme Values



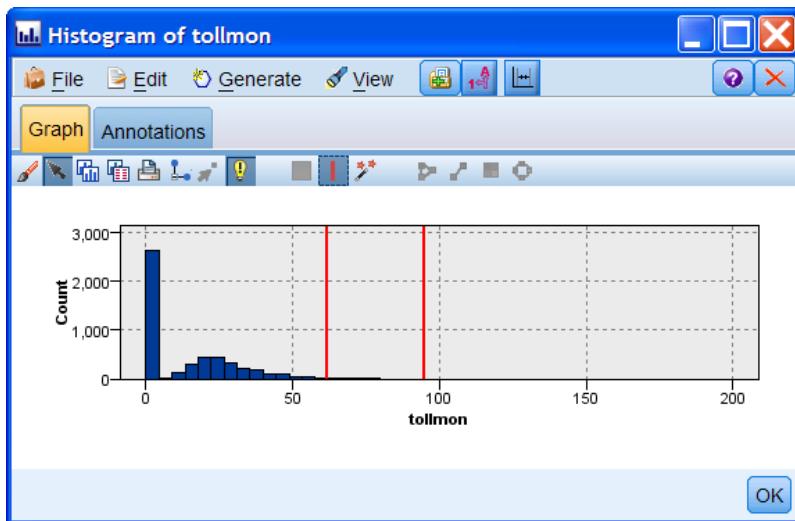
In the Outliers & Extreme Values area, Outliers are defined as values 3 to 5 standard deviations from the mean, and extreme values are 5 or more standard deviations from the mean. These are reasonable defaults, but you can certainly modify them as appropriate. Unless you have hundreds of fields to review, you should normally review a histogram for a field to supplement the statistical information.

The total of 36 records that are outliers or extreme is less than 1 % of the file. These values would be of little concern when constructing models with decision trees where *tollmon* is an independent or predictor field, but for models based on statistical theory (e.g., linear regression) even 36 cases are enough to affect model performance.

It is sometimes a difficult judgment as to when you should be concerned about outliers, even after discovering their existence. A handful of outliers are of no concern, but when they reach a few percent of the file size, they can be, especially if there are many extreme values. If there is time, we recommend creating a copy of a field and modifying the outliers in it, then creating models with and without the modified field to see the effect.

Before considering what to do with these outliers, we need to look at the histogram for *tollmon*.

**Click on the Audit tab
Double-click on the Sample Graph for tollmon**

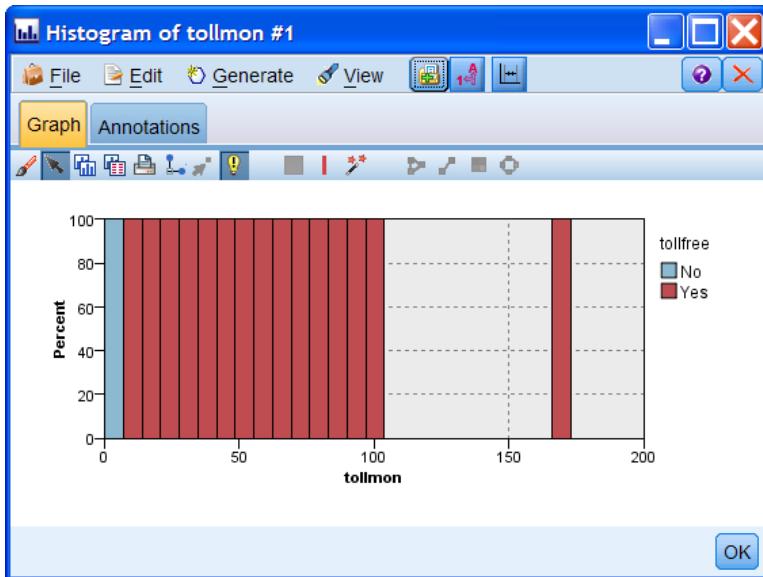
Figure 5.7 Histogram of Tollmon with Limits for Outliers and Extreme Values Added

On the histogram in Figure 5.7 we have marked the approximate cut points for outliers and extreme values. The outlier value is 3 standard deviations from the mean, so approximately 62 ($13.26 + 3 \times 16.31$) and similarly the extreme value, approximately 95, is 5 standard deviations from the mean ($13.26 + 5 \times 16.31$). This is a useful technique to see where they are located in the distribution. Certainly, given the shape of the distribution, those values do look extreme or anomalous.

There are two other key features of the distribution. Over half of the customers have a value of 0. These are either customers who don't make toll calls, or who don't have toll service. As with the field *reason*, a value shared by over half of the file cannot be viewed as anomalous, but we have to understand what underlies it. Second, the distribution of values from just greater than 1 to about 50 minutes is approximately normal, which is suitable for any type of model. But we have the problem of the extreme values and the many records with values of 0.

We'll use a Distribution node with *tollmon* overlaid with *tollfree* (whether a customer has toll free service or not) to explore this further.

- Close the Histogram window
- Add a **Histogram** node to the stream and connect it to the **Type** node
- Edit the **Histogram** node
- Select **tollmon** as the Field
- Select **tollfree** as the Color Overlay
- Click **Options** tab
- Select **Normalize by color** (not shown)
- Click **Run**

Figure 5.8 Histogram of Tollmon with Tollfree Overlaid

All customers *without* toll free service have values of 0 for *tollmon*, so now we understand where all the zero values originated. All customers *with* toll free service have at least a few minutes of toll free calls.

The question then becomes how to handle all the zeros for *tollmon*. The true coding for these customers is actually “not applicable” since these people can’t, by definition, have any toll free minutes. In other words, they should be coded as missing on *tollmon*.

However, if we do so, this raises a new issue. As we discussed in Chapter 4, some modeling procedures in PASW Modeler will not use missing data, and others will handle it in ways that might not be desirable. So just making the value of 0 a blank (missing) for *tollmon* is not necessarily a solution. And we could potentially lose these customers for modeling when they are coded as missing. Additionally, we may not want models to combine the value of 0 with other valid values. Given all this, here are possible solutions:

1. Make no changes to *tollmon*, but include the field *tollfree* in any model. This field identifies those without toll free service and so “controls” for those with a value of 0 on *tollmon*.
2. Group or reclassify *tollmon* into fewer categories, with one of those categories being everyone with 0 minutes. Then be sure to use a modeling technique that keeps that category distinct. You can also use *tollfree* as well, per solution #1.
3. Follow solution #2, but don’t worry about whether those with 0 minutes of toll free service are grouped with customers with only a few minutes. Perhaps, in terms of predictive modeling, there is no real distinction between these two categories.

We are certain you can think of other ways to proceed, but these three provide examples of how to think about fields like *tollmon* and how to incorporate them in models.

Note

If you wish to see the equivalent of a Distribution graph for a continuous field (such as *tollmon*), so that you can see more detail by viewing a bar for each value, you must change the measurement level of the field. You can edit the Type node or Source node, Types tab, and change the measurement

level to ordinal. Then have PASW Modeler reinstantiate the data, either directly in the Type node or by running data through the Type node. After that, the field will be a categorical field, and you can use it in a Distribution graph. For convenience, you may wish to make a copy of the Type node that you modify and retain the original one.

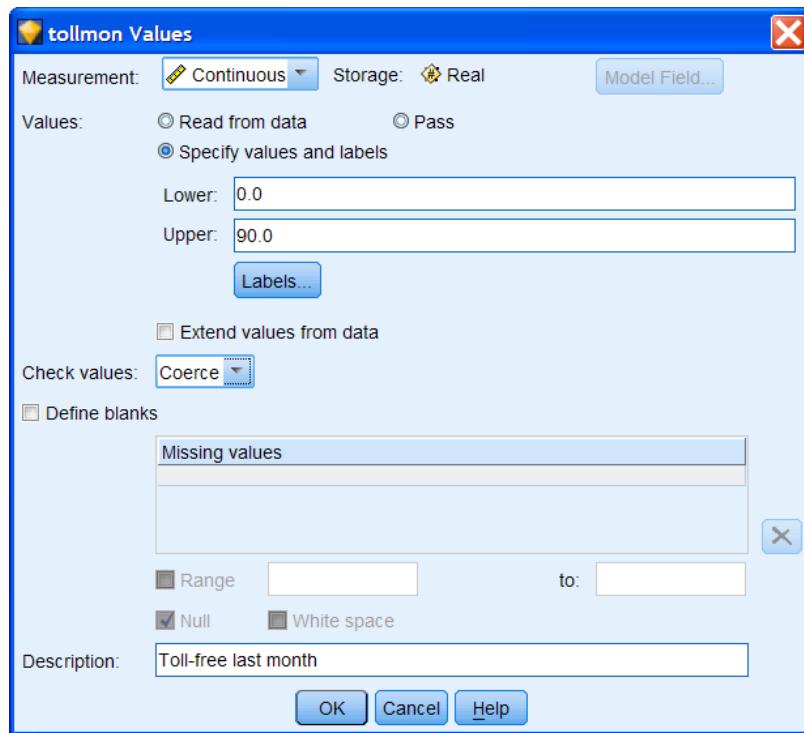
We must return to the issue of outliers on *tollmon*, as we took a detour to examine the customers with a value of 0. Recall that there were 36 cases with outlier or extreme values, and that extreme values began around 95 minutes. Here are several methods for handling very large, or very small, outliers:

1. Delete them from the file: This will certainly work, and if their numbers are very small, this might be an acceptable strategy, if we presume that outlier and anomalous values are of no special interest.
2. Change their values to something less extreme. As we learned in Lesson 4, the Type node has a feature that will “coerce” values to upper and lower limits for a field. The disadvantage of this approach is that all the values for these cases will be the same, causing a pile-up of records at one value.
3. Change their values to something less extreme using a method that doesn’t code them all to the same value. You could give all the outliers one new value, and the extreme records another. Or you could add some variability to the coding by using a random number function to add some random “error” when assigning the new value.

Again, there are more ways to proceed than these three, and you may often wish to try models with and without the modified fields to see what the effect is of making the anomalous data values less extreme.

In our case, we’ll try the first strategy by assigning all outliers and extreme values the new value of 90 minutes for *tollmon*. This value is a reasonable compromise that is large but not overly so.

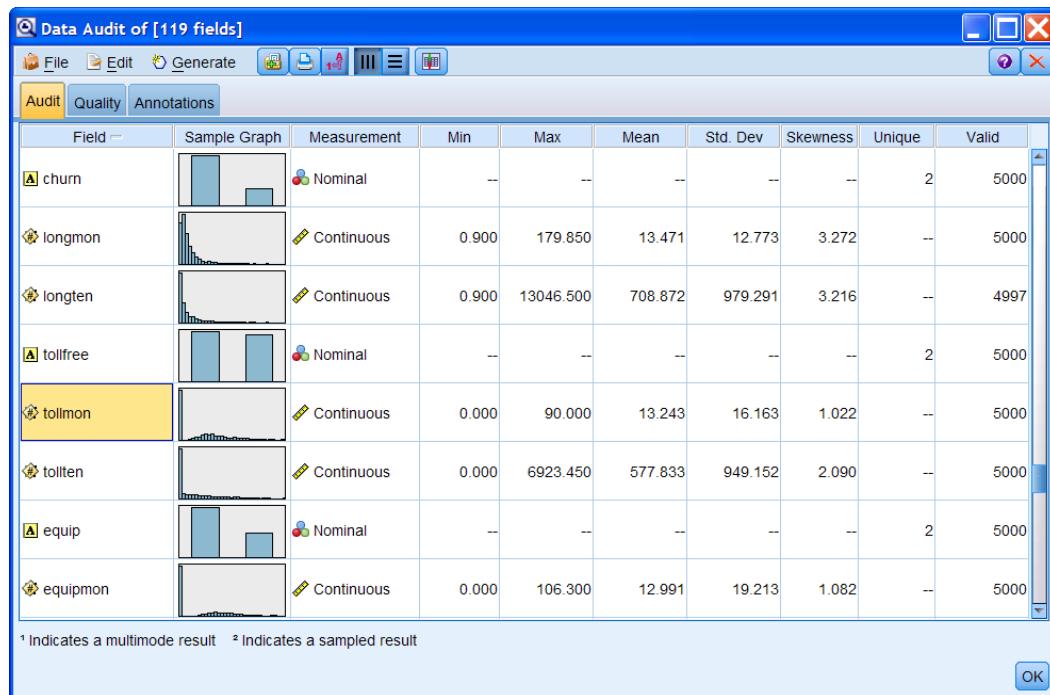
Close the Histogram window
Close the Data Audit window
Edit the **Type** node
Click in the **Values** cell for tollmon and select **Specify**
Change the **Upper** value to **90.0**
Click the dropdown list for Check values: and select **Coerce**

Figure 5.9 Specifying Desired Range for Tollmon

Click **OK**, then click **OK** again

Now we can rerun the Data Audit node.

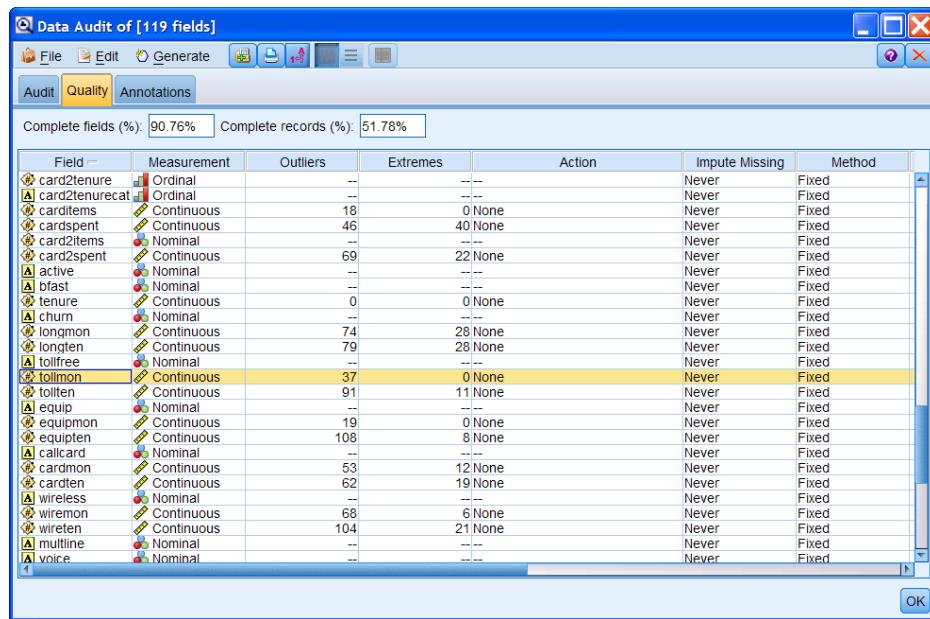
Run the **Data Audit** node
Scroll to **tollmon**

Figure 5.10 Statistical Summaries for Tollmon

The maximum value of *tollmon* is now 90, per our specifications. The mean is 13.24; it was 13.26 previously. The standard deviation is now 16.16; before it was 16.31. Thus, we have changed the overall properties of *tollmon* very little, yet have reduced the magnitude of outliers.

Let's see how many outliers and extreme values now exist for *tollmon*.

Click the **Quality** tab
Scroll to **tollmon**

Figure 5.11 Outliers and Extreme Values for Modified Tollmon


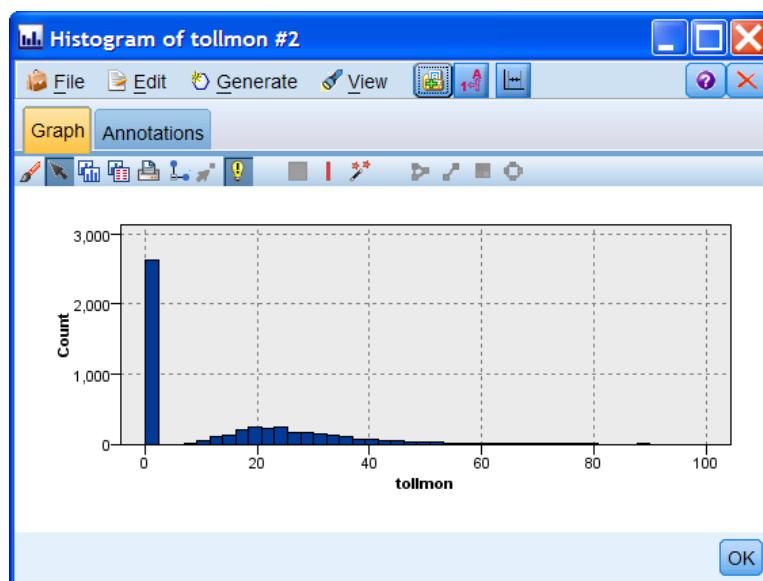
The screenshot shows the Data Audit node interface with the 'Audit' tab selected. The title bar indicates 'Data Audit of [119 fields]'. Below the title bar, there are three tabs: 'Audit' (selected), 'Quality', and 'Annotations'. A status bar at the bottom shows 'Complete fields (%): 90.76%' and 'Complete records (%): 51.78%'. The main area is a table with the following columns: Field, Measurement, Outliers, Extremes, Action, Impute Missing, and Method. The table lists 119 fields, including 'tollmon' which has 37 outliers and 0 extremes. The 'Method' column for 'tollmon' is set to 'Fixed'.

Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method
card2tenure	Ordinal	--	--	Never	Fixed	
card2tenurecat	Ordinal	--	--	Never	Fixed	
carditems	Continuous	18	0 None	Never	Fixed	
cardspent	Continuous	46	40 None	Never	Fixed	
card2Items	Nominal	--	--	Never	Fixed	
card2spent	Continuous	69	22 None	Never	Fixed	
active	Nominal	--	--	Never	Fixed	
bfast	Nominal	--	--	Never	Fixed	
tenure	Continuous	0	0 None	Never	Fixed	
churn	Nominal	--	--	Never	Fixed	
longmon	Continuous	74	28 None	Never	Fixed	
longten	Continuous	79	28 None	Never	Fixed	
tolfree	Nominal	--	--	Never	Fixed	
tollmon	Continuous	37	0 None	Never	Fixed	
tollten	Continuous	91	11 None	Never	Fixed	
equip	Nominal	--	--	Never	Fixed	
equipmon	Continuous	19	0 None	Never	Fixed	
equipten	Continuous	108	8 None	Never	Fixed	
callcard	Nominal	--	--	Never	Fixed	
cardmon	Continuous	53	12 None	Never	Fixed	
cardten	Continuous	62	19 None	Never	Fixed	
wireless	Nominal	--	--	Never	Fixed	
wiremon	Continuous	68	6 None	Never	Fixed	
wireten	Continuous	104	21 None	Never	Fixed	
multline	Nominal	--	--	Never	Fixed	
voice	Nominal	--	--	Never	Fixed	

The Data Audit node reports that there are 37 outliers and no extreme values. At first glance this seems peculiar, since we coerced the anomalous values to reduce the outliers. But, the value of 90 we used for coercion is just below the threshold for extreme values in the original field (94), so what we actually did was reduce the extreme values, not the outliers. Obviously, we could choose a different value and we would get a different number of outliers and extreme records.

What is the distribution of the modified *tollmon*?

Click the **Audit** tab
Double-click the **Sample Graph** for *tollmon*

Figure 5.12 Histogram of Modified Tollmon

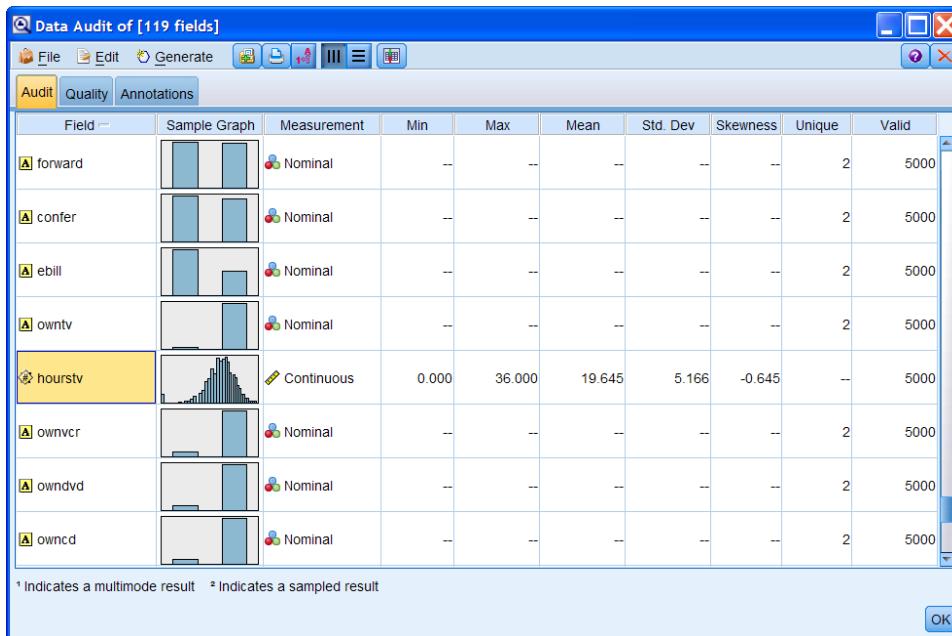
If we temporarily ignore the many customers with a value of 0, the distribution of cases with 1 or more toll free minutes of service looks more symmetrical and less extreme, and thus better suited for modeling.

Close the Histogram window

We can briefly review the distribution of one more continuous field, *hourstv*, which records the number of hours a customer spent watching TV last week.

Scroll to the field **hourstv**

Figure 5.13 Summary Statistics for Hourstv



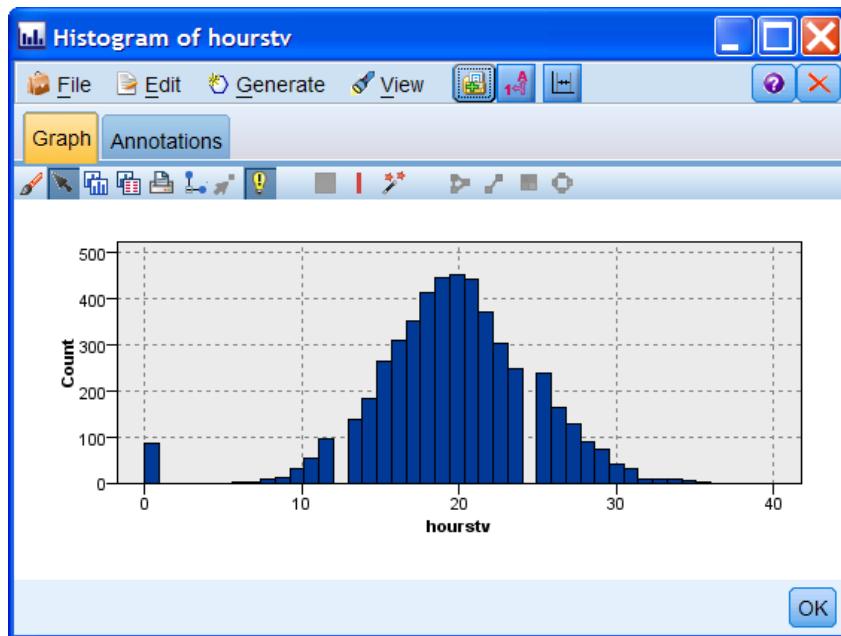
The range of values is from 0 to 36 hours, and the mean is right in the middle at about 19.6. With a standard deviation of about 5.2 there aren't likely to be many outliers, it would seem.

Click on the **Quality** tab (not shown)

However, there are 88 outliers listed for *hourstv* in the Quality tab output. To explore this we need to look at the histogram for *hourstv*.

Click on the **Audit** tab

Double-click on the Sample Graph for **hourstv**

Figure 5.14 Histogram for Hourstv

The distribution is normal, except for the records at a value of 0. These are the outliers identified by the Data Audit node. However, unlike with the situation for *tollmon*, values of 0 here are valid; they are simply customers who didn't watch any TV last week, either because they don't own a television or they were so busy that they didn't have time to watch TV. Whatever the cause, they are an anomaly compared to the bulk of other customers, and they are sufficiently far from the center of the distribution—over three standard deviations—that they could affect a model.

We could move them closer to the other values by recoding these records to a value of 4 or 5. They would still stand out from other cases, but not greatly so. However, some analysts may understandably balk at changing these valid values which contain valuable information that no television was watched last week. If so, another alternative is to create a flag field that indicates whether or not a person watched any television, and then include that flag field along with *hourstv* in any models. This is similar to one of the strategies we discussed for *tollmon*.

Tip

You can generate a Derive node to create the flag field directly from the Histogram.

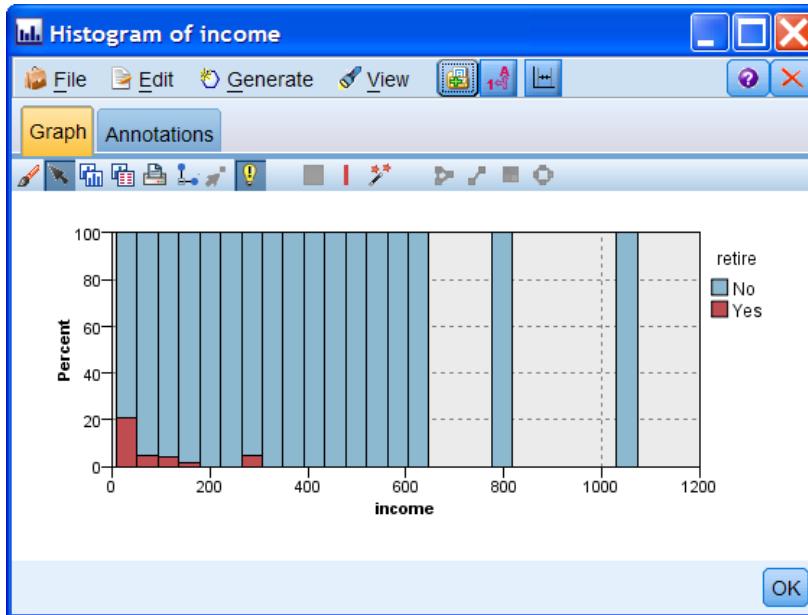
5.5 Outliers in Two Fields (Categorical and Continuous)

A record may not look unusual in its value on any one field, but clearly be anomalous in its joint distribution on two, or more, fields. It takes a bit more imagination, and work, to find this type of data anomaly because the Data Audit node looks at only one field at a time. But you can easily do so with Distribution and Plot nodes, or a Matrix node for fields that are both categorical. Typically you look for anomalies on two or more fields only after you have adjusted for anomalies on each individual field.

We'll begin by looking at the joint distribution of a categorical and continuous field, *retire* and *income*, respectively. The first field measures whether or not the customer is retired; the second field measures household income. We'll create a histogram of *income* overlaid by *retire*.

- Close the Histogram window
- Close the Data Audit output window
- Edit the existing **Histogram** node
- Click **Plot** tab
- Select **income** as the Field
- Select **retire** as the Color Overlay
- Click **Run**

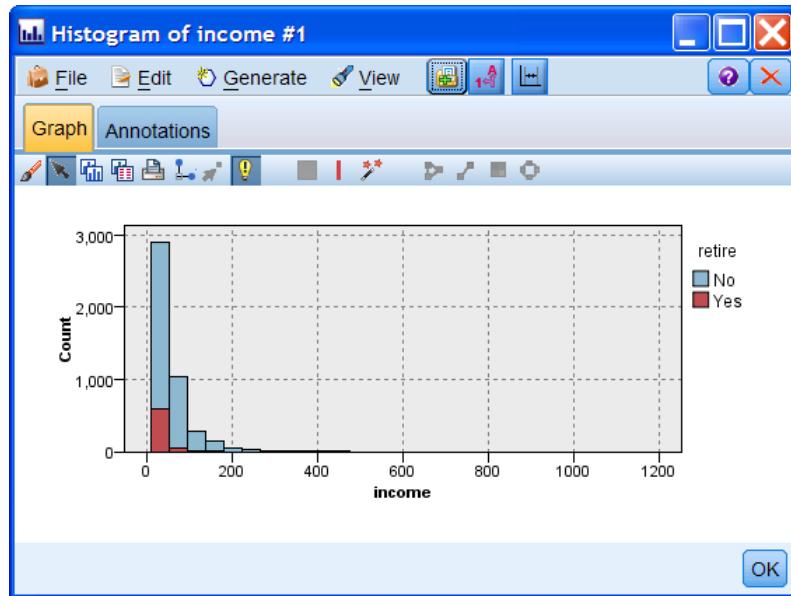
Figure 5.15 Histogram of Income with Retirement Status Overlay



We have discovered something unusual. Those customers who are retired generally have lower incomes than those who are still working. But there is a notable exception. There is a group of customers whose household income is well over 250K but who are retired. It is possible that the customer's spouse is still working. It is possible that the retired customer lives with a daughter or son who is earning a high salary. It is, though, also possible that there is something unusual about this group, or that these are data errors. After all, the other retired customers have an income distribution that falls off rapidly after about 50K.

Of course, the current histogram is normalized and displayed in percentages, not raw case counts. Let's rerun the Histogram and make this change.

- Close the Histogram window
- Edit the **Histogram** node
- Click on the **Options** tab
- Click on **Normalize by color** to deselect it
- Click **Run**

Figure 5.16 Histogram of Income with Retirement Status Overlay, Not Normalized

We can now see that there are very few customers with incomes over 250K, no matter whether they are retired or not. In fact, there are only 20 customers in the band of interest, and only 1 of them is retired (so that customer is 5% of the band; refer to Figure 5.15).

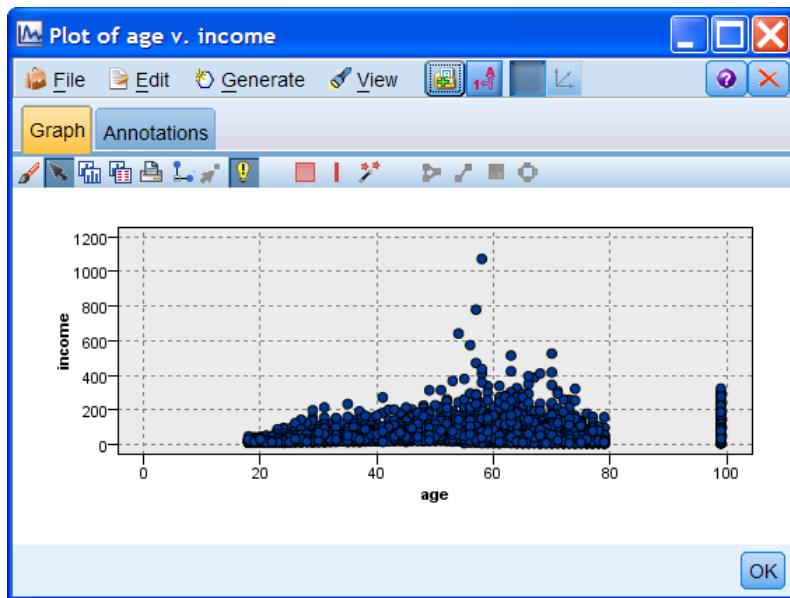
Fortunately, we don't need to be concerned with this one customer. One anomalous case in a file of 5,000 is tolerable. If, instead, there had been dozens of retired people with a high household income, we would need to do further exploration. As always, the first thought might be a data error. If that possibility can be ruled out, then we can think about whether we might need to modify the income values for some of the retirees. And, as mentioned at the start of this section, since *income* is highly skewed with many extreme values, we would normally have made adjustments to these values first before looking at the joint distribution of *income* with other fields.

5.6 Outliers in Two Continuous Fields

A case can be an outlier in the distribution of two continuous fields. In fact, as with outliers on single fields, an outlier on two continuous fields is the more standard situation. First, to demonstrate an instance where there are few, if any outliers, we will look at the relationship of age and income. We would expect that there should be a positive relationship between the two, so that income rises, on average, with age (or at least until retirement age). Since both these fields are continuous, we can use a Plot node. (A Collection node would also display the relationship between the two but wouldn't allow you to easily find anomalous records.)

- Close the Histogram window
- Add a **Plot** node to the stream and connect it to the **Type** node
- Edit the **Plot** node
- Select **age** as the **X field** and select **income** as the **Y field**
- Click **Options** tab
- Select **Use all data** in the When number of records greater than area
- Click **Run**

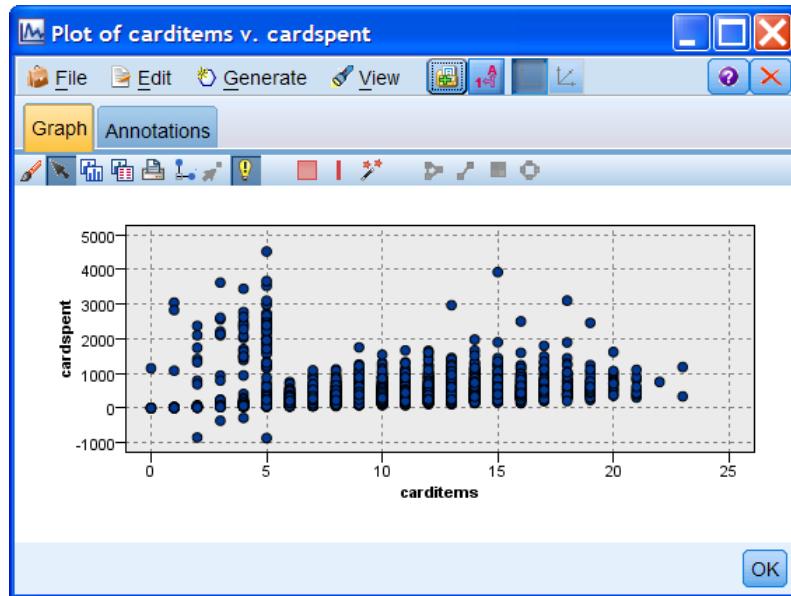
We need to check the option to use all the data, not just a sample, to be sure we find all the outliers. If we were working with a very large data file, though, we might instead take a (large) sample.

Figure 5.17 Scatterplot of Age and Income

We observe the expected relationship in the graph. The maximum value of income rises with age, but then declines after about age 70, as people retire. There are many customers with an age value of 99, which is missing for age, so these records are not anomalies (refer to Lesson 4). Although there are a few very high incomes, they occur at older ages. We don't have customers with very high incomes at very young ages. We do have customers with low incomes at older ages, but they are not anomalies, as many other customers fall into this general group. So in summary, this is a plot where there are no obvious anomalies.

Next we examine the relationship between the amount of money spent on a customer's primary credit card last month (*cardspent*) and the number of items/transactions (*carditems*).

- Close the Plot window
- Edit the **Plot** node
- Click the **Plot** tab
- Select **carditems** as the **X field**
- Select **cardspent** as the **Y field**
- Click **Run**

Figure 5.18 Scatterplot of Carditems and Cardspent

There is unquestionably something anomalous about many of the records with 5 or less items on their credit card last month. The general distribution between the two fields is that the amount spent tends to rise with the number of items, although there is an interesting possible decline between 15 and 20 items (perhaps for people who make many small purchases with their credit card). But there are many customers who have very large charges but have five or fewer items (and there appears to be an error for a customer with no items but a charge over 1,000 dollars).

Some of this group of customers are not anomalies on either field but are certainly anomalous when viewed in the context of both fields. Maybe many of these customers took several airline flights last month and so have large charges for just a few items. In that case, we might not consider them to be anomalies, although they might be of interest from a marketing perspective—frequent travelers may well be interested in more telecommunication services.

We can investigate them further by using the Interactive capabilities of the Plot window to draw a rectangle around these points, then generate a Derive or Select node that can be used in a stream to explore other characteristics of this group. There is no reason to modify the values of these customers on either *carditems* or *cardspent* at this point, unless we find data errors or other related issues. Instead, we may have identified an interesting group of customers who demand extra attention before we begin modeling.

5.7 The Anomaly Node

Records can be anomalies on several fields at once, but this is almost impossible to detect by hand. To assist in finding such records, PASW Modeler provides the Anomaly node. The Anomaly node is located in the Modeling palette.

Anomaly detection models identify outliers, or unusual cases, in the data by using clustering analysis. The Anomaly node creates a cluster model with clusters that identify “normal” cases, or peer groups, into which similar records fall. Each record can then be compared to others in its peer group to identify possible anomalies. The further away a record is from the cluster center, the more likely it is to be unusual.

Unlike other modeling methods that store rules about unusual cases, anomaly detection models store information on what normal behavior looks like. This makes it possible to identify outliers even if they do not conform to any known pattern, and it can be particularly useful in applications, such as fraud detection, where new patterns may constantly be emerging.

Each record is assigned an anomaly index, which is the ratio of the group deviation index to its average over the cluster that the case belongs to. The larger the value of this index, the more deviation the case has than the average. Cases with an index value greater than 2 could be good anomaly candidates because the deviation is at least twice the average, although you typically use a specific index value threshold based on file size and the percent of cases you wish to identify as anomalies.

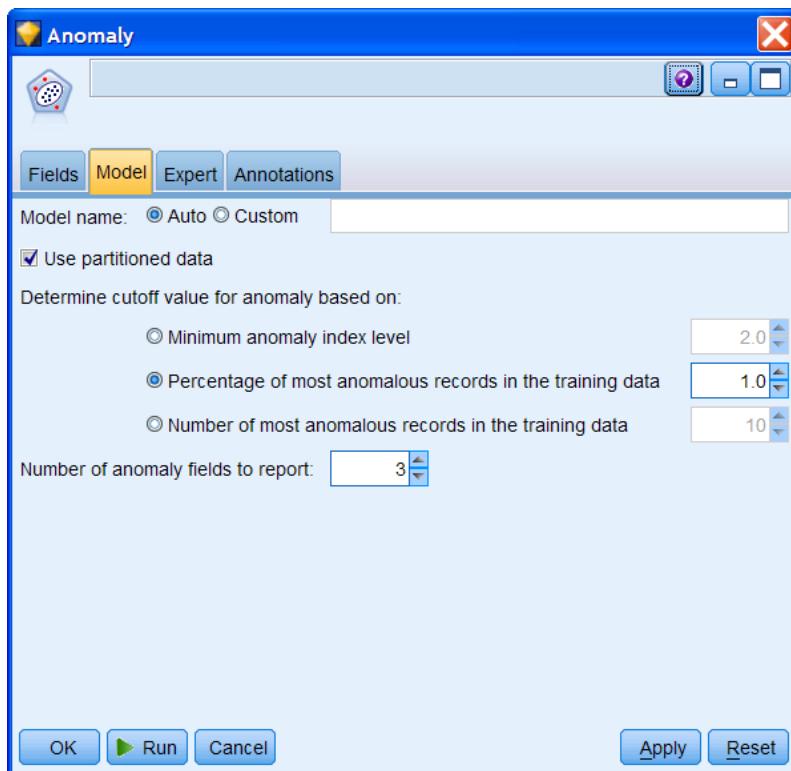
The Anomaly node uses all fields with Role Input and ignores target fields (Role In or Both). However, you usually don't want to use every field in the data file to look for anomalies. Typically you would only use those fields you expect to use in modeling.

As with the other, simpler methods of searching for anomalies that we have reviewed, just because a record is anomalous is no reason not to include it in a model. When anomalies are found (and in a large data file the Anomaly node will always find anomalous records), you need to do additional investigation to see what is unusual about these cases and what, if anything, to do about it.

In this example, we will use a subset of fields from the customer data to identify anomalous records.

- Close the Plot window
- Add an **Anomaly** node from the Modeling palette to the stream and connect it to the **Type** node
- Edit the **Anomaly** node

Figure 5.19 Anomaly Node Model Tab



The classification of anomalous records is based, by default, on a percentage of records in the (training) data. This value is used as a parameter in the model, but *not* the actual percentage of records to be flagged during scoring. Alternatively, you can use a minimum anomaly index value or an absolute number of records. We'll stay with the default.

We need to specify which fields to include.

Click the **Fields** tab

Click **Use custom settings**

Select all the contiguous fields from **region** to **bfast** (not shown)

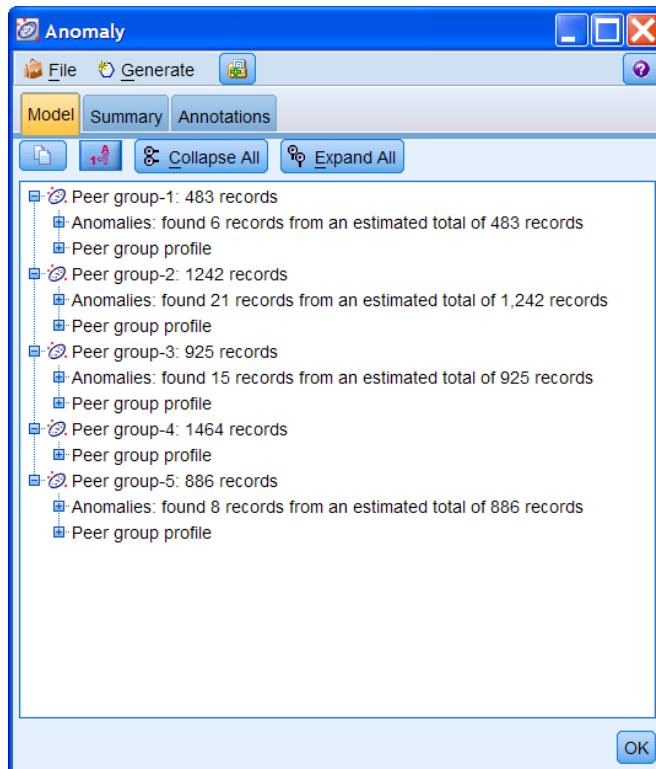
Click **Run**

The Anomaly node, like any modeling node, creates a model in the Models manager.

Right-click on the Anomaly model and select **Browse**

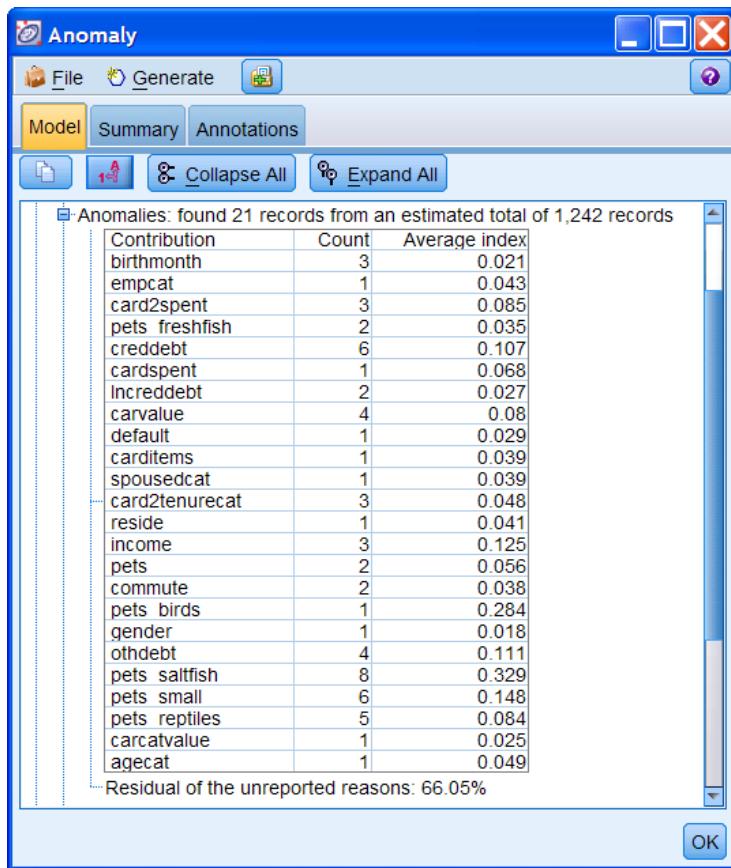
Click on the plus icon  in front of each of the **Peer Groups** (shown in Figure 5.20) so they are expanded

Figure 5.20 Anomaly Model Results



There are five clusters or peer groups with a total of 50 anomalous records identified; that is the 1% of the cases that we requested. All peer groups except group 4 have some records with anomalies. We'll explore group 2 further since it has the most records with anomalies.

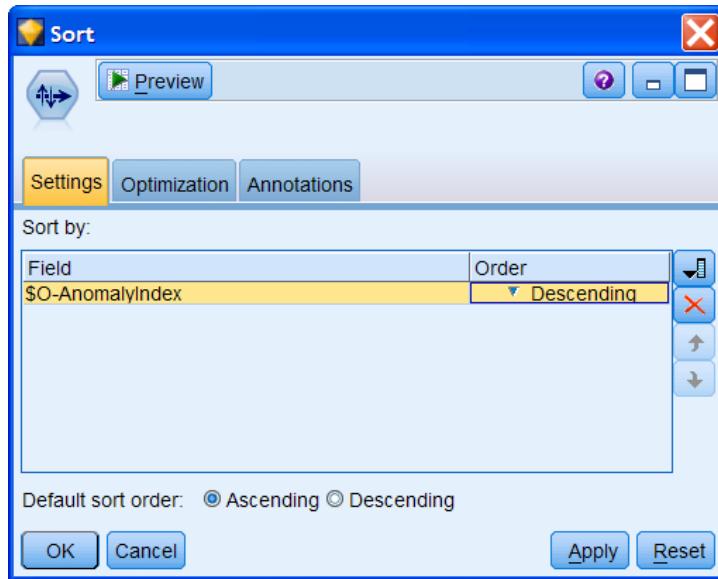
Click on the plus icon  in front of the line **Anomalies: found 21 records from an estimated total of 1242 records**

Figure 5.21 Anomalies Table for Peer Group 1

There were 21 anomalous records identified in peer group 2. This table lists the fields that contributed to a case being identified as an anomaly. The most frequently used fields were *pets_saltfish* (number of salt water fish owned), *pets_small* (number of small animals owned) and *creddebt*. (credit card debt in thousands). The highest Average Index values (all over .1) were for *pets_saltfish*, *pets_birds*, *pets_small*, *income*, *othdebt*, and *creddebt*.

Likewise, you could examine the characteristics of the anomalies for each of the peer groups. The Anomaly model doesn't identify which cases are anomalies. To see which records are anomalies we need to use the model nugget. You will find that the model nugget is connected to the Type node automatically when running the Anomaly node.

- Close the Anomaly model node window
- Add a **Sort** node to the stream from the Record Ops palette and connect the **Anomaly model** to the **Sort** node
- Edit the **Sort** node
- Select the field **\$O-AnomalyIndex** as the sort field
- Change the sort Order to **Descending**

Figure 5.22 Sorting Records by the Anomaly IndexClick **OK**Connect a **Table** node to the **Sort** nodeRun the **Table** node

Scroll to the last columns in the Table

Figure 5.23 Anomalous Records Sorted by Anomaly Index

	\$O-Anomaly	\$O-AnomalyIndex	\$O-PeerGroup	\$O-Field-1	\$O-FieldImpact-1	\$O-Field-2	\$O-FieldImpact-2	\$O-Field-3	\$O-FieldImpact-3
1	T	3.812	5	creddebt	0.366	income	0.201	othdebt	0.146
2	T	2.785	5	othdebt	0.382	creddebt	0.197	card2spent	0.093
3	T	2.561	2	pets saltfish	0.215	othdebt	0.214	creddebt	0.093
4	T	2.279	3	pets saltfish	0.238	pets	0.074	empcat	0.039
5	T	2.070	1	pets reptiles	0.298	pets	0.040	carown	0.039
6	T	2.006	3	pets saltfish	0.209	address	0.072	agecat	0.048
7	T	1.946	2	creddebt	0.114	pets small	0.070	cardspent	0.068
8	T	1.884	2	pets saltfish	0.416	pets	0.064	pets freshfish	0.028
9	T	1.837	3	pets saltfish	0.120	carown	0.043	cartype	0.043
10	T	1.833	2	creddebt	0.212	pets reptiles	0.136	Increddebt	0.029
11	T	1.831	3	pets saltfish	0.121	pets dogs	0.081	pets	0.066
12	T	1.746	1	pets small	0.306	pets dogs	0.072	pets	0.055
13	T	1.741	1	pets reptiles	0.492	commute	0.030	birthmonth	0.019
14	T	1.741	3	pets reptiles	0.251	cardspent	0.122	reason	0.033
15	T	1.717	5	income	0.225	othdebt	0.132	creddebt	0.084
16	T	1.717	1	pets birds	0.348	pets reptiles	0.065	debinc	0.030
17	T	1.716	3	pets reptiles	0.444	reason	0.033	hometype	0.026
18	T	1.714	2	pets saltfish	0.457	birthmonth	0.020	gender	0.018
19	T	1.708	2	pets saltfish	0.459	othdebt	0.045	birthmonth	0.021
20	T	1.704	3	pets saltfish	0.183	address	0.073	pets	0.054

For each record, the model creates 9 new fields. The field **\$O-Anomaly** is a flag field identifying anomalous records (with a value of *T*). The field **\$O-AnomalyIndex** contains the anomaly index value; the maximum value is 5.0, but most of the cases have values below 2.0. The next set of columns lists the specific fields on which a case was anomalous and the impact of that field's value on the anomaly index. This lets you begin reviewing data values for these records by focusing on the fields that make the greatest contribution to the anomaly index.

Only six records have anomaly index values above 2. For the most anomalous record, economic-related fields are identified as being important, including *creddebt*, *income*, and *othdebt*. If we scroll to the left, we will see that this customer has one of the largest incomes, over 1,000K. The second most anomalous record also has economic-related fields as the most important; and both of these records are in peer group 1. On the other hand, the third record lists ownership of reptiles as a pet as an important field. This person is 68 years old and commutes by bicycle to work. (So presumably is in reasonably good shape, if nothing else.)

There is much work to do after you have generated an Anomaly model, but it has great potential at helping you find unusual, and interesting, records.

Summary

In this lesson we have introduced a number of ways of finding and dealing with outliers and anomalous data values and records. You should now be able to:

- Use the Data Audit node to search for unusual values on single fields
- Use various graphs to look for anomalies on two fields at once
- Use the Anomalies node to search for anomalies on many fields at once

Exercises

In these exercises you will use a new data file, *custandhol.dat*, which contains information on customer holidays arranged by a leading travel company. This table lists the fields in the file, which include customer information and holiday details.

CUSTID	Customer reference number
NAME	Customer name
DOB	Date of birth
GENDER	Gender
REGION	Home location
NUMPARTY	Number in party
HOLCOST	Total cost of holiday
NIGHTS	Number of nights away
TRAVDATE	Departure date
HOLCODE	Holiday code
COUNTRY	Country
POOL	Usage of a pool
ACCOM	Type of accommodation
DIST_TO_BEECH	Distance to beach in kms

1. Open the *Define_custandhol.str* file and run the Table node to instantiate the data fields.
2. Connect a Data Audit node and run it. Which fields have outliers? Do any have extreme values?
3. Examine the histogram for *HOLCOST*. Given the distribution of this field and the number of records with outlier values, what alternatives would you consider to handle the outliers?
4. Before making any changes to *HOLCOST*, let's examine the relationship of *NUMPARTY* and *HOLCOST* for outliers. Hint: Make *NUMPARTY* the X field. You might expect the cost to be higher for holidays with more people. Use the Plot node to examine the relationship. Do you see the expected relationship? Is there any pattern to the joint outliers? (HINT: Use the Statistics node to get the correlation coefficient for this relationship.)
5. Now, let's look at one possibility for handling the outlier values for *HOLCOST*. Given what we have seen, we could simply leave them as is; but for practice, we'll look at a couple of other possibilities. First, use the Derive node to make a copy of *HOLCOST*; name the new field *HOLCOST_OUT*. Attach a Type node to instantiate the new field and make certain the measurement level is continuous. Attach a Table to the Type node and run it.
6. Now attach a Data Audit node to the Type node and run it. Click in the Action column cell for *HOLCOST_OUT* in the Quality tab of the Data Audit output. You can choose one of these options to handle the outlier and extreme records. Select Coerce as your choice. Select the *HOLCOST_OUT* row and click Generate...Outlier & Extreme Supernode for the selected field only. (We will learn details about Supernodes in a later lesson.)

7. Connect the generated Supernode to the Type node downstream from the *HOLCOST_OUT* node. Connect a Data Audit node to the Supernode node and Run. Examine the histogram and statistics for *HOLCOST_OUT*. How different are they from the original *HOLCOST*?
8. Now attach a Type node to the Supernode and instantiate the data.
9. Attach an Anomaly node to the Type node. Include all fields except *CUSTID*, *NAME*, and *HOLCOST*. Accept the other defaults and Run. Browse the model. How many peer groups are found? What were the most important fields in identifying anomalous cases?
10. Attach a Sort node to the Anomaly nugget on the stream canvas and edit it to sort in descending order on *\$O-AnomalyIndex*. Finally, attach a Table node and Run. Do any of the cases have Anomaly Indices over 2?

Lesson 6: Introduction to Data Manipulation

Objectives

- Introduce several field operations: Filter, Field Reorder, Derive, and Reclassify
- Show how to check your data transformations
- See how to automatically generate Field operation nodes

Data

In this lesson we will use the text data file *Risk.txt*. The data file contains information concerning the credit rating and financial position of 4117 individuals, along with basic demographic information, such as marital status and gender.

6.1 *Introduction*

In the Data Preparation phase of the CRISP-DM process, you construct the final dataset for modeling. This involves a variety of potential activities, including creating new fields or transforming existing ones, selecting groups of records or sampling from a larger data file, and cleaning data based on checks on data quality.

These tasks are often performed several times, and not in any particular order. That is, you may do some data exploration, followed by some data preparation, then some data exploration of the new fields you create, and then some more data preparation. Even after beginning modeling, it is common to return to the data preparation phase if a model isn't performing adequately or if the output from modeling provides clues about data changes that could be made.

Following this logic, we will do some data preparation in this lesson, but then return to data understanding in Lesson 7.

Techniques to modify and prepare data can be found in either the Record Ops palette (containing tools for manipulating records) or Field Ops palette (containing tools for manipulating fields).

In this lesson we introduce several field operation nodes, including the Filter node, which removes unwanted fields from analysis; the Reorder node, which reorders fields in the data stream and dialogs; the Derive node, used to create new fields in the data stream; and the Reclassify node, which is used to change the coding of or collapse categories for categorical fields.

We will also demonstrate how the Derive and Reclassify nodes can be automatically created using the Generate menu available in the output windows of nodes introduced in the previous lessons. Before discussing the nodes themselves we introduce the CLEM language.

6.2 A Brief Introduction to the CLEM Language

Control Language for Expression Manipulation, or CLEM, is a powerful language for analyzing and manipulating the data that flow along PASW Modeler streams. Data miners use CLEM extensively in stream operations to perform tasks as simple as deriving profit from cost and revenue data or as complex as transforming web log data into a set of fields and records with useable information. CLEM is used in Derive, Select, Sample, Filler, Balance and Report nodes, among others. It permits you to:

- Compare and evaluate conditions
- Derive new fields
- Insert data from records into reports

For a detailed introduction to the CLEM language the reader is referred to the *PASW Modeler 14.0 User's Guide*. In this section we will introduce the basic concepts and commonly used functions available in CLEM.

CLEM expressions are constructed from values, fields, operators, and functions. Values can be:

- Integers: 3, 50, 10000
- Real Numbers: 4.51, 0.0, -0.0032
- Strings (within single quotes): 'male', 'married'

Field names can be referred to:

- Directly: *risk*, *income*
- Within quotes if it is a special field name (usually produced by PASW Modeler when doing machine learning): '\$R-risk', '\$N-profit'

Operators commonly used are given in Table 6.1.

Table 6.1 Commonly Used Operators in CLEM

+	Add	>	greater than
-	Subtract	<	less than
*	Multiply	>=	greater than or equals
/	Divide	<=	less than or equals
**	Raise to the power	=	equal to
div	return the quotient	/=	not equal to
rem	return the remainder on dividing	mod	return the modulus
><	Joins string expressions together (concatenation)		

A few of the commonly-used functions are given in Table 6.2.

Table 6.2 Commonly Used Functions in CLEM

round	rounds to the nearest integer away from 0.5
abs	gives the absolute value
sqrt	Takes the square root
log	Natural logarithm
exp	Raises e to the power of
min / max	Returns the minimum or maximum of its arguments
substring(start, length, string)	Returns part of a string, from start for a specified length

For example:

Sqrt (abs (famincome - income))	will return a number equal to the square root of the absolute difference between the fields income and famincome (family income).
'Mr '>< surname	will return a string consisting of 'Mr Name', where 'Name' represents the value of the field called surname
Age >= 65	will return T (True) if the age field is greater than or equal to 65 and F (False) if not.

CLEM expressions will either return a result or evaluate to true or false.

Important Note about Case Sensitivity

CEML expressions are case sensitive so be forewarned! This includes field names.

6.3 Field Operations and the Filter Node

PASW Modeler has a number of nodes that allow you to manipulate fields within the dataset. In this section we introduce the Filter node that can rename fields and remove unwanted fields from the data stream. If these functions need to be performed when the data are first read, the filter tab of any source node can be used (see Lesson 3).

In the last lesson we checked the distribution of a few of the fields in our dataset. When data mining, two potential problems may occur within a field:

- A large proportion of missing records
- All (or almost all) records having the same value (invariant)

The Filter node (or Filter tab of a source node) allows data to pass through it and has two main functions:

- To filter out (discard) unwanted fields
- To rename fields

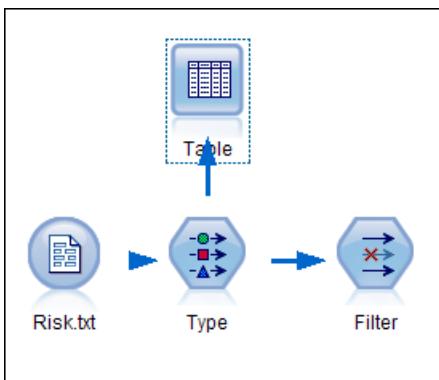
If the Stream Canvas is not empty, choose **File...Close Stream** (and click No if asked to save)

Click **File...Open Stream**, navigate to the **c:\Train\Modeler\Intro** directory and double-click **Riskdef.str**

Place a **Filter** node from the Field Ops palette to the right of the **Type** node

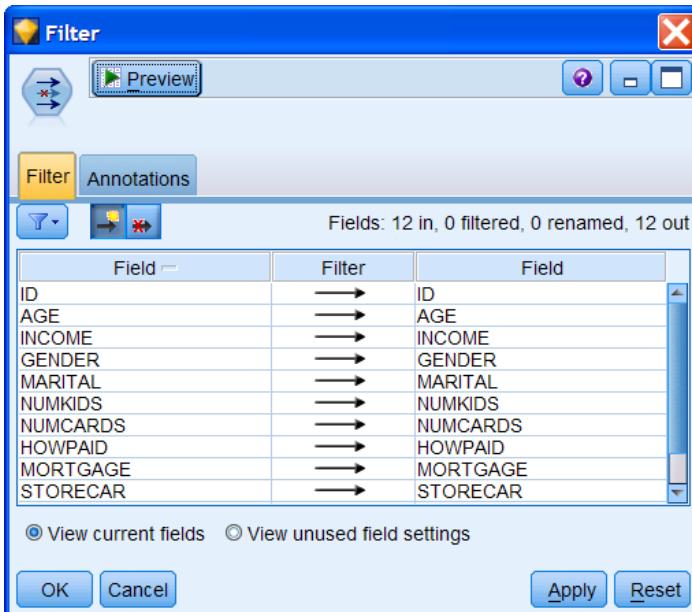
Connect the **Type** node to the **Filter** node

Figure 6.1 Stream with a Filter Node



Right-click on the **Filter** node, and then click **Edit**

Figure 6.2 Filter Node Dialog



The left column lists the field names as the data stream enters the Filter node. The right column shows the field names as the data stream leaves the Filter node. By default the lists are the same.

Text at the top of the dialog indicates the number of fields entering the Filter node, the number of fields filtered, the number of fields renamed, and the number of fields leaving it.

To Change the Name of a Field

To demonstrate changing a field name, we will change the name *STORECAR* to *STORECARDS* (the number of store credit cards).

Click the text **STORECAR** in the right column (right of the arrow)

Type the new name **STORECARDS** in the text box (replace the original name, or simply append **DS** to it)

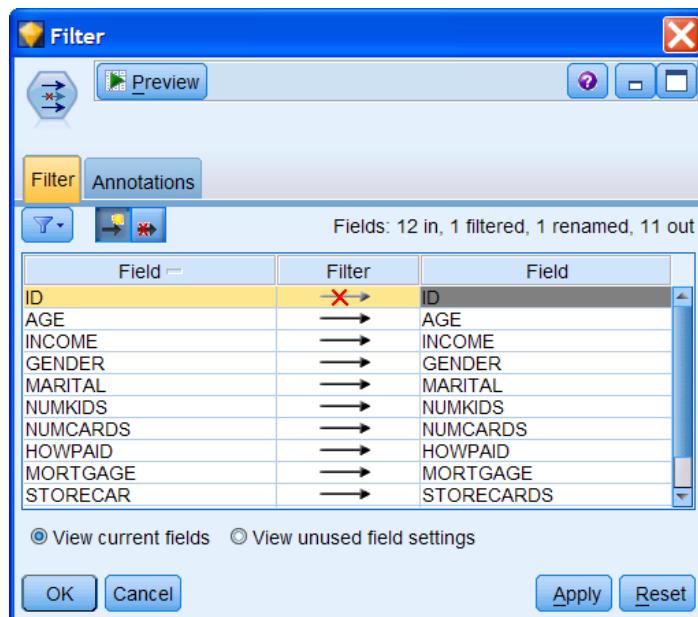
The new name should appear in the right column (see Figure 6.3).

To Filter Out Fields

To demonstrate how to remove fields from the data stream, we will filter out the *ID* field. This involves clicking on the arrow connecting the input (left column) to the output (right column) in the Filter node dialog.

Click on the arrow next to **ID**

Figure 6.3 ID Removed from Stream and STORECAR Renamed



To reinstate a previously filtered field, click on the crossed arrow. The original arrow will be displayed and the output field name will be reinstated.

Multiple fields can be removed. Simply click and drag from the arrow for the first field to be omitted to the arrow for the last field, highlighting the fields to be omitted and then click anywhere on the highlighted area (or right-click, then click Remove the selected fields). To reinstate multiple omitted fields, simply repeat the process. The Filter options menu provides a set of options that are useful when working with a large number of fields.

Click the **Filter options** button

Figure 6.4 Filter Options Menu

Filter options include removing or including all fields, toggling the remove/include settings for all fields, removing or renaming duplicate field names, anonymizing field names, and editing multiple response sets. Removing or renaming duplicate fields are especially helpful when working with database files with many fields, since they provide an easy way of setting the filter options for all fields.

Press the **Esc** key to close the Filter Options menu

The quickest way to check that the Filter node is doing its job is to connect it to a Table node and view the output. We will view this table shortly.

Click **OK** to close the **Filter** dialog box

6.4 Field Reordering

Another useful field operation is to reorder the fields, which would affect their ordering in dialog boxes and data streams. For example, you might want a specific field ordering in a table to better compare the outcome with predictions from different models, or it might be easier to locate field names in dialogs if they were alphabetically ordered. The Field Reorder node will reorder fields downstream of the node and has several options for field reordering, including custom ordering. To illustrate:

Place a **Field Reorder** node from the Field Ops. Palette to the right of the **Filter** node

Connect the **Filter** node to the **Field Reorder** node

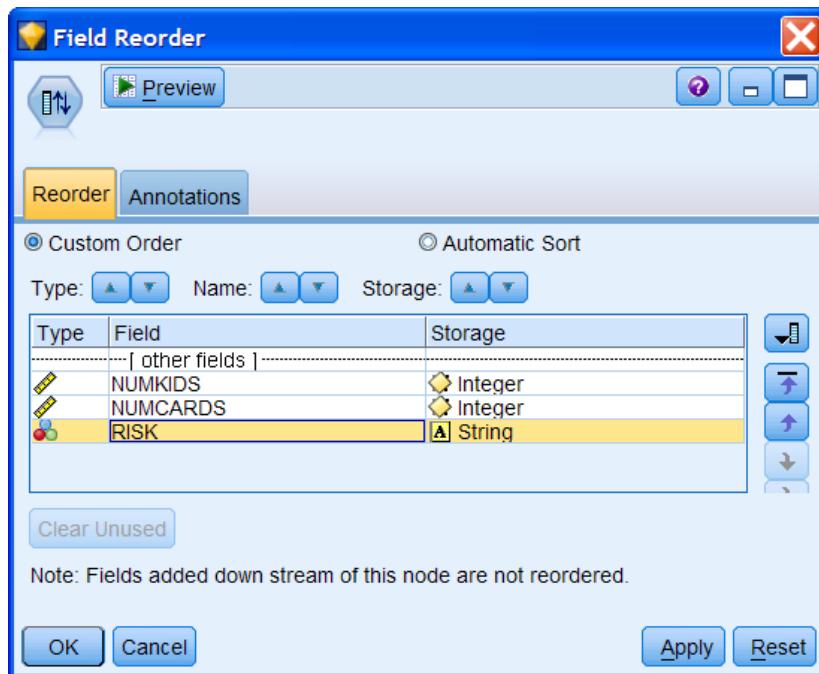
Right-click the **Field Reorder** node, and then click **Edit**

Click the **Field list** button

Select (Ctrl-click) **NUMKIDS**, **NUMCARDS**, and **RISK** in the Select Fields dialog

Click **OK**

Click **RISK** in the Field Reorder dialog

Figure 6.5 Field Reorder Node Dialog

The field order shown in the Field Reorder dialog controls field ordering downstream. The [*other fields*] item represents fields not explicitly listed in the Field Reorder dialog. The current ordering would have *NUMKIDS*, *NUMCARDS* and *RISK* appearing as the last three fields, preceded by the other fields in their original order.

You can change the order of selected fields using the buttons. Selected fields or the [*other fields*] item can be moved up or down one position or moved to the top or bottom of the list. In addition, when *Custom Order* is selected, the fields in the list can be sorted in ascending or descending order by *Type*, *Field* (name), or *Storage*. When any of these sorts are performed, the [*other fields*] item moves to the bottom of the list.

If the *Automatic Sort* option is chosen, then all fields can be sorted by *Type*, *Field*, or *Storage*.

Click the **Move selected fields to the top** button

This will reorder the fields so that *RISK* appears first, followed by all fields in their original order except *NUMKIDS* and *NUMCARDS*, which appear last.

Click the **Preview** button

Figure 6.6 Table Following the Filter and Field Reorder Operations

The screenshot shows a software window titled "Preview from Field Reorder Node (11 fields, 10 records) #1". The window has a menu bar with "File", "Edit", "Generate", and "OK" buttons. Below the menu is a toolbar with icons for "Table" and "Annotations". The main area displays a table with 11 columns and 10 rows. The columns are labeled: RISK, AGE, INCOME, GENDER, MARITAL, HONPAID, MORTGAGE, STORECARDS, LOANS, NUMKIDS, and NUMCARDS. The data rows show various values for each field, such as "good risk" for RISK and "44" for AGE.

ID	RISK	AGE	INCOME	GENDER	MARITAL	HONPAID	MORTGAGE	STORECARDS	LOANS	NUMKIDS	NUMCARDS
1	good risk	44	59944	m	married	monthly	y	2	0	1	2
2	bad loss	35	59692	m	married	monthly	y	1	0	1	1
3	good risk	34	59508	m	married	monthly	y	2	1	1	1
4	bad loss	34	59463	m	married	monthly	y	1	1	0	2
5	good risk	39	59393	f	married	monthly	y	1	0	0	2
6	good risk	41	59276	m	married	monthly	y	1	1	1	2
7	good risk	42	59201	m	married	monthly	y	2	0	0	1
8	good risk	31	59193	f	married	monthly	y	1	1	1	2
9	bad loss	28	59179	m	married	monthly	y	2	1	1	1
10	good risk	30	59036	m	married	monthly	y	2	1	1	1

The *ID* field is no longer present and the *STORECAR* field has been renamed. *RISK* is now the first field while *NUMKIDS* and *NUMCARDS* are in the last two positions.

We have examined the Filter node as a method of renaming and discarding fields within the data and have seen how to use the Field Reorder node. It is often the case, however, that the values themselves within the fields need to be altered, or new fields created as combinations of existing fields. In the next section we will introduce the Derive node as a method of performing such data manipulations.

Close the Preview window

Close the **Field Reorder** node

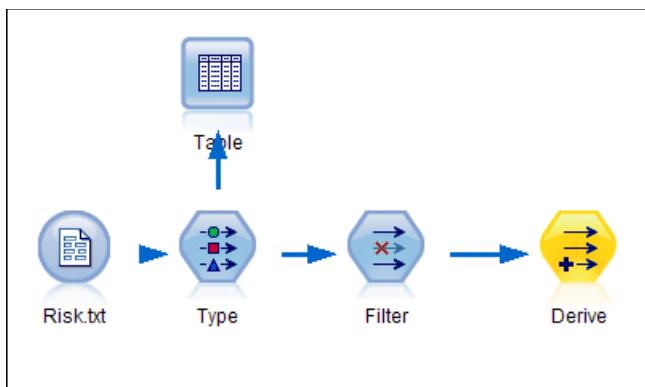
Right-click the **Field Reorder** node, then click **Delete**

6.5 The Derive Node

In order to make full use of the modeling techniques available in PASW Modeler, it will invariably be necessary to modify data values or create new fields as functions of others. The Derive node calculates a new value based on a CLEM expression for every record passed through it. To enter the CLEM expression and the new field name (a name is automatically assigned) you need to edit the Derive node.

You can use the Insert menu to add a node to the Stream Canvas.

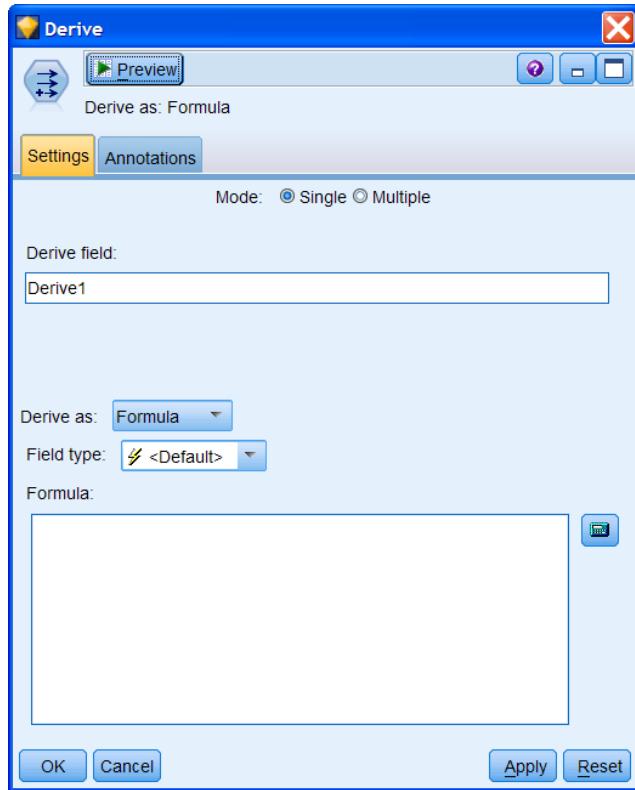
Click **Insert...Field Ops...Derive**

Figure 6.7 Adding a Derive Node

Note that the node is automatically connected to the stream and the active (selected) node.

Right-click the **Derive** node, then click **Edit**

Figure 6.8 Derive Node Dialog



The new field name is entered in the *Derive field* text box. Remember that PASW Modeler is case sensitive with respect to field names.

The Derive node offers six different methods to create a new field. Clicking the *Derive as* drop-down list will reveal these options:

Table 6.3 Derive As Choices to Create New Field

Formula	The new field is the result of an arbitrary CLEM expression.
Flag	The resulting field will have a True or False response (flag), reflecting a specified expression.
Nominal	The new field will have values assigned from members of a specified set.
State	The new field's value represents one of two states. Switching between these states is triggered by specified conditions.
Count	The new field is based on the number of times a specified condition is true.
Conditional	The new field is the result of one of two expressions, depending on the value of a condition.

The Count and State derive types are most often used with time series or sequence data and are discussed in the *Preparing Data for Data Mining* training course.

Once the type of derivation is chosen, the dialog box changes appropriately. The measurement level of the field to be derived can explicitly be set using the Field type option. For the moment, we will leave it to its default value.

Single versus Multiple Derive Mode

A Derive node is usually used to calculate a single new field, which is why the *Single* Derive Mode option button is selected by default. For instances in which you need to calculate multiple new fields using the same operations applied to a series of fields, the *Multiple* Derive Mode option is available. In this mode you select the fields to which the operations will be applied, indicate how the new fields are to be named (by adding a user-specified prefix or suffix to the original field names), and specify the data transformation operations. To accommodate this, the Derive node dialog will expand when the *Multiple* Mode option is selected. In this course, we demonstrate the Single Derive mode. Some examples of multiple Derive mode are: applying a natural log transformation to a set of fields that will be used in a neural net; creating a series of flag fields coded as F (0 or negative balance) or T (positive balance) based on a set of financial account fields.

Derive Type Formula

For this example we will calculate a composite measure of potential debt, which is equal to the sum of the number of credit cards (*NUMCARDS*), number of store cards (*STORECARDS*—after renaming within the Filter node) and number of loans (*LOANS*) for each record.

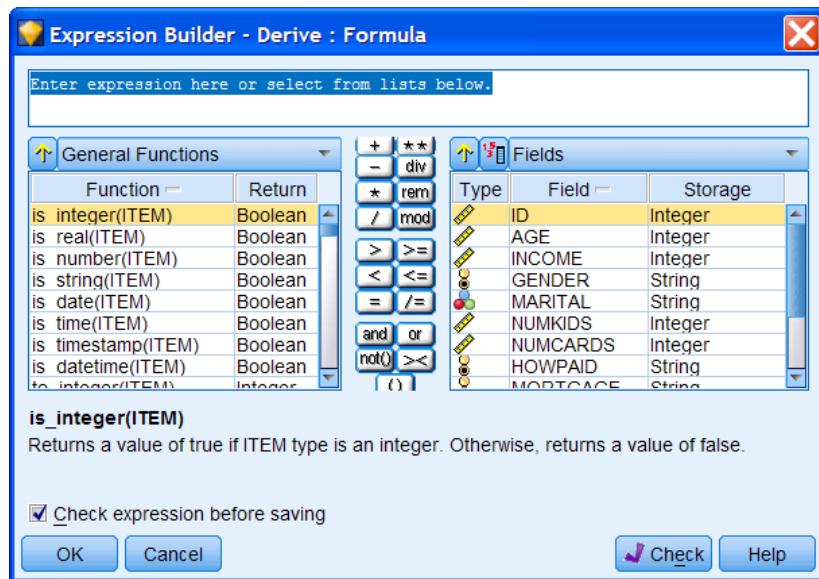
Select **Formula** from the **Derive as:** drop-down list (if necessary)
Type **SUM DEBT** in the **Derive field:** text box (replacing the current name)

You can type the equation into the Formula text box, but it is easier to invoke the Expression Builder in which you can create an expression by clicking the operation buttons and selecting fields and functions from list boxes.

Click the **Expression Builder**  button

The expression is built in the large text box. Operations (addition, subtraction, etc.) can be pasted into the expression text box by clicking the corresponding buttons. The Function list contains functions available in PASW Modeler.

By default, a general list of functions appears, but you can use the drop-down list above the Function list box to display functions of a certain type (for example, Date and Time functions, Numeric functions, Logical functions). Similarly, by default, all fields in the stream are listed in the Fields list box and can be pasted into the expression by clicking the Insert button  after the field is selected. The Fields list box can display all fields, recently used fields, parameters, or globals. The latter two categories are discussed in the *PASW Modeler User's Guide*.

Figure 6.9 Expression Builder Dialog

Additionally, this dialog can display field information for a selected field, which, in turn, can be pasted into the expression. To illustrate:

Click **MARITAL** in the **Fields** list box

Click the **Select from existing field values** button

Figure 6.10 Field Values for MARITAL in the Insert Value Dialog

Since *MARITAL* is a nominal field, the Insert Value dialog contains a list of its values and these can be pasted into the expression using the **Insert** button. Depending on the field's type, different information will display—similar to what we saw when examining the Types tab. In addition to allowing you to paste expressions, the Expression Builder will check the validity of the expression. Such features provide a powerful tool with which to construct expressions.

Now to build the equation:

Click **Close** button to close the Insert Value dialog

Click **NUMCARDS** in the Fields list box, then click the **Insert** button

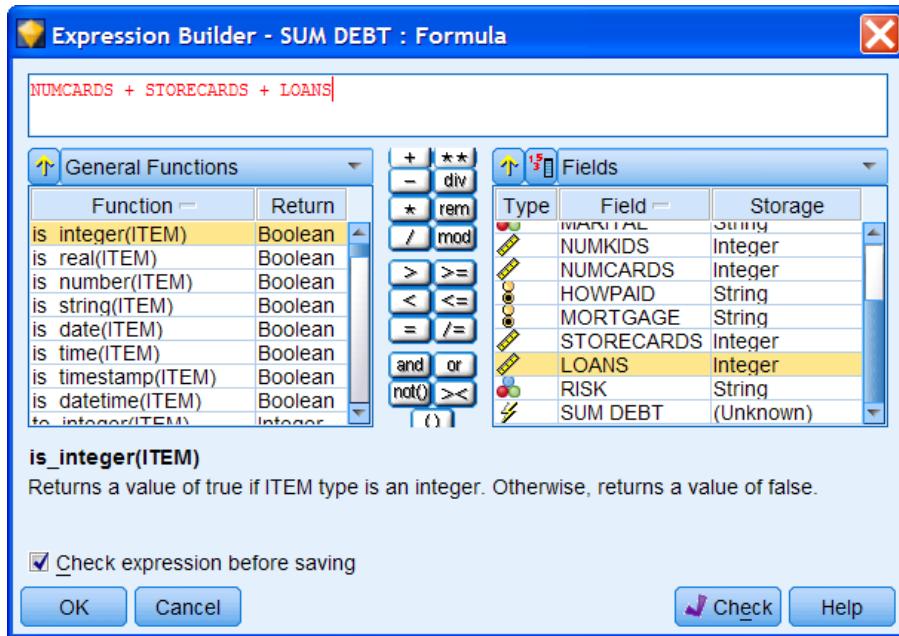
Click the plus button

Click **STORECARDS** in the Fields list box, then click the **Insert** button

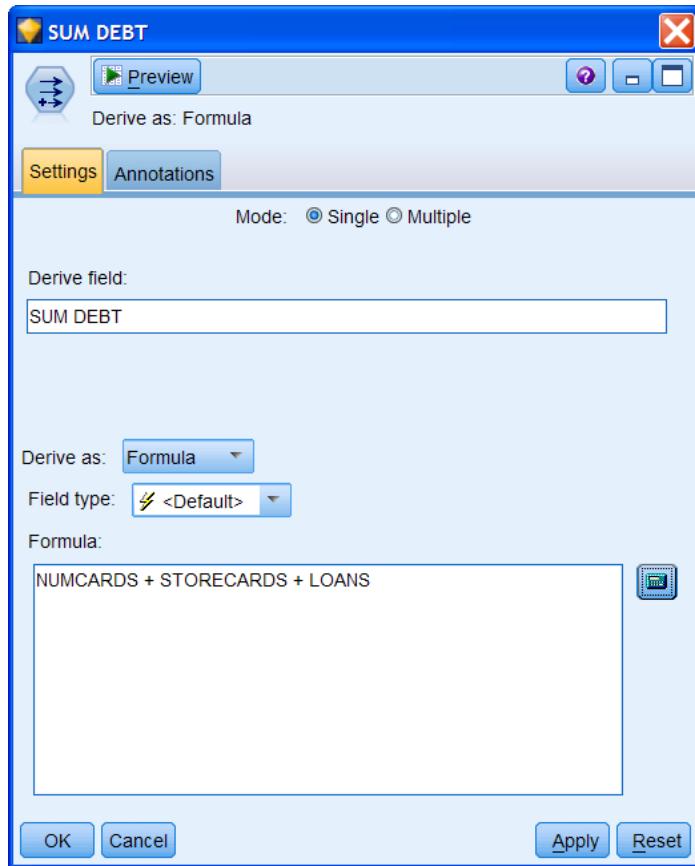
Click the plus button

Click **LOANS** in the Fields list box, then click the **Insert** button 

Figure 6.11 Completed Expression in the Expression Builder



Click **OK**

Figure 6.12 Completed Derive Node Dialog Box for a Formula

Click OK

On clicking the OK button, we return to the Stream Canvas, where the Derive node is labeled with the name of the new field. The node itself doesn't create any output, just passed downstream with the newly created field.

Derive Type Flag

For the next example we will create a new field, *CHILDREN*, which is True if the number of children field (*NUMKIDS*) is greater than zero, and False if not.

Place a **Derive** node from the **Field ops** palette to the right of the **Derive** node named **SUM DEBT**

Connect the **Derive** node named **SUM DEBT** to the new **Derive** node

Right-click the new **Derive** node, then click **Edit**

Click **Flag** on the **Derive as:** drop-down list

Type **CHILDREN** in the **Derive field:** text box (replace the original name)

Type **NUMKIDS > 0** in the **True when:** text box

Figure 6.13 Derive Node Dialog Box for Type Flag

If the number of children (*NUMKIDS*) is greater than 0 for a record, the *CHILDREN* field will be assigned the value “T”. Otherwise, it will be assigned the value “F”. These values can be changed by entering text in the *True value:* and *False value:* text boxes.

Click **OK**

Derive Type Nominal

For the third example we will create a new field called *INCGROUP*, which is the *INCOME* field banded into 3 bands:

- Under 20,000
- 20,000 to 35,000
- Over 35,000

This is a typical procedure in data mining where we try other versions of a field, even a continuous one, to see if it helps improve a model. If we wanted to create income categories that were equal width, equal sized, or were based on standard deviation width, we would instead use the Binning node from the Field Ops palette.

Place a **Derive** node from the **Field Ops** palette to the right of the **Derive** node named **CHILDREN**

Connect the **Derive** node named **CHILDREN** to the new **Derive** node

Right-click the new **Derive** node, then click **Edit**

Click **Nominal** on the **Derive as:** drop-down list

The dialog box contains two options to be completed for each member of the nominal field:

- *Set field to* indicate a value in the new field when a particular condition is met.
- *If this condition is true* indicating the test condition for a particular value

PASW Modeler will test the conditions and assign the value stored in *Set field to* for the first condition that applies. The *Default value* is assigned if no condition test applies.

To enter each category of the nominal field, type its value in the *Set field to* text box and the test condition in the *If this condition is true* text box. If you have a required value for the nominal field when none of the conditions apply, enter this in the *Default value* text box. The Expression Builder can be used to construct the If condition.

Type **INCGRP** in the **Derive field:** text box

Type **No income** in the **Default value:** text box

Type **Low** in the first **Set field to:** text box

Type **INCOME < 20000** in the **If this condition is true:** text box

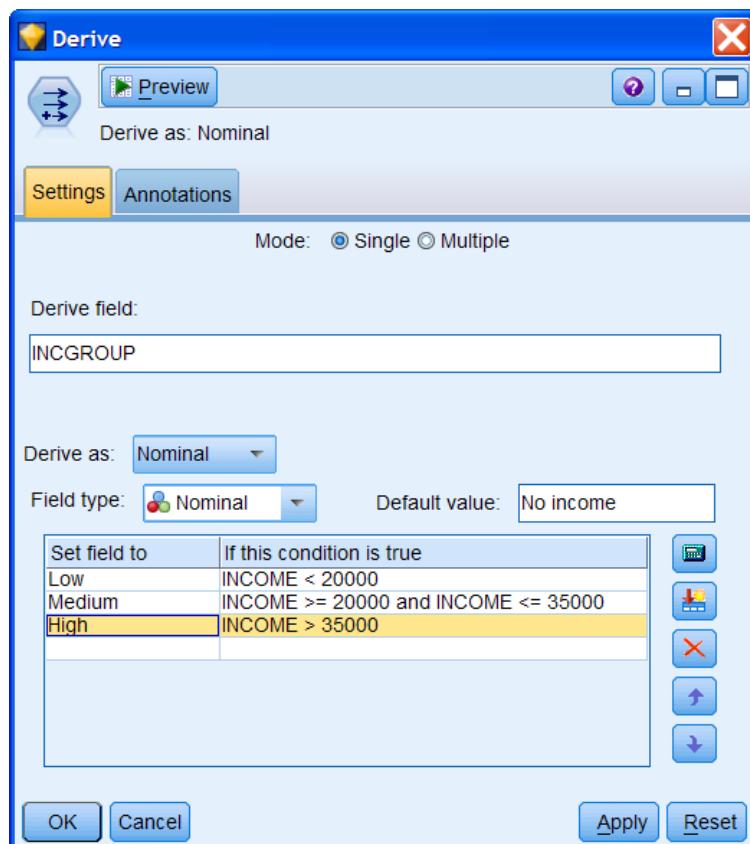
Type **Medium** in the next **Set field to:** text box

Type **INCOME >= 20000 and INCOME <= 35000** in the **If this condition is true:** text box

Type **High** in the **Set field to:** text box (replace old value)

Type **INCOME > 35000** in the **If this condition is true:** text box

Figure 6.14 Derive Node Dialog for Nominal Derive



Click **OK**

Derive Type Conditional

For the last example of the Derive node we will create a new field called NEWRISK that will be another measure of the risk of offering credit to a consumer based on three of the existing fields: *NUMCARDS*, *STORECARDS*, and *LOAN*. Basically, those consumers with lower values on these three fields will be in one category, and any high value will place someone in the more risky category.

Place a **Derive** node from the **Field Ops** palette to the right of the **Derive** node named **INCGROUP** in the Stream Canvas

Connect the **Derive** node named **INCROUP** to the new **Derive** node

Right-click the new **Derive** node, then click **Edit**

Click **Conditional** on the **Derive as** drop-down list

Type **NEWRISK** in the **Derive field:** text box

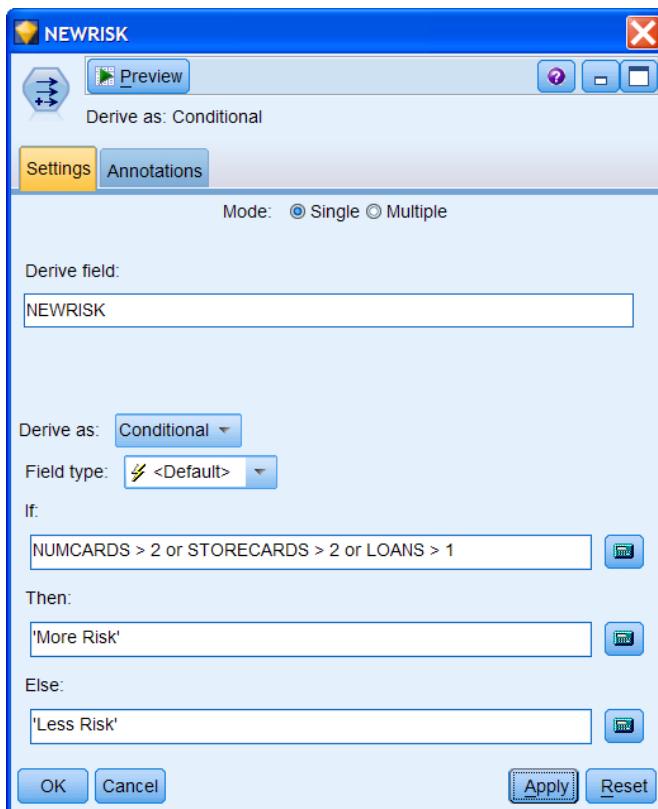
Type **NUMCARDS > 2 or STORECARDS > 2 or LOANS > 1** in the **If:** text box

Type **'More Risk'** in the **Then:** text box

Type **'Less Risk'** in the **Else:** text box

Be sure to include the apostrophes around the values in the Then: and Else: text boxes since text values must be put between apostrophes.

Figure 6.15 Derive Node Dialog for Conditional Derive



The conditional type will only allow two conditions. If the If: expression applies, the expression in the Then: box will be calculated, otherwise the Else: expression will be calculated.

Click **OK**

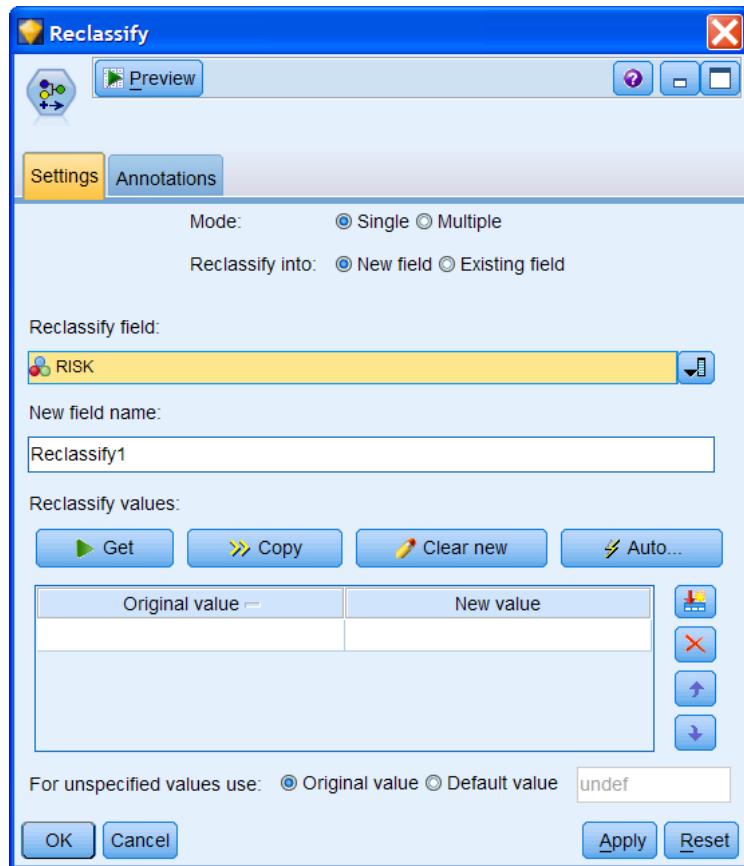
It is important to check your work when creating new fields with the Derive node. We will do so, but first introduce another useful node, the Reclassify node.

6.6 Reclassify Node

The Reclassify node allows you to reclassify, or recode, the data values for categorical fields. For example, a field that stores a customer's specific job position may be more useful for prediction if it is reclassified into broader job categories. The reclassified values can replace the original values for a field, although a safer approach is to create a new field, retaining the original. We will demonstrate by reclassifying the three values of the *RISK* field (bad loss, bad profit, good risk) into two values (bad and good).

Place a **Reclassify** node from the **Field Ops** palette to the right of the **Derive** node named **NEWRISK** in the Stream Canvas
 Connect the **Derive** node named **NEWRISK** to the **Reclassify** node
 Right-click the **Reclassify** node, then click **Edit**
 Click the **field list** button  for **Reclassify field** and select **RISK**

Figure 6.16 Reclassify Dialog



As we saw for the Derive node, the Reclassify node supports Single and Multiple modes. Multiple mode would be useful if the same reclassification rules were to be applied to a number of fields. By default, the new values will be placed in a new field, although the *Reclassify into Existing field* option permits you to modify the values in an existing field.

Within the *Reclassify values* group, the Get button will populate the *Original value* column with values from the upstream Type node or Types tab. Alternatively, you can enter the original values directly. The Copy button will copy the values currently in the *Original value* column into the *New Value* column. This is useful if you want to retain most of the original values, reclassifying only a few. The Clear new button will clear values from the *New value* column (in case of errors), and the Auto button will assign a unique integer code to each value in the *Original value* column. This option is useful for replacing sensitive information (customer IDs, customer names, product names) with alternative identifiers, or reclassifying string data to numeric.

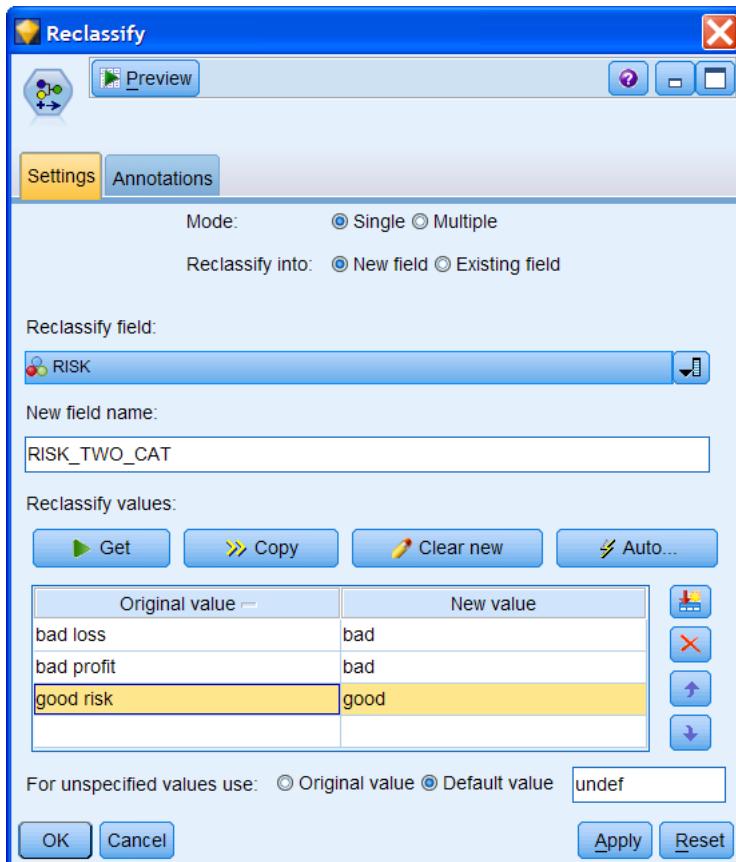
You have options to use the *Original value* or a *Default value* when a value not specified in the *New value* column is encountered in the data stream.

- Type **RISK_TWO_CAT** in the **New field name** box
- Click the **Get** button
- Click the For unspecified values use: **Default value** option button
- Type **bad** in the **New value** box for the **bad loss** row
- Type **bad** in the **New value** box for the **bad profit** row
- Type **good** in the **New value** box for the **good risk** row

Tip

Be careful not to click in the row below “good risk” as this will create another, null value and cause an error when this node is run.

Figure 6.17 Completed Reclassify Dialog



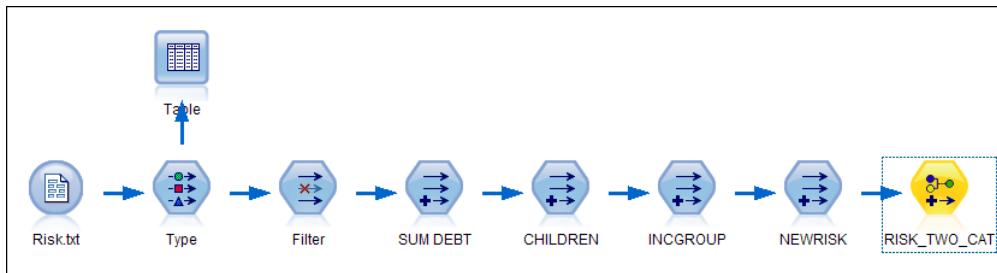
The Reclassify dialog will create a field named *RISK_TWO_CAT* that will have the values *bad*, *good*, or *undef*.

Click **OK**

6.7 Executing Field Operation Nodes Simultaneously

In the previous examples we attached each of the field operations nodes (except the first) to the previous one. Because of this, all the new fields will be added to the same stream, which is normally desirable. This allows us to use them altogether downstream for modeling. If we wished to create a separate stream for each field operation node, we could have attached each directly to the Type node, but this would be unusual.

Figure 6.18 Field Operation Nodes Placed in One Stream

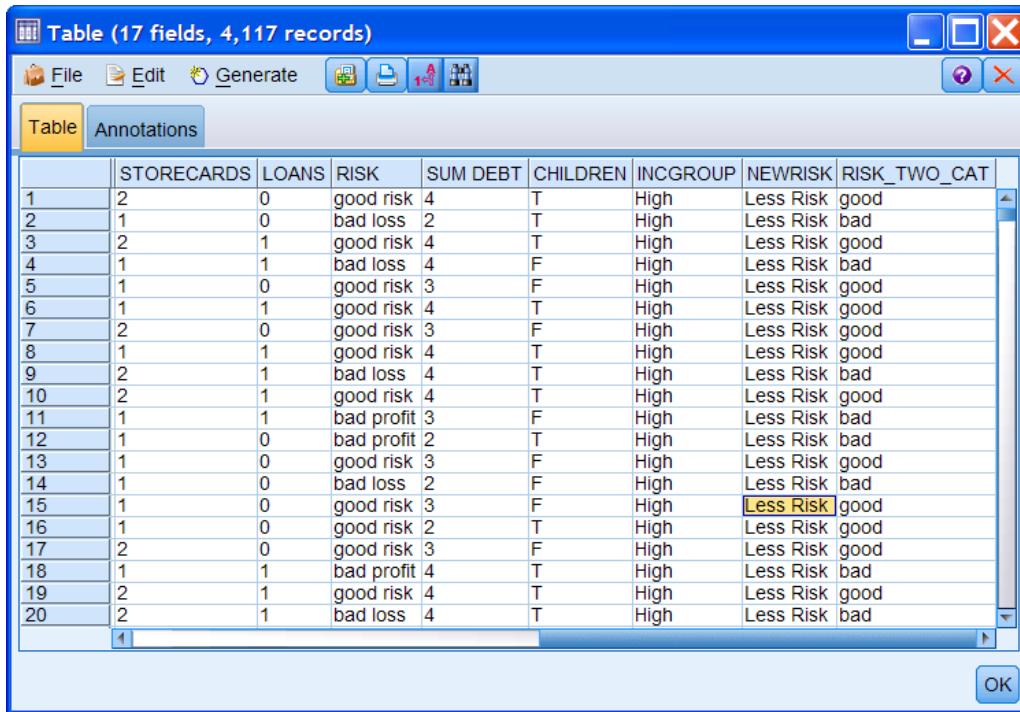


To demonstrate, we will add a new Table node to the stream.

Place a **Table** node from the Output palette above the **Reclassify** node (named **RISK_TWO_CAT**)

Connect the **Reclassify** node to the new **Table** node

Right-click the new **Table** node, then click **Run**

Figure 6.19 Table Showing Fields Created by Derive and Reclassify Nodes


The screenshot shows a Windows application window titled "Table (17 fields, 4,117 records)". The menu bar includes "File", "Edit", "Generate", and various icons. Below the menu is a tab bar with "Table" selected. The main area is a grid of data with 20 rows and 9 columns. The columns are labeled: STORECARDS, LOANS, RISK, SUM DEBT, CHILDREN, INCGROUP, NEWRISK, and RISK_TWO_CAT. The data shows various combinations of values across these fields, such as different risk levels (good, bad, less risk) and debt amounts (0, 1, 2, 3, 4). An "OK" button is visible at the bottom right of the window.

	STORECARDS	LOANS	RISK	SUM DEBT	CHILDREN	INC GROUP	NEW RISK	RISK TWO CAT
1	2	0	good risk	4	T	High	Less Risk	good
2	1	0	bad loss	2	T	High	Less Risk	bad
3	2	1	good risk	4	T	High	Less Risk	good
4	1	1	bad loss	4	F	High	Less Risk	bad
5	1	0	good risk	3	F	High	Less Risk	good
6	1	1	good risk	4	T	High	Less Risk	good
7	2	0	good risk	3	F	High	Less Risk	good
8	1	1	good risk	4	T	High	Less Risk	good
9	2	1	bad loss	4	T	High	Less Risk	bad
10	2	1	good risk	4	T	High	Less Risk	good
11	1	1	bad profit	3	F	High	Less Risk	bad
12	1	0	bad profit	2	T	High	Less Risk	bad
13	1	0	good risk	3	F	High	Less Risk	good
14	1	0	bad loss	2	F	High	Less Risk	bad
15	1	0	good risk	3	F	High	Less Risk	good
16	1	0	good risk	2	T	High	Less Risk	good
17	2	0	good risk	3	F	High	Less Risk	good
18	1	1	bad profit	4	T	High	Less Risk	bad
19	2	1	good risk	4	T	High	Less Risk	good
20	2	1	bad loss	4	T	High	Less Risk	bad

Although not shown here, another useful field operation node is the Binning node, which is often used to convert numeric fields into nominal fields—for example, income into income groups. This node is documented in the *Preparing Data for Data Mining* course guide.

Checking New Fields

Whenever you create new fields with any node, you should always double-check your work. It is possible for you to make an error when creating a new field, either by entering a formula incorrectly, misspecifying a value for a field, or making a logical error in an IF statement. So long as the CLEM expression you enter is syntactically correct, PASW Modeler will not generate an error message, so simply not receiving an error does not mean that the new field is correct.

How to check a new field depends somewhat on how that field was created.

- For fields created from formulas, you might simply review the output from a Table node and calculate a few values yourself to check the equation
- For fields created with logical conditions from other fields, or from the Reclassify node, you might use the Matrix node as a check.

If you review the tabular output, you can, for example, easily confirm whether the new field *SUM_DEBT* was correctly constructed from the sum of *NUMCARDS*, *STORECARDS*, and *LOANS* (it was).

To verify that the field *RISK* was correctly reclassified into *RISK_TWO_CAT*, we can use a Matrix node, which is essentially a crosstab table (there will be a more detailed review of the Matrix node in Lesson 7).

Click **File...Close** to close the Table window

Place a **Matrix** node from the Output palette below the Reclassify node named RISK_TWO_CAT

Connect the **Reclassify** node to the **Matrix** node

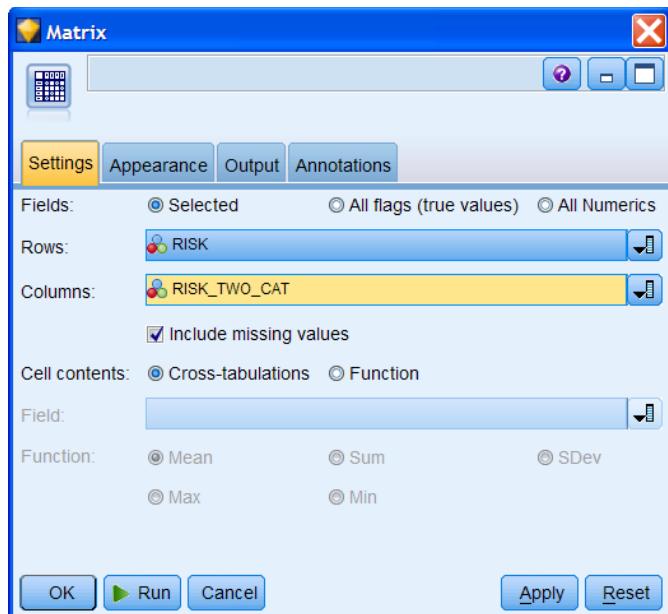
Double-click the **Matrix** node

Click the Field Chooser button and select **RISK** for the Rows: field

Click the Field Chooser button and select **RISK_TWO_CAT** for the Columns: field

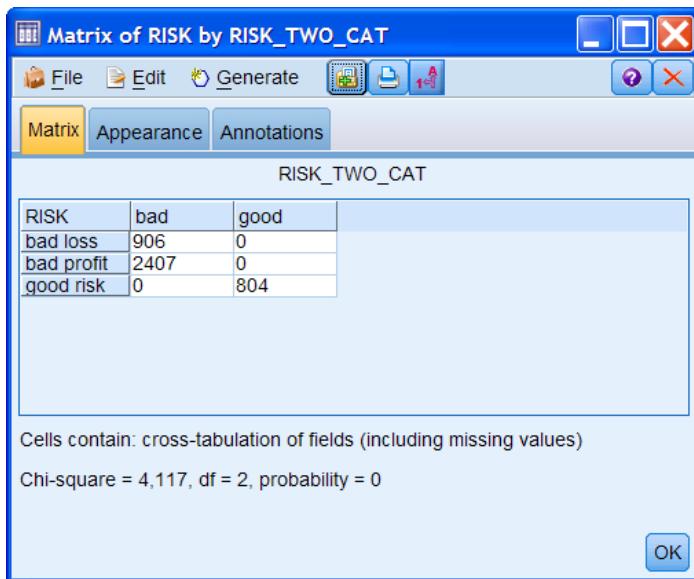
By default the Matrix node will produce a table showing the counts in the cells for records that have combinations of values on the two fields.

Figure 6.20 Matrix Node Request to Check RISK_TWO_CAT



Click Run

The table in the Matrix Browser output allows us to check the reclassification of **RISK** because all the bad loss and bad profit values should have values of *bad* on **RISK_TWO_CAT**, and the good risk value should have a value of *good*. This is indeed the case.

Figure 6.21 Matrix Browser of RISK and RISK_TWO_CAT

Close the Matrix Browser window

Hint

You may be tempted to use a Matrix node to check the construction of the field *CHILDREN*, but you would soon discover that the field *NUMKIDS* from which it was created doesn't appear in the field list in the Matrix node dialog. This is because *NUMKIDS* is of continuous and the Matrix node is designed to only display fields on the rows and columns that are categorical (flags, nominal fields, ordinal fields).

Although PASW Modeler contains this feature to prevent you from asking for output that doesn't make much sense, or that would be difficult to use, it admittedly is a limitation on what you can do. To use *NUMKIDS* in a Matrix node, you can connect a new Type node to the Reclassify node, then modify the type of *NUMKIDS* to make it an ordinal field (ask your instructor to show you how to do this if you are interested). Then you can use the Matrix node to check *CHILDREN*.

Missing Values in Derive Nodes

We discussed missing data in PASW Modeler in Lesson 4, and we mentioned there that defining missing data in the Type node (or Types tab of a Source node) doesn't necessarily mean that PASW Modeler will do anything special with the missing data. In other words, recognizing that a field has missing values doesn't automatically lead to doing something special when PASW Modeler encounters a missing value on a record.

When deriving new fields, the type of missing data and the type of new field being created interact to affect how the missing data is handled. Briefly, when a numeric field is being created with a formula, if any of the fields in the formula has a null value (*\$null\$*), the answer returned will be a null value (*\$null\$*). If instead a nominal or flag field is being created, a missing value on a field being used in a flag or conditional derivation will not prevent PASW Modeler from using the remaining fields to calculate a value for the new field.

Thus, in the examples above, imagine that for the first record *NUMCARDS* was blank in the file *Risk.txt*. PASW Modeler would store the null value for numeric data in that field. When it calculated *SUM_DEBT* for that record, the resulting value would also be the null value (see Figure 6.22). But the missing value for *NUMCARDS* would not prevent PASW Modeler from creating a valid value for *NEWRISK* (it created the value Less Risk for the first record because *STORECARDS* was not greater than 2 and *LOANS* was not greater than 1).

Figure 6.22 Effect of Missing Data on Creating New Fields

	MARITAL	NUMKIDS	NUMCARDS	HOWPAID	MORTGAGE	STORECARDS	LOANS	RISK	SUM DEBT	CHILDREN
1	married	1	\$null\$	monthly	y	2	0	good risk	\$null\$	T
2	married	1	1	monthly	y	1	0	bad loss	2	T
3	married	1	1	monthly	y	2	1	good risk	4	T
4	married	0	2	monthly	y	1	1	bad loss	4	F
5	married	0	2	monthly	y	1	0	good risk	3	F
6	married	1	2	monthly	y	1	1	good risk	4	T
7	married	0	1	monthly	y	2	0	good risk	3	F
8	married	1	2	monthly	y	1	1	good risk	4	T
9	married	1	1	monthly	y	2	1	bad loss	4	T
10	married	1	1	monthly	y	2	1	good risk	4	T
11	married	0	1	monthly	y	1	1	bad profit	3	F
12	married	1	1	monthly	y	1	0	bad profit	2	T
13	married	0	2	monthly	y	1	0	good risk	3	F
14	married	0	1	monthly	y	1	0	bad loss	2	F
15	married	0	2	monthly	y	1	0	good risk	3	F

You need to keep this behavior in mind when creating new fields. You may wish to handle the missing data explicitly, or even remove it from the stream (see Lesson 10).

6.8 Automatically Generating Operational Nodes

In this lesson we have introduced methods of manually adding operational nodes. PASW Modeler also allows automatic generation of many of the nodes we have manually created. In the previous lessons, output windows often contained a Generate menu. This menu frequently allows automatic generation of Derive and other nodes. For example, the Distribution node output window can generate Select, Derive, Balance, and Reclassify nodes. We will now demonstrate two examples of automatically generating nodes.

Automatically Generating a Derive Node to Group Income

In this section we generate a Nominal type Derive node—possibly the most tedious to create manually (although in many cases the Binning node will accomplish the task easily for numeric data). We will group the *INCOME* field into four categories.

- Place a **Histogram** node from the Graphs palette near the **Reclassify** node
- Connect the **Reclassify** node to the **Histogram** node
- Double-click the **Histogram** node
- Click the field list button and select **INCOME** (not shown)
- Click the **Run** button

After the Histogram graph window appears we want to work with the data in the graph. This means we need to work in Explore mode.

- Click **View...Explore Mode** (if necessary)

Explore mode allows you to examine the data and values represented by the graph. The main goal of exploration is to analyze the data and then identify values using bands, regions, and marking to generate Select, Derive, or Balance nodes. We want to create bands for binning and we need to use the Draw Bands toolbar button . Once selected, you click on the graph to create a band.

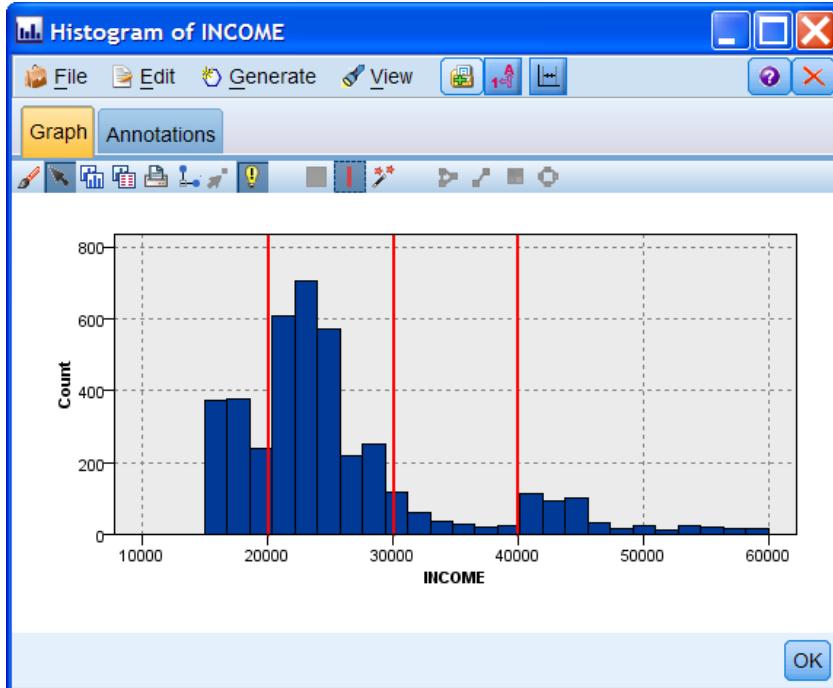
Click on the Draw Bands tool 

Click on the **Histogram** graph at (approximately) **20,000** (a vertical line should appear); Hint: the band will appear where the arrow is pointing on the tool, not where the red vertical line is located

Click on the **Histogram** graph at (approximately) **30,000**

Click on the **Histogram** graph at (approximately) **40,000**

Figure 6.23 Histogram with Interactive Lines Dividing Income into Four Bands



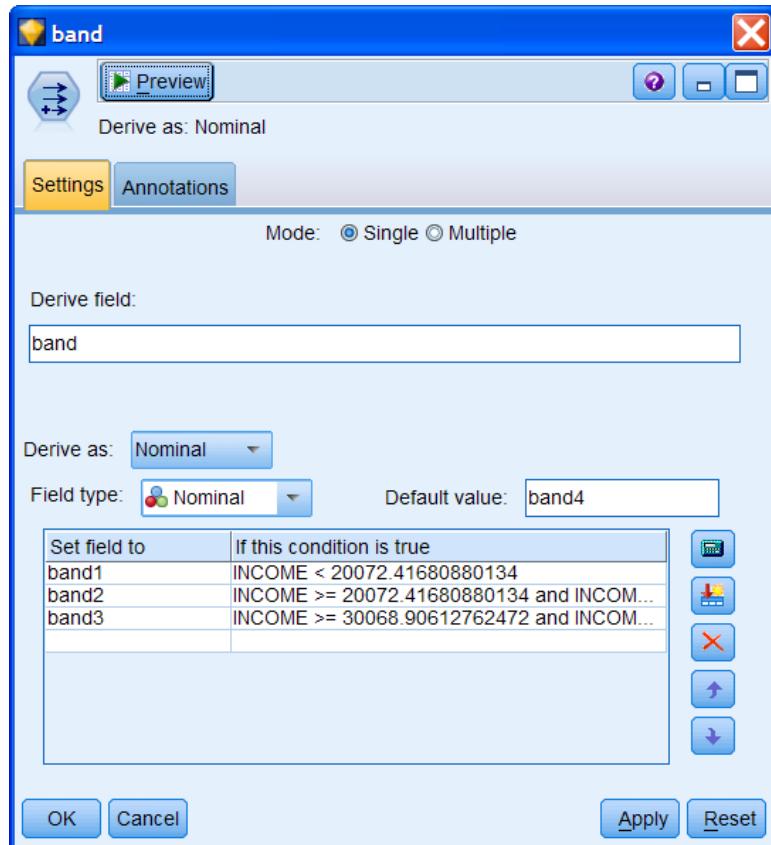
The lines define *bands* in the graph that divide the histogram into four groups. If you don't like the position of a line, you can click on it and drag it back and forth.

Click **Generate...Derive Node for Bands**

Click **File...Close** to close the Histogram window

A Derive node named **band** should appear in the top corner of the Stream Canvas.

Double-click on the new **Derive** node (named **band**)

Figure 6.24 Automatically Generated Derive Node Dialog

A Derive node of type Nominal has been generated. We can see the condition for a record to be classified in *band1* (roughly, income under 20,000). The conditions can be edited (for example, to make the cutoff exactly 20,000). The field name can also be changed by typing a new name in the *Derive field* text box (for example, INCOME CATEGORY in place of *band*).

Click **Cancel** to close the generated **Derive** node

Automatically Generating a Reclassify Node to Group Risk

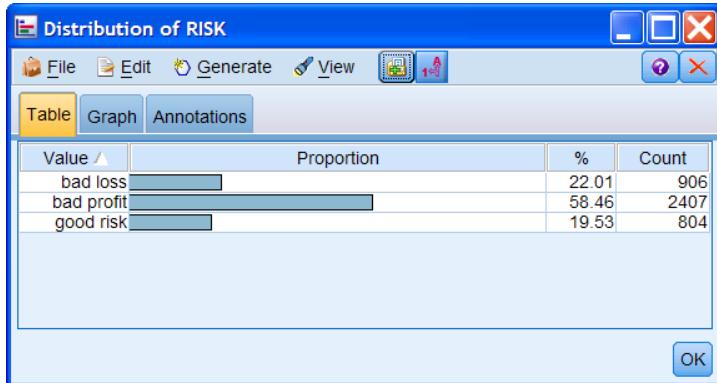
Earlier in the lesson we grouped *RISK* into two categories by grouping *bad loss* and *bad profit* and putting *good risk* in a group by itself. We did this with the Reclassify node. Another method to accomplish this is to use the Distribution node for a categorical field, then group the values in the table view of the Distribution output.

- Place a **Distribution** node from the Graphs palette to the right of the **Reclassify** node
- Connect the **Reclassify** node to the **Distribution** node
- Double-click the **Distribution** node
- Click the field list button and select **RISK** (not shown)
- Click the **Run** button

In the resulting table view, we can see rows corresponding to each category of *RISK*. What we now need to do is to group the values, which is done in two steps:

1. Select the categories to group using Ctrl-Click
2. Click Edit...Group from the menu to complete the operation.

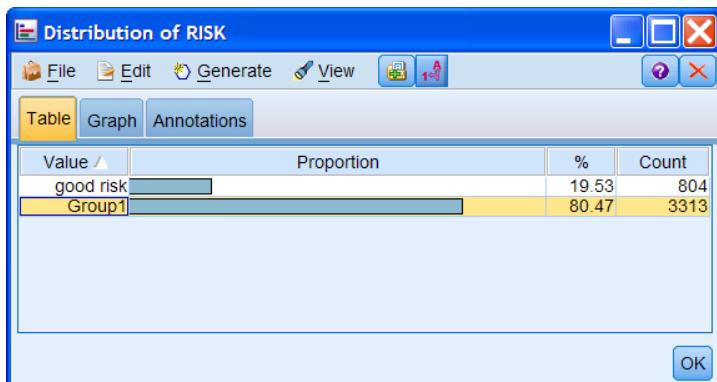
Figure 6.25 Distribution Table View for Risk



Hold down the **Control** key, and then click on the bars for **bad loss** and **bad profit**
Click **Edit...Group**

In Figure 6.26 we can observe that the two categories have been merged into Group1 (note that even if we weren't going to use the Generate menu, grouping categories in the Derive output for display purposes can be helpful).

Figure 6.26 Group Created for Risk

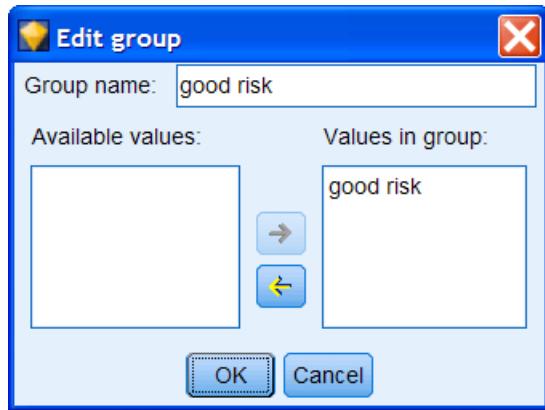


At this point, the Group1 category can be renamed. We also need to group the single *good risk* category for the Generate operation.

Click on the bar for **good risk**
Click **Edit...Group** (not shown)

Now we'll rename the Groups, which is done one at a time.

Click **Edit...Edit Group**
Change the Group name to **good**

Figure 6.27 Edit Group Dialog for Group2

Now we do the same for Group1.

- Click **OK**
- Click on the bar for **Group1**
- Right-click and choose **Edit Group**
- Change the Group name to **bad** (not shown)
- Click **OK**

Now we are finally ready to use the Generate menu to create a Reclassify node, which will be added to the upper left of the canvas.

- Click **Generate...Reclassify Node (groups)**
- Close the Distribution output browser window
- Double-click on the new **Reclassify** node named **(generated)**

The Reclassify dialog looks just like the one we constructed ourselves from scratch. Using the Generate menu in the Distribution output window provides an alternative method which you may prefer when there are a larger number of categories to collapse into a smaller number of groups.

Figure 6.28 Generated Reclassify Node for Risk

At this point we should save these data manipulation nodes in a PASW Modeler Stream file named *Data Preparation.str*.

Click **OK** to close the Reclassify dialog
Click **File...Save Stream As**
Type **Data Preparation.str** in the File Name box
Click **Save**

Summary

In this lesson you have been given an introduction to a number of methods of manipulating your data.

You should now be able to:

- Enter simple expressions using CLEM
- Use a Filter or Field Reorder node
- Create different types of Derive nodes
- Use the Reclassify node
- Use PASW Modeler to automatically generate Derive and Reclassify nodes

Exercises

In this exercise we some manipulation on the fields in the charity data.

1. Read data from *charity.sav* (Read names and labels; Read labels as data).
2. Connect a Filter node between the Statistics File node and the Table node. Edit the Filter node to filter out the field TITLE and rename the SEX field to GENDER (Remember: Case counts!). Rename any other field of your choice. Run the Table node to see the changes.
3. Connect a Field Reorder node between the Filter node and the Table node. Move the fields AGE, AGEBAND, and GENDER to the beginning of the file. Run the Table node to see the results of the Filter and Field Reorder nodes.
4. Remove the Filter node, Field Reorder node, and Table node.
5. Use Derive nodes to create two new fields, *PreSpend per Visit* and *PostSpend per Visit* as a ratio of the appropriate fields. Use the Expression Builder and check the formulas in the dialog boxes.
6. Use a Reclassify node to create a field named *Title_Gender*, which is coded Male or Female based on the values in the *Title* field.
7. Connect a Table node after the Reclassify node and run to see the results. Further check the results using the Matrix node (Output palette) or Distribution node (Graphs palette) for the *Title_Gender* field and the Statistics node (Output palette) for the derived fields. What do you note about the number of cases in the two derived fields? Can you explain the missing cases?
8. Create a histogram of the field called *TOTSPEND (Total Spend)*.
9. We are going to automatically generate a Derive node that creates a new field containing four bands of *TOTSPEND (Total Spend)*. Use the mouse to create three lines on the histogram where you would like to split the data. Generate the Derive node, using the Generate menu.
10. Connect the new Derive node to the Statistics File source node. Edit the Derive node and change the name of the new field to *Banded Total Spend*. Attach a Table node to the Derive node, or place the Derive node between the Statistics File node and the existing Table node, and run this section of the stream. View the new field in the data table. What graph might you run to check the results?
11. Save the stream under the name *ExerLesson6.str*.

Lesson 7: Looking for Relationships in Data

Objectives

- Introduce the Web and Matrix nodes to investigate relationships between categorical fields
- Illustrate the use of correlation within the Statistics node to investigate relationships between continuous fields
- Introduce the Means node to compare the means between independent groups to see if a significant difference exists
- Demonstrate how to request appropriate graphs with the Graphboard node
- Show how PASW Modeler graphs can be edited and modified

Data

In this lesson we will work with the credit risk data (*Risk.txt*) used in previous lessons. The data file contains information concerning the credit rating and financial position of 4117 individuals, along with basic demographic information, such as marital status and gender.

7.1 Introduction

The Data Understanding phase of the CRISP-DM model includes a search for interesting insights into the data, especially relationships between fields. Typically, you will examine the relationship between the target field (credit risk in this example), and the inputs or predictor fields. You want to see which fields are strongly associated with the target and which fields are not (one reason to do so is to reduce the number of fields as input for a modeling node, as some models—see Lesson 11—perform better with a somewhat limited number of predictors).

You should also investigate the *pattern* of the relationship between two fields. Is the association linear, or not? That is, does an increase in an input lead to a commensurate increase in the outcome? In addition, are there any outliers that are influencing the relationship, or records that don't fit the general pattern? Whatever you discover could influence the modeling technique(s) you choose.

Although the end of a data-mining project is normally the development of a powerful model, simple relationships can still be helpful in answering the questions that motivated a project. We may find that revenue is directly related to length of time as a customer, or those customers with a certain mobile phone plan and who have higher incomes are more likely to switch providers. Although these patterns are not substitutes for a full model, they can often be used along with a model.

With respect to our current dataset, simple questions may include:

- Is credit risk directly related to income?
- Do males differ from females with respect to credit risk?
- If an individual has a large number of current credit cards, loans, and store cards, does this mean that he is more likely, or less likely, to be a bad credit risk?

The methods used for examining relationships between fields depend on the measurement level of the fields in question. In the following sections we will introduce several techniques, some for studying relationships between categorical fields and one for investigating relationships between continuous fields. In addition, we will create graphs to examine these relationships.

7.2 Studying Relationships between Categorical Fields

In this section we introduce two nodes useful for examining whether categorical fields are related. The first is the Matrix node used to display the relationship between a pair of categorical fields. We then show how to visualize the relationship between two or more categorical fields with the graphical Web node.

7.3 Matrix Node: Relating Two Categorical Fields

The Matrix node performs crosstabulations of two categorical fields within the data (recall that we briefly used it in the previous lesson to check on a data transformation), and it shows how values of one field are related to those of a second field. A third, continuous, field can be included as an overlay field to see how it varies across the categorical pair relationship. The Matrix node is located in the Output palette and is thus a terminal node (since it creates output there is no need for data to pass through it downstream).

In this example we will use the Matrix node to see whether there are any relationships between the field we will try to predict, credit risk (*RISK*), and some of the other categorical fields within the data. We begin with investigating whether there is a difference between males and females with respect to their credit risk.

We will build our stream starting with the data source and Type nodes saved in the Riskdef.str stream file.

Click **File...Open Stream**, navigate to the **c:\Train\ModelerIntro** directory and double-click on **Riskdef.str**

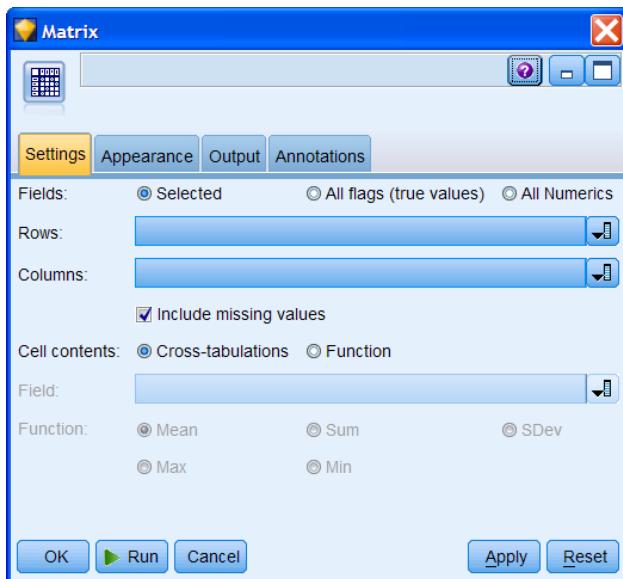
Place a **Matrix** node from the **Output** palette to the right of the **Type** node

Connect the **Type** node to the **Matrix** node

Double-click on the **Matrix** node

The default setting of the *Fields:* option (*Selected*) will display the fields selected in the *Rows* and *Columns* boxes (one field in each), which is what we want. Note that only one field for the *Rows* and one field for the *Columns* can be selected. Thus the Matrix node will produce one matrix at a time.

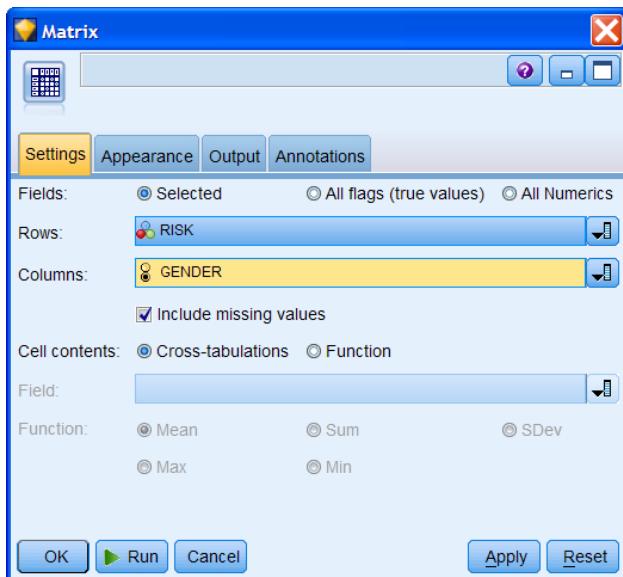
Less often used, the *All flags* option will create a symmetric matrix in which one row and one column appear for each Flag field and the cell values contain the co-occurrence counts of True values for the flag fields. Finally, the even more rarely used *All Numerics* option will produce a table with one row and one column for each continuous field, and the cells contain the sum of the products of each pair of fields (a cross-products table).

Figure 7.1 Matrix Node Dialog

The default option for Cell contents is *Cross-tabulations*. The alternative is a function applied to a third selected overlay continuous field; the *Sum*, *Mean*, *Min(imum)*, *Max(imum)*, or *Sdev (Standard Deviation)* of this field can be displayed for each of the cells in the matrix.

Within the Matrix dialog box:

Click the Fields list button in the **Rows:** list and select **RISK**
 Click the Fields list button in the **Columns:** list and select **GENDER**

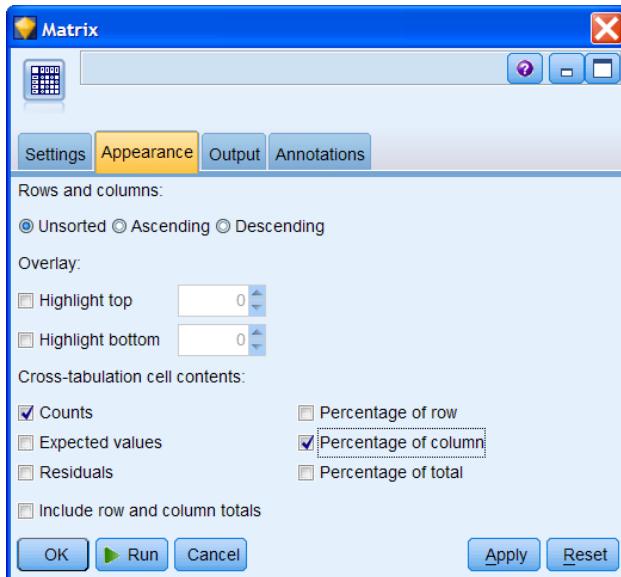
Figure 7.2 Matrix Node Fields Selected

The default table contains counts in the cells of the matrix and the results of a Chi-square test of independence. Alternatively, different percentages and also expected values and residuals can be

requested by using the *Appearance* tab. We will ask for column percentages in order to compare men and women with respect to their credit risk.

Click the **Appearance** tab
Select **Percentage of column**

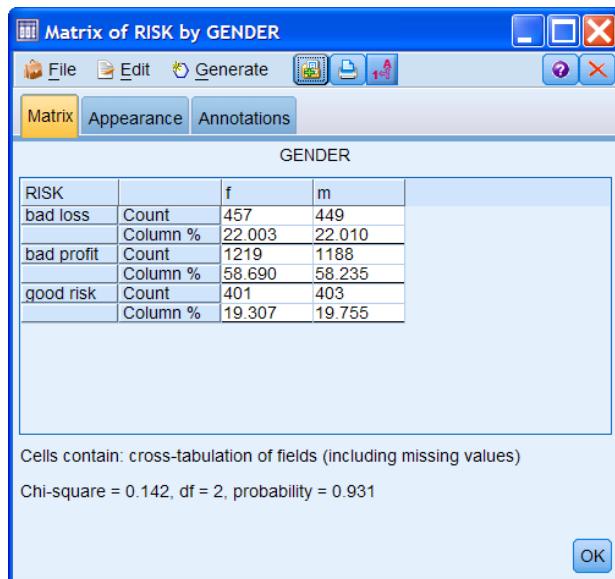
Figure 7.3 Matrix Node: Appearance Tab



Cells with the highest or lowest values in the table can be highlighted by entering the number of cells in the *Highlight top/bottom* options. This feature can be useful when percentages are displayed. As with most output nodes, the Output tab can be selected for setting the output destination; by default, output is sent to the screen.

Click the **Run** button

Examining the matrix table, there appear to be no differences between the two genders in their distribution across credit risk categories. For instance, 22.003% of the women are categorized as bad loss, while 22.010% of the men belong to this category. The Chi-square test probability value of 0.931 further confirms this conclusion that *GENDER* is independent from type of *RISK* in this data file.

Figure 7.4 Matrix Table of Gender and Credit Risk

Does our finding of no relationship between gender and credit risk imply that gender should not be used for modeling? The answer to that question is a bit more complicated than you might expect. There can be *interaction effects* between predictors such that, while gender itself may not be a good predictor, the effect of another predictor may be different depending on whether a person is male or female. In that case, leaving gender out of a model may reduce its accuracy.

Of course, without doing lots of additional data exploration by looking at three-way relationships you won't be able to determine this before modeling begins. Still, the possibility of an interaction, and the ability of many models to use interactions naturally (especially decision trees and neural nets) argues for not automatically throwing out a field such as *GENDER*.

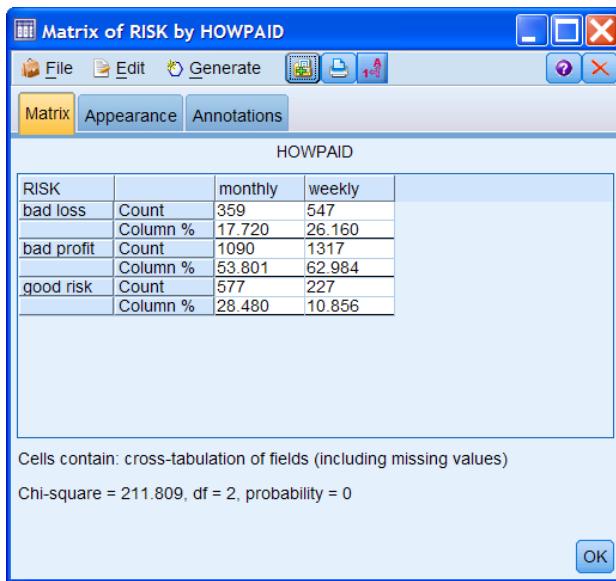
To repeat this exercise for a second categorical field in relation to credit risk we can either edit the existing Matrix node or create a new one. We'll edit the existing node and next investigate the relationship between credit risk and how often an individual is paid (monthly or weekly).

Close the Matrix output window

Double-click on the **Matrix** node, and then click the **Settings** tab

Click the Fields list button in the **Columns:** list and select **HOWPAID**

Click **Run**

Figure 7.5 Matrix Table of Salary Payment Schedule and Credit Risk

The matrix table suggests that individuals paid monthly are more likely to be good credit risks than those paid weekly (28.5% against 10.9%). The Chi-square test (probability = 0, rounded off) provides statistical evidence that credit risk is related to how often a person is paid.

At this point in the data-mining process, business understanding becomes pertinent. The difference in good risk percentage between the two groups certainly seems large enough to be important, but a business user of these data or of the final model should be consulted to ascertain whether this difference is indeed substantively or practically large enough. The best models are not developed in isolation by the analyst but instead with continuing interaction with those who understand the data and the business questions being addressed from a practical and operational perspective.

Close the Matrix output window

Note

One useful feature is that the summaries requested in the Appearance tab can be changed after the matrix output is generated. Different summaries can be obtained directly from the Matrix node output, without requiring re-execution of the Matrix node. Thus, you could easily view row percentages within the matrix output displayed in the last two figures.

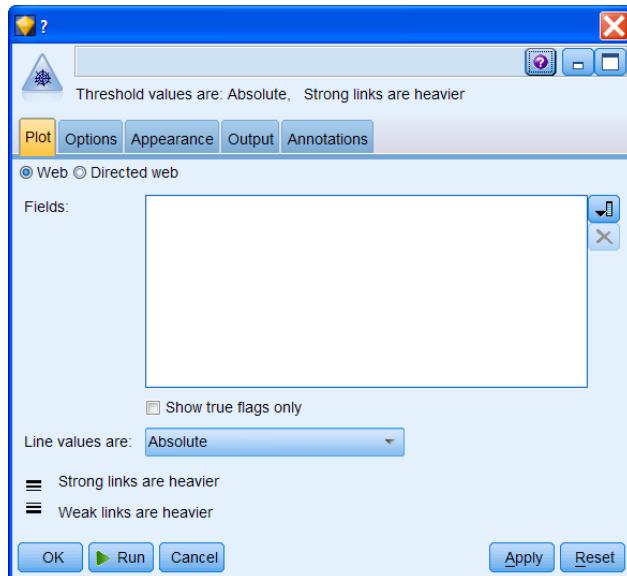
7.4 The Web Node

The Web node, located in the Graphs palette, can be used to show the strength of connection between values of two or more categorical fields. A web plot consists of points representing the values of the selected categorical fields. These points are connected with lines whose thickness depicts the strength of the connections between the values (number of records with both categorical values). Lines thus represent the cells of a matrix table.

Thin lines represent weak connections while heavy, solid lines represent strong connections. Intermediate strength connections are drawn as normal lines. Web displays are interactive and it is possible to vary the threshold settings (what defines a weak or strong connection), hide irrelevant fields, modify the layout, and generate nodes.

Place a **Web** node from the **Graphs** palette near the **Type** node
 Connect the **Type** node to the **Web** node
 Double-click the **Web** node

Figure 7.6 Web Node Dialog

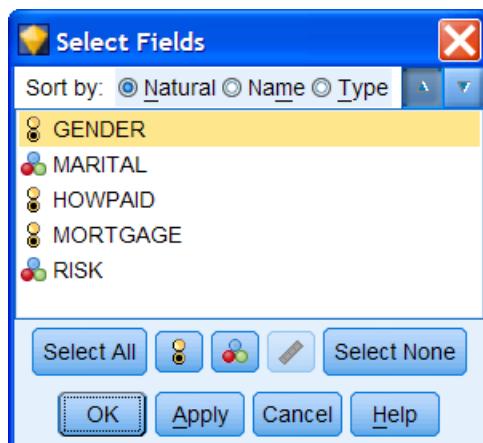


Two types of plots are available. In a web plot the relations between all selected categorical fields are shown, while in a directed web plot only relations involving a specified target field are displayed.

The *Show true flags only* check box allows only the True response for flag fields (as defined in the Type node or Types tab of a source node) to be shown in a web plot. This is a very useful feature in displaying the relationships among many products either bought or not (yes or no) by a customer, as we will see in a later lesson.

Click the **Field List** button

Figure 7.7 Select Fields Dialog



Only categorical fields (flags, nominal fields, ordinal fields) are eligible for a web plot. All fields, all flag fields, or all nominal fields can be chosen at once by clicking the respective button. In addition

Select None can be chosen to deselect fields. The *Select all continuous fields in list* button is inactive, since continuous fields are not appropriate for web plots.

In this example we will investigate the relationship between credit risk and the two other categorical fields: marital status and whether the individual has a mortgage or not.

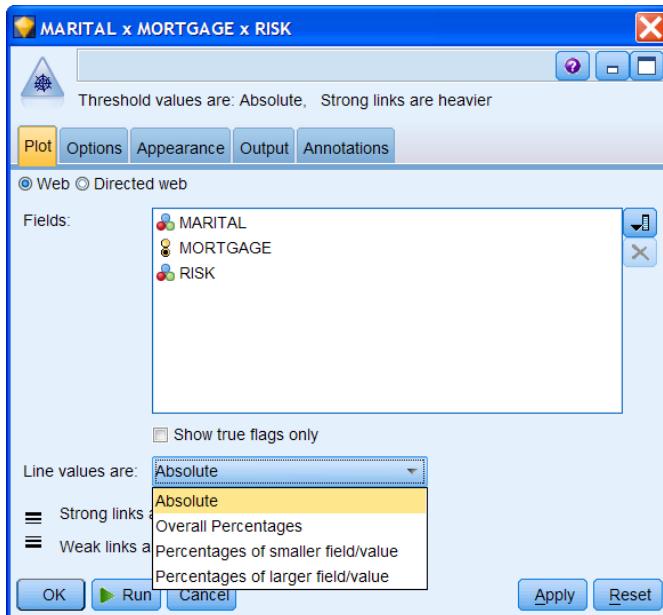
Select **MARITAL**, **MORTGAGE** and **RISK** (Use Ctrl-Click to select multiple fields)
Click **OK** to return to the Web dialog box

The web plot will show strong, normal, and weak links (or continuously varying links). What constitutes a strong or weak link is defined by the threshold value. Several types of thresholds are available.

Click the **Line values are** drop-down list

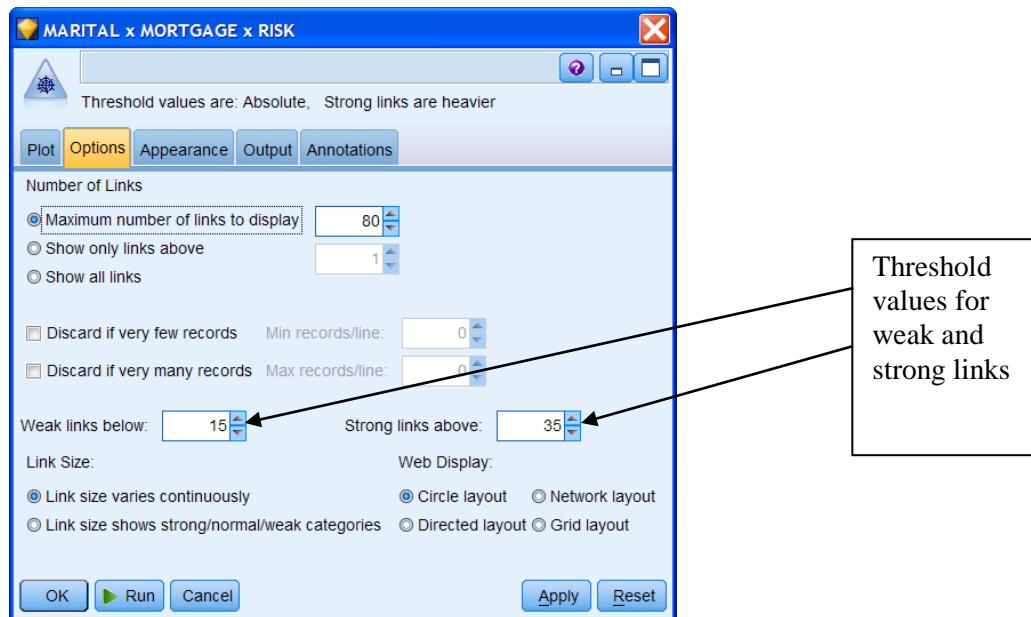
Line values can represent counts (*Absolute*), percentages of the overall data (*Overall Percentages*), or percentages of either the smaller or larger field/value. For example, if 100 records have value X for one field, 10 records have value Y for a second field, and there are 7 records containing both values, the connection involves 70% of the smaller field and 7% of the larger field.

Figure 7.8 Threshold Types



The threshold values themselves are set under the Options tab.

Click the **Options** tab

Figure 7.9 Setting Threshold Values

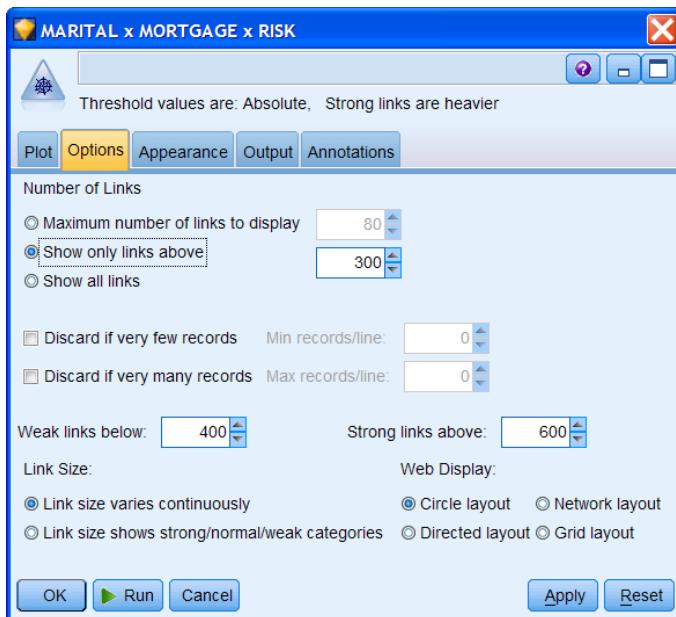
Threshold
values for
weak and
strong links

The number of links in the web plot is controlled by 1) choosing a maximum number of links; 2) displaying links only above a specified value; or 3) displaying all links. The *Discard* options allow you to ignore connections supported by too few records.

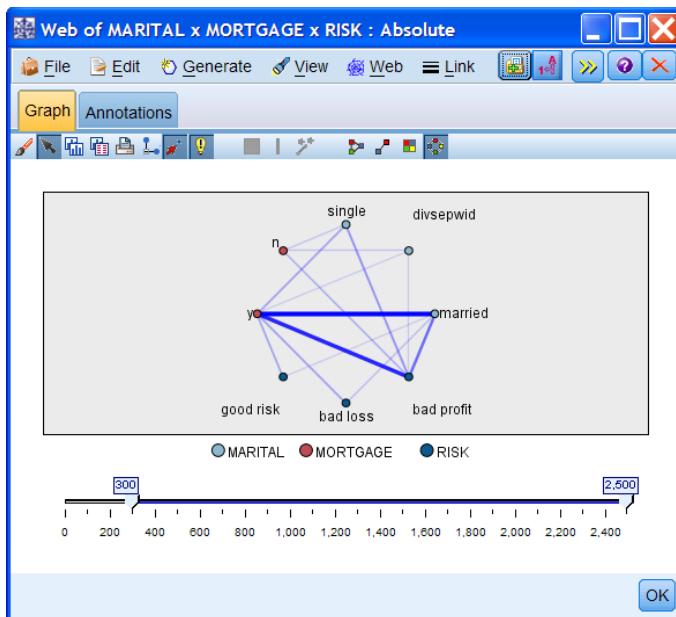
The *Link Size* options control the size of links. The *Link size varies continuously* option will display a range of link sizes reflecting the variation in connection strengths based on actual data values. The alternative, *Link size shows strong/normal/weak categories*, will display three strengths of connections—strong, normal, and weak. The cut-off points for these categories can be specified above as well as in the final graph.

We have over 4000 records and we will set the thresholds initially at 300, 400 and 600 records, respectively. We will see how this can be adjusted once the plot has been produced.

Click **Show only links above** option button
 Type **300** in the **Show only links above** box
 Type **600** in the **Strong links above:** box
 Type **400** in the **Weak links below:** box

Figure 7.10 Options for the Web Plot**Click Run**

Drag the lower corner of the resulting Web graph window to enlarge it

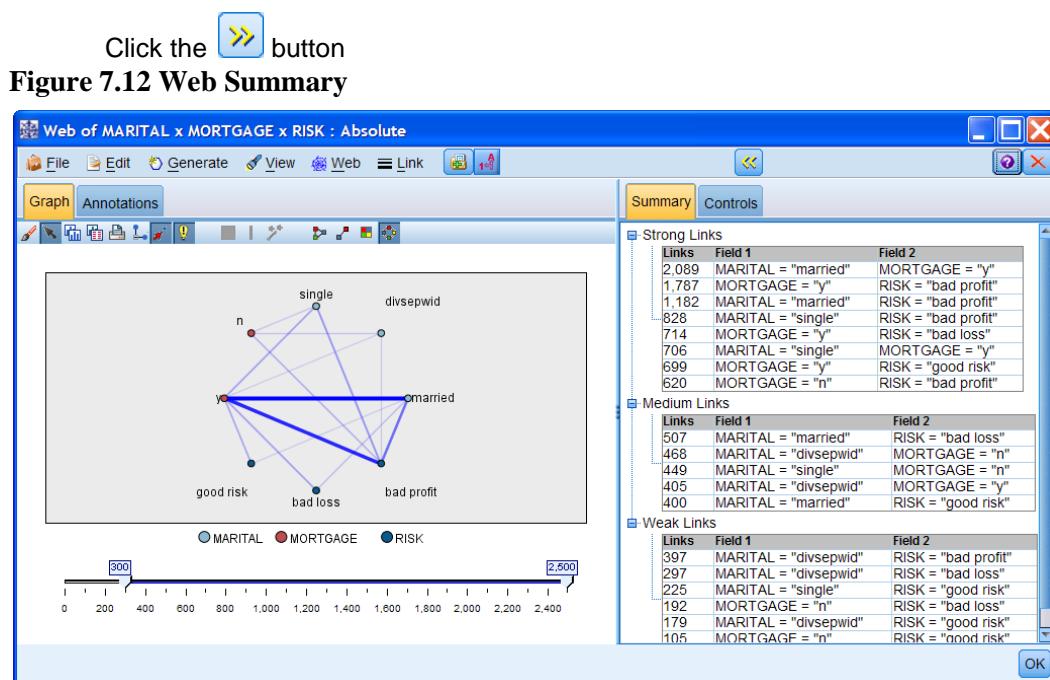
Click the **View** in the toolbarSelect **General** and **Interactions** in the dropdown list (if necessary)**Figure 7.11 Web Plot of Marital Status, Having a Mortgage, and Credit Risk**

Remember, the thickness of the lines varies continuously, reflecting how strong a link is. At first glance, the link between *married* and *y(es)* (*MORTGAGE*) is the strongest. Looking at the different categories of *RISK*, *bad profit* is strongly connected to *y* as well, and (although somewhat less strongly) to *married*. Apparently, the *bad profit* group is associated with married persons and with

those who have a mortgage (based on the pairwise counts), although not necessarily married persons *with* a mortgage (the web plot does not display three-way relationships).

Besides a visual inspection of the thickness of the link to look for weak and strong links, we can ask for the associated counts in several ways. One method is to position the cursor over a link; a pop-up will display the number of records having that particular combination of values. To do this you first have to activate tooltips by clicking on that button  in the toolbar. The disadvantage of this method is that we have to check the links one-by-one.

An alternative is to ask for a web output summary, using the web summary button  in the Toolbar.

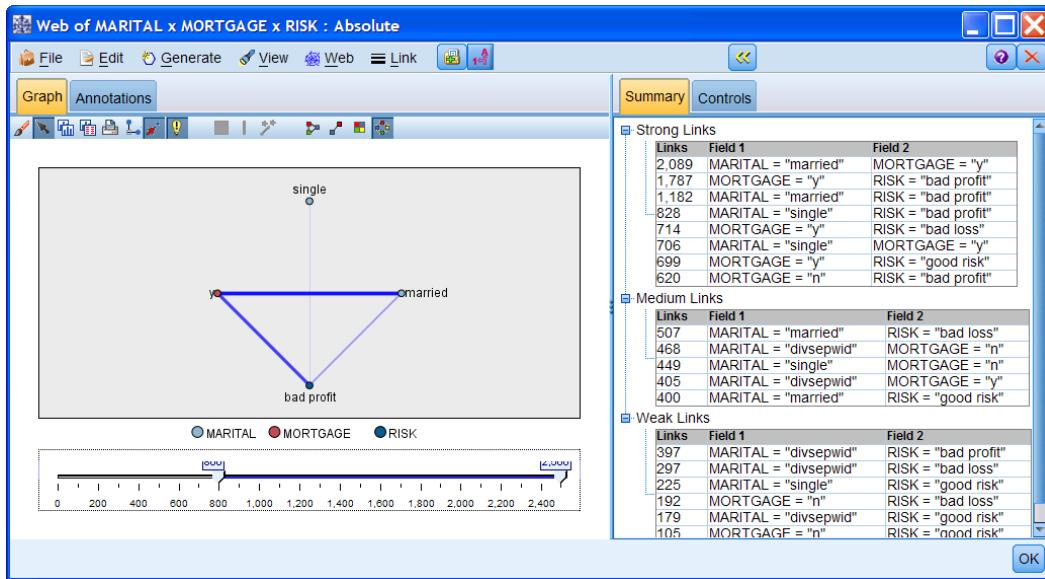


From the summary we see that there are several weak links, as we defined them (links with counts less than 400). Most of these are not displayed because the plot only shows links with at least a count of 300. Having seen the counts, we will change the web plot in order to see the strong links more clearly. PASW Modeler allows you to do this in several ways.

First, we might want to use the slider control, located in the bottom of the window. In the web graph window the slider starts at value 300 and ends at 2500. The Links slider allows you to control which links are displayed, and you can use it to set the minimum or maximum displayed link size.

Use the **Links slider control** to discard links weaker than about **800** by moving the left slider to the right

Now the Web graph shows only four links. In this way, you can increase or decrease the complexity of the display.

Figure 7.13 Web Plot: Setting Minimum Link Size Interactively

Besides setting the minimum link size interactively, you can also re-specify what constitute weak and strong links:

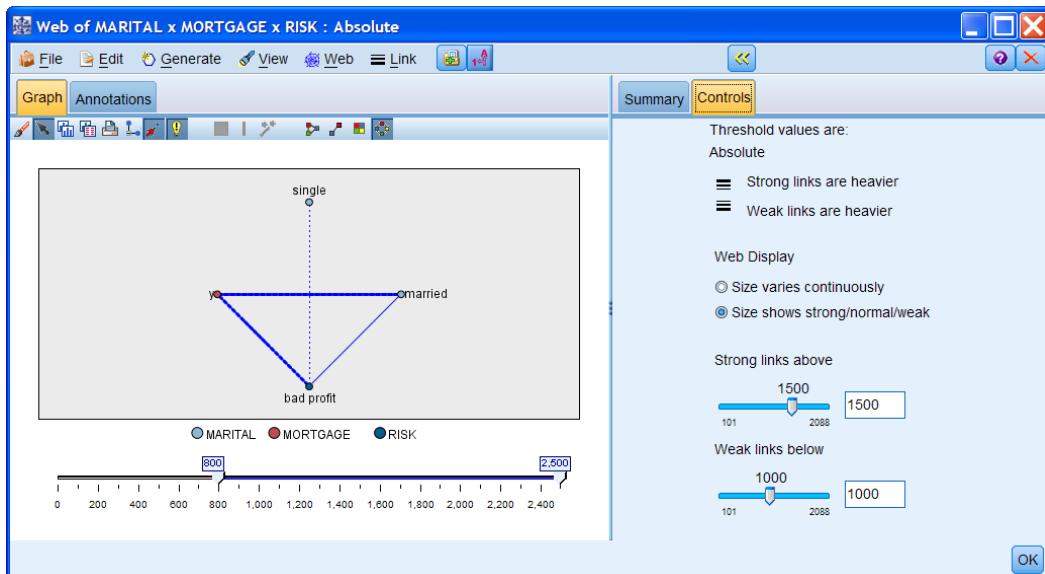
Click the **Controls** tab in the Web Summary output

Click the option button for **Size shows strong/normal/weak**

Use the slider control or text box to set the value for **strong links above** to **1500** and the value for **weak links below** to **1000**

Hit **Enter**

Move the Links slider to set the minimum value **800**

Figure 7.14 Web plot: Setting Strong and Weak Link Size Interactively

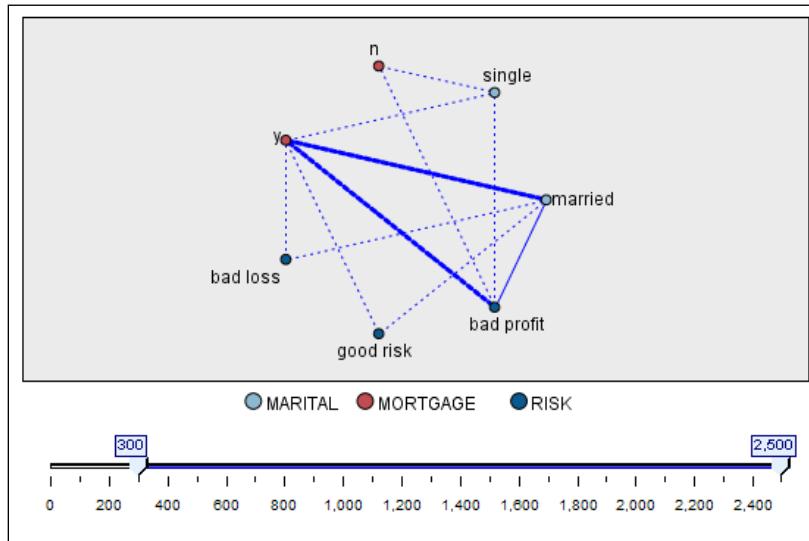
The displayed lines don't change, only their characteristics. Now the connection between married and bad profit is a medium link, and the connection between single and bad profit is a weak link.

In this way, with the combination of the slider to display links, and the controls to define the type of link, you can modify the web plot to investigate the relationships between categories.

To facilitate a better interpretation of the web plot, we should mention the options of hiding categories or moving categories. For instance, in our plot *divsepwid* was not connected to any of the three *RISK* categories, so we can hide this category.

- Move the Links slider to set the minimum value 300
- Right-click the point representing **divsepwid**
- Click **Hide and Replan** from the context menu

Figure 7.15 Hiding Divorced, Separate, and Widowed Category

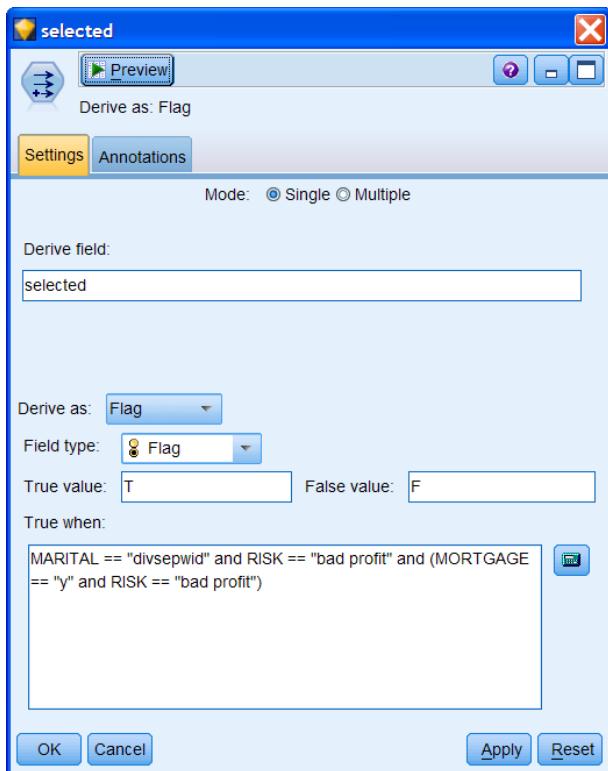


To make a web graph more readable, you can move a category by simply dragging it to a new location in the plot with the left mouse button (not shown).

As in other output windows we have the opportunity to generate a Select or Derive node. For example, suppose that we want to create a flag field indicating whether an individual is married, has a mortgage and belongs to the bad profit category:

- Click the link connecting **married** and **y** (the link will turn red if selected)
- Hold down the Control key and click the link connecting **bad profit** and **y** (the link will turn red if selected), not shown
- Click **Generate...Derive Node ("And")**
- Close the Web plot window
- Double-click the new Derive node in the Canvas

As before, a Derive node will be generated and will appear in the Stream Canvas. We could have created this node ourselves but it would have taken some work. This node can be edited or included in the stream as is.

Figure 7.16 Derive Node Created from Web Graph

Close the Derive dialog

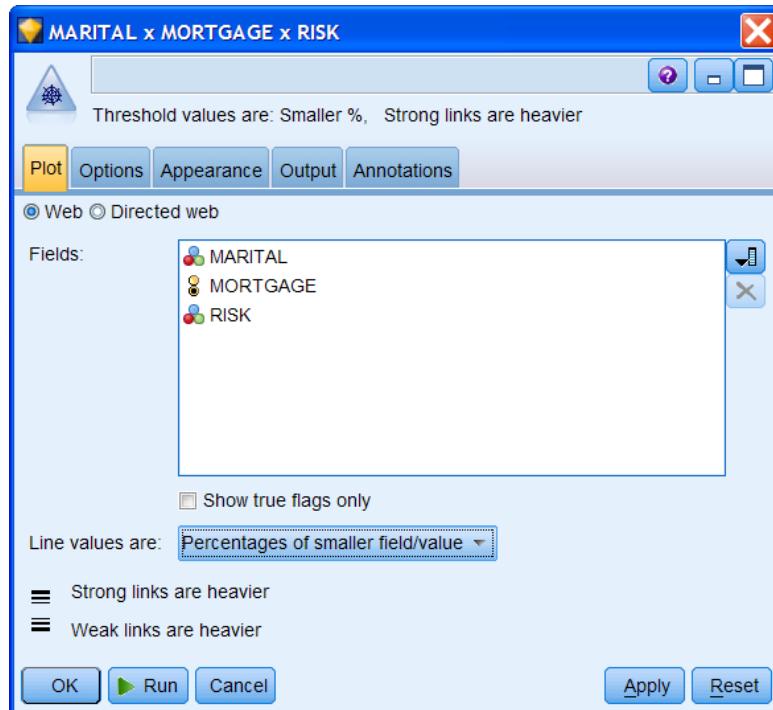
Using Percentages in Web Graphs

The web graph we constructed is based on the absolute number of records that jointly share a pair of values. Categories with more records tend, all things being equal, to have stronger links in such a web graph (for example, those who are married outnumber by far those in the other two categories). But for evaluating relationships between fields in a crosstabulation we invariably use percentages, which standardize for row and column totals. Let's see how different this same web graph looks if we change the line values basis to percentages.

Double-click the **Web** node to edit it

Click on the **Plot** tab

Click on the drop-down for Line values are: and select **Percentages of smaller field/value**

Figure 7.17 Specification to Use Percentages of Smaller Field

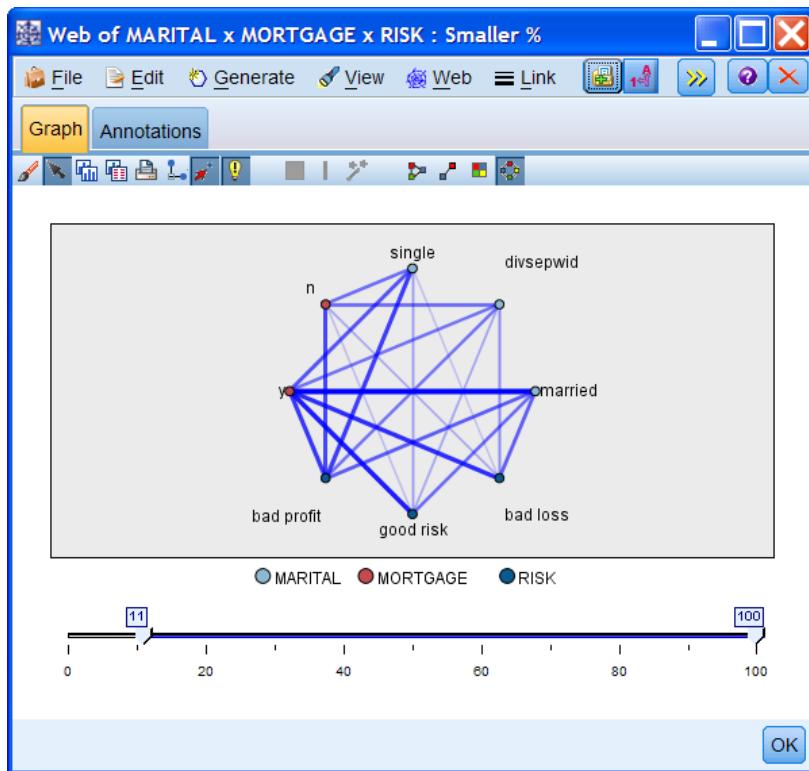
This selection will base percentages on whichever category has the smaller total number of cases in the two categories that define a connection line. For example, if we are examining the relationship between good risk and people who are married, if the good risk category has fewer total records compared to the married category, percentages will be based on that category.

Clearly, this can make it tricky to evaluate a web graph based on percentages, since we can't select the standard row or column percentages that are available in a Matrix node, and the field used as a basis for the percentages can and probably will vary from one category pair to another. Nevertheless, as we will see, it is useful to look at multiple bases for the line values.

We need to make changes in the weak and strong link values to take account of the fact we are using percentages now, as these are not completely reset.

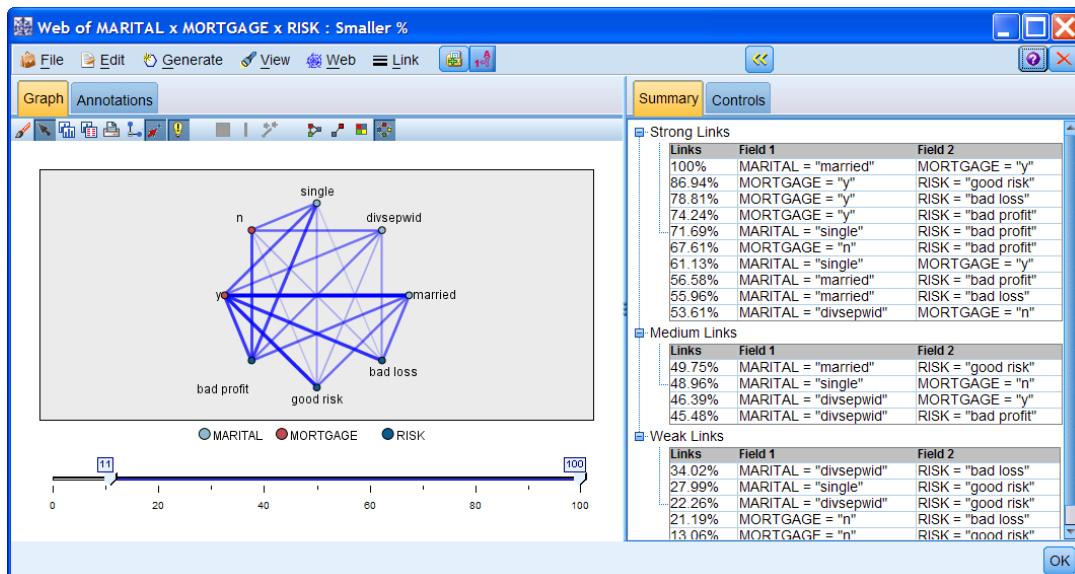
- Click the **Options** tab
- Type **10** in the **Show only links above** box
- Type **35** in the **Weak links below:** box
- Type **50** in the **Strong links above:** box (not shown)
- Click **Run**

Compare the resulting web graph in Figure 7.18 to the previous one (see Figure 7.). Would we still conclude that the bad profit category is related to married persons and those who have a mortgage? The answer is clearly no. Now, those with a mortgage are more associated with the good risk category, and married people have about the same level of association with the three categories of *RISK*. This is a very different view of the relationships between these three fields.

Figure 7.18 Web Graph Based on Percentage of Smaller Field

To take this one step further, we can look at the actual percentages.

Click button

Figure 7.19 Web Summary

We see now that the percentage differences are not quite as great as the web plot might indicate. Thus, for those with a mortgage, the percentages associated with the three *RISK* categories vary only from 86.9 to 74.2, and all are listed as Strong links, given our criterion.

The important lesson is that you should use multiple views to investigate a relationship when using a Web graph.

Close the Web Graph window

We will now explore relationships between continuous fields in the data.

7.5 Correlations between Continuous Fields

When investigating relationships between continuous fields, a linear correlation is commonly used. It measures the extent to which two continuous fields are linearly associated. The correlation value ranges from -1 to $+1$, where $+1$ represents a perfect positive linear relationship (as one field increases the other field also increases at a constant rate) and -1 represents a perfect negative relationship (as one field increases the other decreases at a constant rate). A value of zero represents no linear relationship between the two fields.

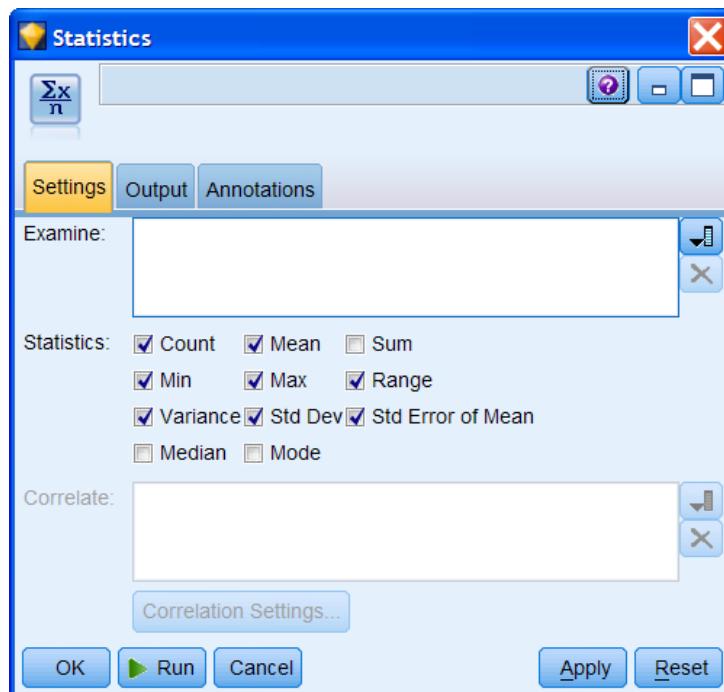
Earlier we used the Data Audit node to produce basic descriptive statistics for continuous fields. The Statistics node, which produces summary statistics but no graphs, can provide correlations between fields.

Place a **Statistics** node from the Output palette near the **Type** node in the Stream Canvas

Connect the **Type** node to the **Statistics** node

Double-click on the **Statistics** node

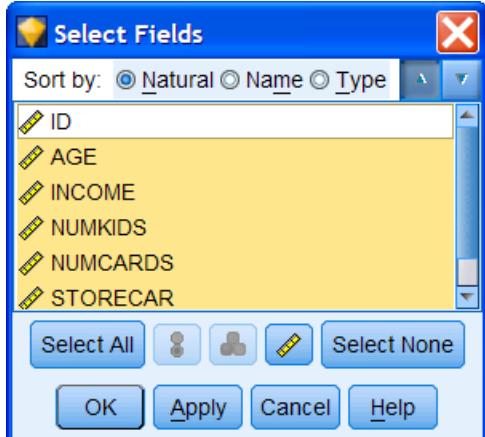
Figure 7.20 Statistics Node



In this example we will ask for summary statistics for all continuous fields, and correlations between them, excluding ID.

Click the **Examine:** field list button 
Click  to select all continuous fields
Ctrl-click **ID** to deselect it

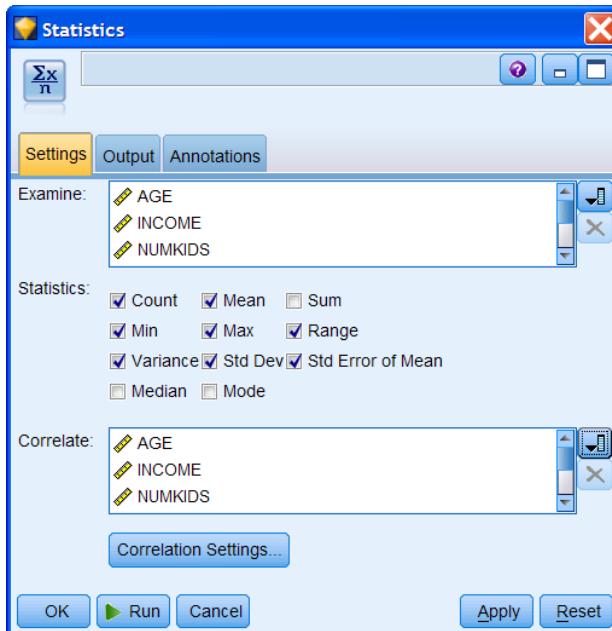
Figure 7.21 Selecting Fields to Examine



Click **OK**

Click the **Correlate:** field list button 
Click  to select all continuous fields
Ctrl-click **ID** to deselect it, and then click **OK**

Figure 7.22 Statistics Dialog: Requesting Correlations

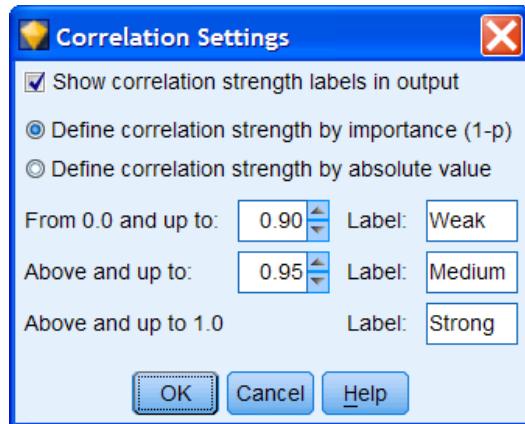


Correlations between the fields selected in the *Correlate* list and those selected in the *Examine* list will be calculated. Similar to the web plot, labels will be attached to weak and strong relationships.

With the *Correlation Settings* dialog you define what you consider to be weak and strong relationships.

Click the **Correlation Settings** button

Figure 7.23 Defining Correlation Strength Labels



The Correlation Settings dialog has two options for defining and labeling the strength of the correlation. Here, the labels are based on importance, which is calculated by subtracting the significance value of the correlation from one. The closer this value is to one, the greater the likelihood that the two fields are related (i.e., not independent). This doesn't tell you how strong the relationship is.

The alternative form of labeling is based on the absolute value of the Pearson correlation. By default, correlations up to .33 (in absolute value) are defined as weak, between .33 and .66 as medium and above .66 as strong. These default values can be changed in the respective text boxes. It is generally recommended that you label correlations based on importance rather than by absolute value since the first decision about a correlation is whether it is significant (important). For instance, a correlation of .66 may be highly significant in one dataset but not significant at all in another. On the other hand, in data-mining projects we often use many thousands, even millions, of records. In that case, almost all correlations will be significant and show an importance of 1. So the larger the sample size, the more you should rely on the actual value of the correlation. The smaller the sample size, look first at the importance, then at the correlation.

Click **OK**

Click **Run**

Figure 7.24 Statistics Output with Correlations

The screenshot shows the 'Statistics' dialog box for six selected fields. The 'Statistics' tab is active. The output is organized by field:

- AGE:**
 - Statistics:** Descriptive statistics for AGE.
 - Pearson Correlations:** Correlation matrix between AGE and other fields.
- INCOME:**
 - Statistics:** Descriptive statistics for INCOME.

AGE Statistics:

	Count	4117
Mean	31.820	
Min	18	
Max	50	
Range	32	
Variance	97.550	
Standard Deviation	9.877	
Standard Error of Mean	0.154	

AGE Pearson Correlations:

	INCOME	0.295	Strong
INCOME	NUMKIDS	0.435	Strong
INCOME	NUMCARDS	0.600	Strong
INCOME	STORECAR	0.478	Strong
INCOME	LOANS	0.258	Strong

INCOME Statistics:

	Count	4117
Mean	25580.212	
Min	15005	
Max	59944	
Range	44939	
Variance	76857960.835	
Standard Deviation	8766.867	
Standard Error of Mean	136.632	

Along with the descriptive statistics, the report displays the correlation between each pair of selected fields (each field selected in the Correlate field list with each field selected in the Examine field list). The report also suggests how to interpret the strength of the linear association between the fields, according to the definitions set with the Correlation Settings button.

Scrolling through the output, we find strong positive linear relationships between number of loans, children, store cards and credit cards.

One limiting aspect of using correlations is that they only give an indication of linear relationships between the continuous fields. There may be no linear relationship between two fields but there still may be a relationship of another functional form. For example, the correlation between age and income is very weak but experience suggests that these two fields are related to one another in a curvilinear fashion. Income rises with age but then eventually declines at around retirement age. Data mining often consists of finding such nonlinear relationships when developing a model.

This implies that a low correlation between two fields is not a reason to drop a field as a predictor. This is analogous to what we discussed above for categorical fields, except that there we emphasized possible interaction effects. And, in fact, one could have an interaction effect between a categorical and continuous field in their joint effect on an outcome.

Hint

All of these cautions and caveats may be a bit confusing to the novice data miner. It might seem that there is never an instance during data exploration and understanding when we would discard a field based on its relationship to a target, but that isn't the case. Often, business understanding, in conjunction with data exploration, may lead you to drop fields. For example, we saw above that gender was unrelated to credit risk. If those with a business and/or data understanding tell you that gender has never been an important predictor in the past, and/or that they prefer not to develop models that include gender, then those practical considerations, plus the data exploration results, would argue strongly for dropping gender as a predictor.

Close the Statistics output window

7.6 Means Node: Analyzing the Relationship between Continuous and Categorical fields

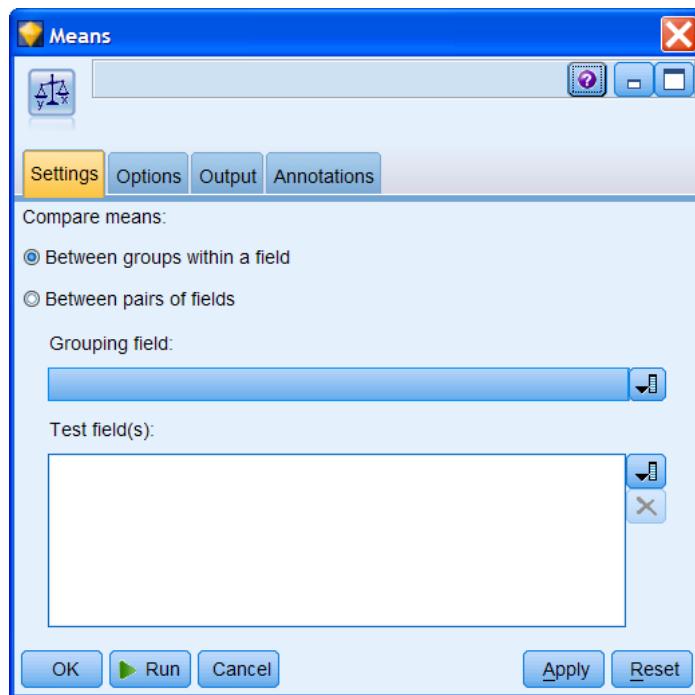
The Means Node compares the mean differences between independent groups or between pairs of related fields. For example, prior to modeling it would be useful to investigate whether there are significant differences between the three credit risk categories based on income, number of credit cards, number of children, etc. The node calculates a one-way Analysis of Variance based on the selected fields. In cases where there are only two groups, the results are essentially the same as an independent sample t test.

Place a **Means** node from the Output palette near the **Type** node on the Stream Canvas

Connect the **Type** node to the **Means** node

Double-click on the **Means** node to edit it

Figure 7.25 Means Node



By default, the option *Between groups within a field* compares the means for independent groups within a field. The *Between pairs of fields* option is used to compare mean values of two related

fields. For example, you could compare the average level of telephone usage from the same customers before and after a price reduction. When this option is used, the node calculates a paired-sample t test on all pairs of fields you define.

The Options tab allows you to set threshold p values used to label results as important, marginal and unimportant (or other labels you prefer). Level of importance is equal to $1-p$ value, as with the correlations. By default, an importance value below 0.90 is considered Unimportant, between 0.90 and 0.95 is labeled Marginal, and values greater than 0.95 are labeled Important. You can change these settings if you wish.

In this example, we will test for mean differences between the three categories of RISK and each of the continuous fields.

Within the Means dialog box:

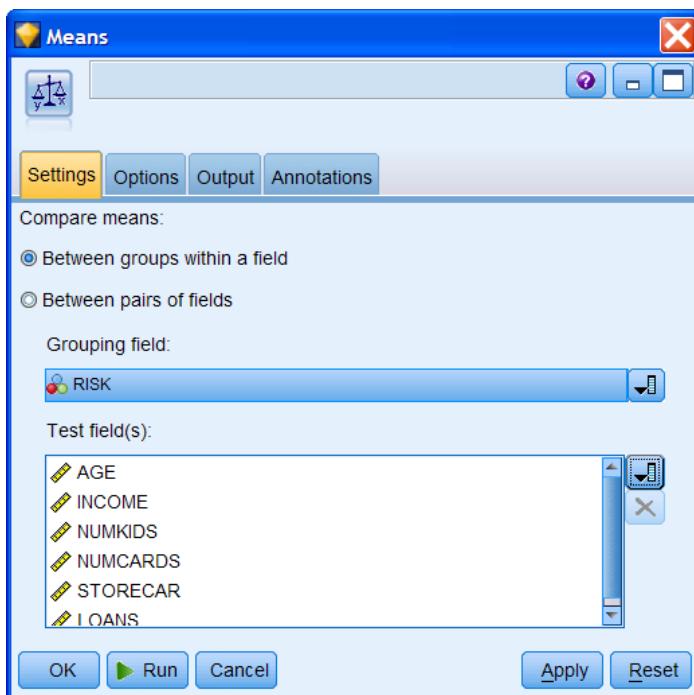
Click the Fields list button in the **Grouping field:** list and select **RISK**

Click the Fields list button in the **Test field(s):** list button

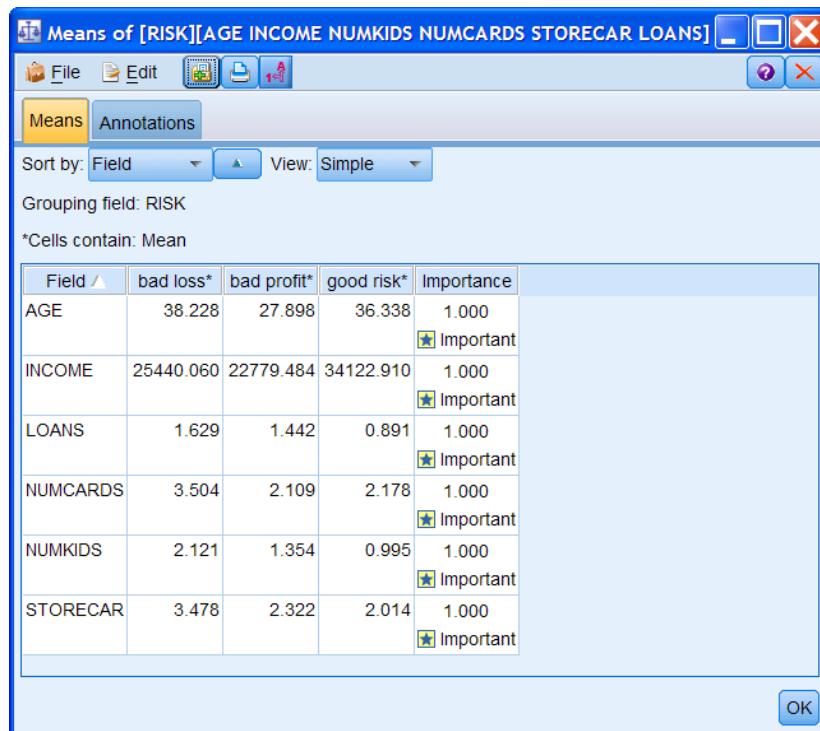
Click  to select all continuous fields

Ctrl-click **ID** to deselect it

Figure 7.26 Completed Means Dialog



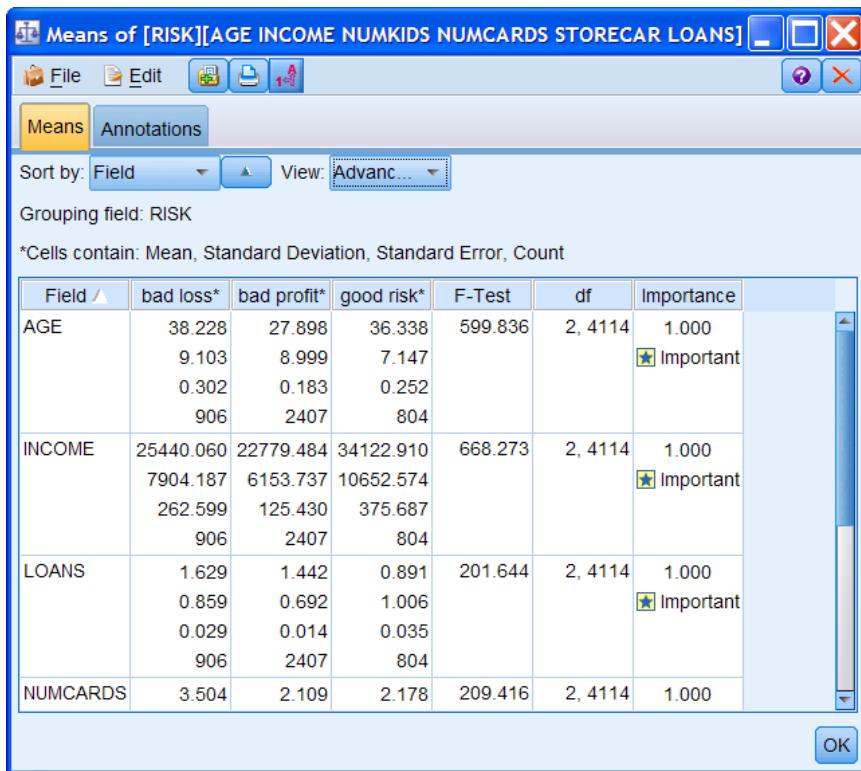
Click **Run**

Figure 7.27 Means Node Output – Simple View

The cell values under each *RISK* category are the group means within the field in the left column. For example, the mean age for people in the bad loss category is 38.2 years, for those in the bad profit category is 27.9 years, and for those in the good risk category is 36.3 years. The Importance column indicates that the group means differ significantly on all the fields. In this instance, the probability values for all the fields are 0.0 (rounded off). It would appear that all these fields should be included in a model predicting *RISK*.

The *Sort by:* option allows you to sort the results by column heading. The *View* option allows you to specify the level of detail you want displayed in the results. The *Simple* view displays only the cell means and importance values in the output. The *Advanced* view also includes the Standard Deviation, Standard Error, Count, the F-Test value, and degrees of freedom.

Click the **View:** drop-down arrow and select **Advanced**

Figure 7.28 Means Node Output – Advanced View

Here we see that while the groups differ significantly on all the fields, a comparison of the F-Test values indicates that *INCOME* and *AGE* differences are more pronounced than any other fields (the larger the F the more the differences). Regardless of that statistical evaluation, what are most important for any assessment are often the actual and relative differences between categories. Thus, the absolute difference in the mean for *INCOME* between the highest and lowest category (34,123 to 22,779) is about 11,350, while the relative difference between the highest and lowest categories is about 50%.

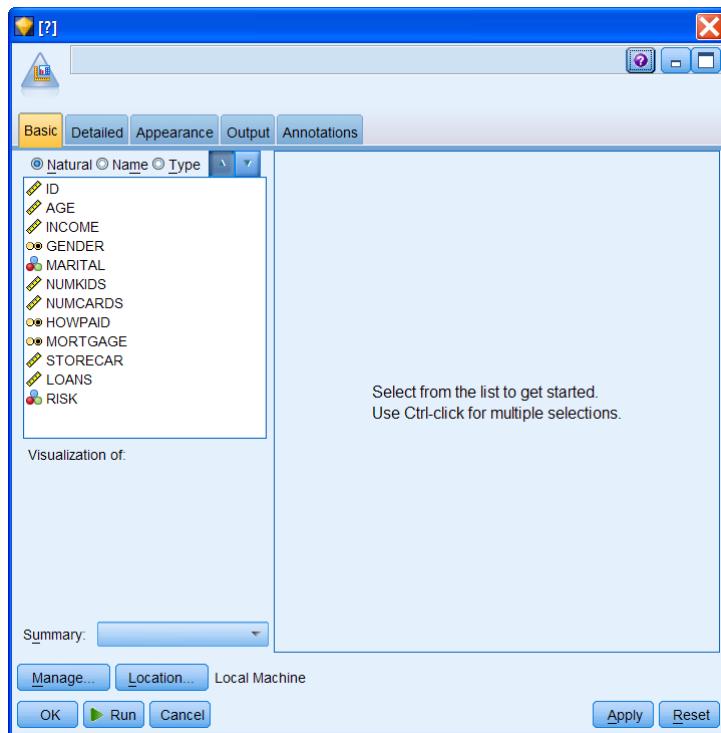
[Close the Means Output browser window](#)

7.7 Using the Graphboard Node to Examine Relationships

The Graphboard node allows you to choose from many different graphs types (bar charts, pie charts, histograms, scatterplots, heatmaps, bubble plots, etc.) in one single node. The node can help you select the appropriate graph, and it contains numerous graphs that are not available in the separate graph nodes in PASW Modeler. You select the fields you wish to explore, and the node then presents you with a choice of graph types that work for your data. The node automatically filters out any graph types that would not work with the field choices. You can also define detailed, or more advanced, graph options in the Detailed tab, or just use this tab to create a graph from scratch.

We'll demonstrate the Graphboard capabilities by examining the relationship between both a categorical (*GENDER*) and a continuous (*STORECAR*) field with *RISK*.

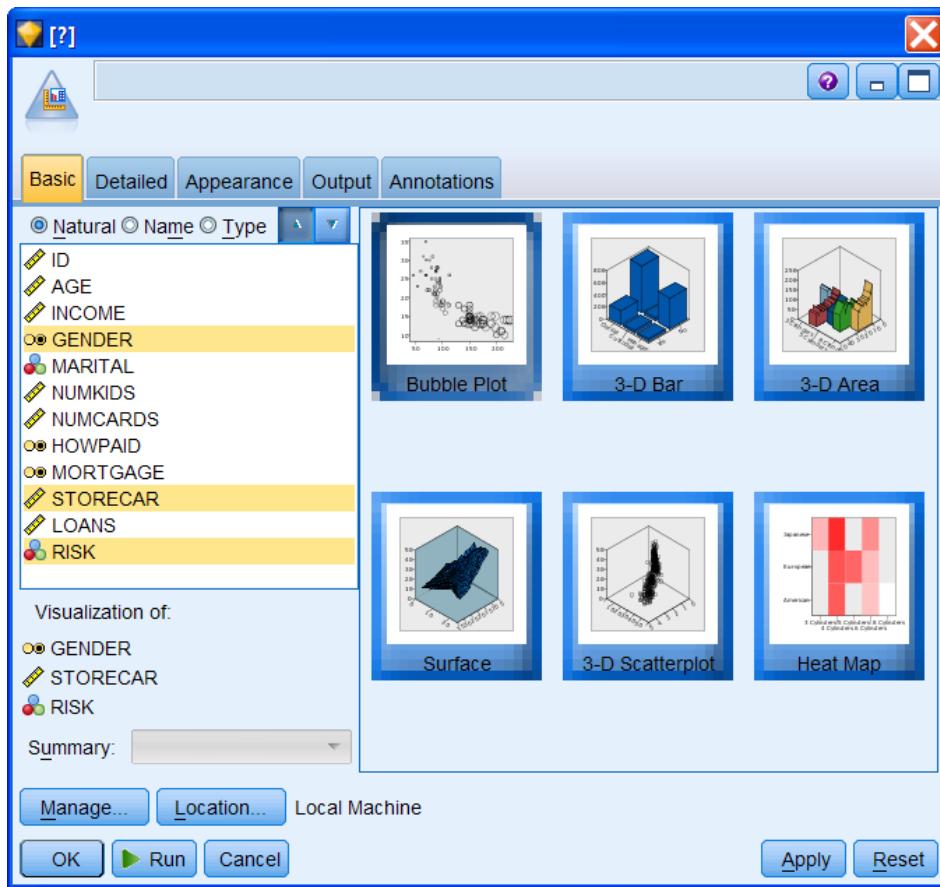
- Add a **Graphboard** node to the stream near the Type node
- Connect the **Type** node to the **Graphboard** node
- Edit the **Graphboard** node

Figure 7.29 Graphboard Node Dialog

As the message says in the dialog, the user is asked to select one or more fields from the list; as selections are made, Graphboard begins to make recommendations.

Using Ctrl-click, select **GENDER**, **STORECAR**, and **RISK**

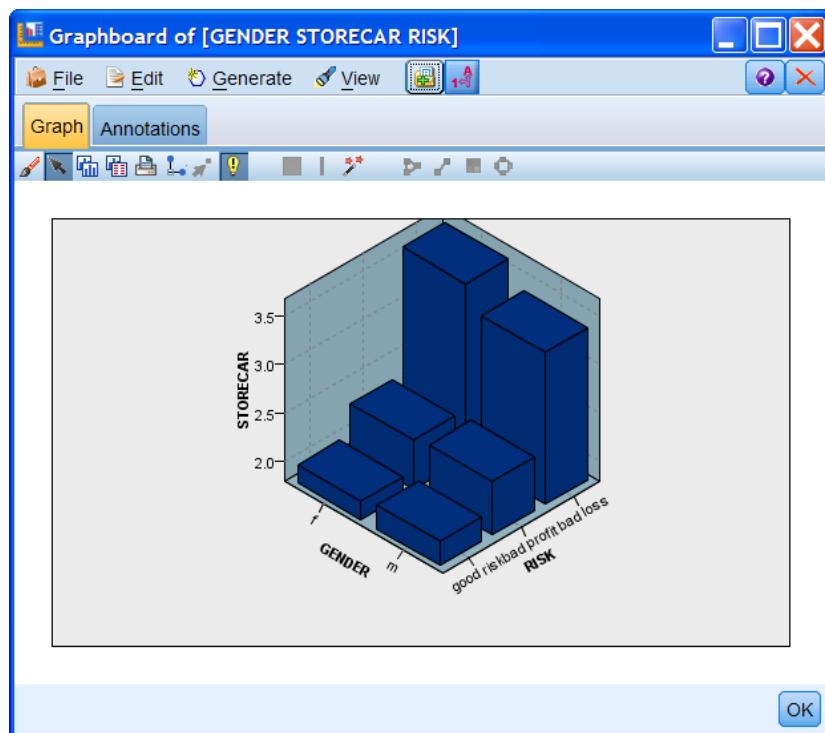
The graphs displayed shift with each selection. Finally there are several graphs recommended.

Figure 7.30 Graph Types Recommended

The first choice is a 3-D bar chart, which calculates a summary statistic for a continuous field and displays the results for the joint categories of two categorical fields. By default, the summary statistic is the *Sum* (see Summary dropdown in lower left). We'll change this to the *Mean*.

Select the **3-D Bar** chart by clicking it
 Click **Summary** dropdown and select **Mean**
 Click **Run**

The bar chart is displayed at an angle so all three dimensions are visible. It is clear that there is a relationship between these three fields. The number of department store credit cards increases from *good risk* to *bad profit* to *bad loss*. There doesn't appear to be much of an effect of gender, though, as the bars are about the same height for both males and females.

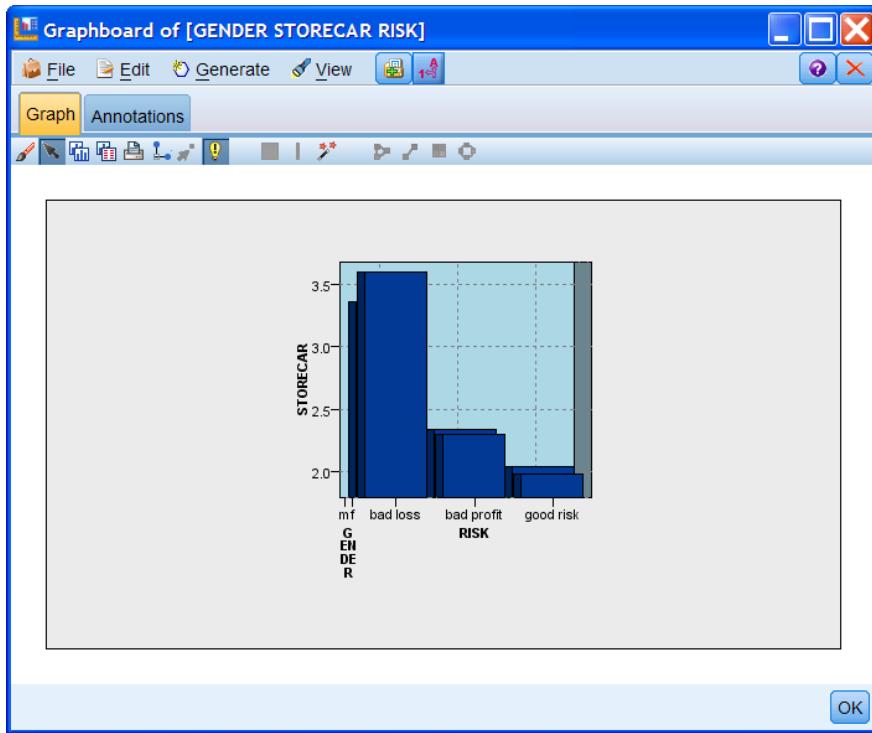
Figure 7.31 3-D Bar Chart of GENDER, RISK, and STORECAR

If you hover the mouse cursor over a bar, a pop-up displays the values for all three fields for that bar (not shown, but try it).

The graph can be rotated to make it easy to see the bars and their height. To do so, click anywhere in the graph. The cursor changes into this symbol and you can drag the mouse to rotate the graph in any axis.

Rotate the graph so that the **GENDER axis** is displayed into the screen/page as in Figure 7.32

This makes it clear that there isn't much difference in the relationship between *STORECAR* and *RISK* by categories of *GENDER*; it also shows very directly the relationship between those first two fields.

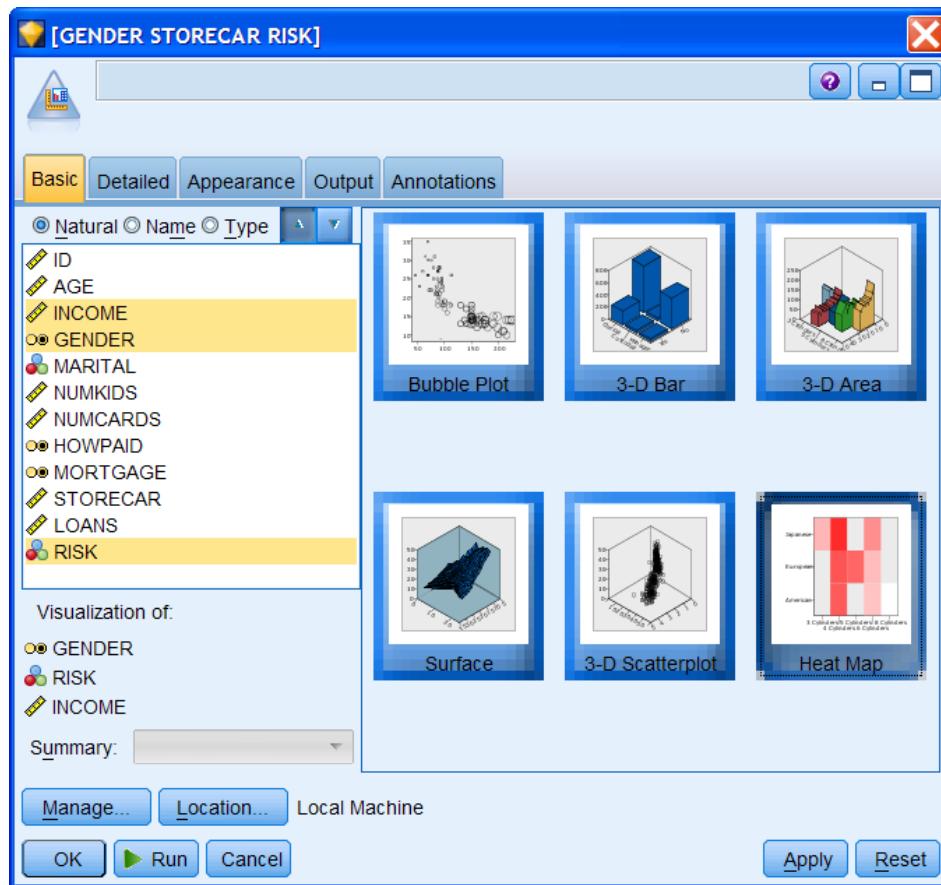
Figure 7.32 3-D Barchart Rotated with GENDER Axis into Page/Screen

Close the 3-D barchart

We can use the Graphboard node to easily switch to a different graph type and also change one field.

- Edit the **Graphboard** node
- Ctrl-click **STORECAR** to deselect it
- Ctrl-click **INCOME** to select it
- Click the **Heat Map** sample graph

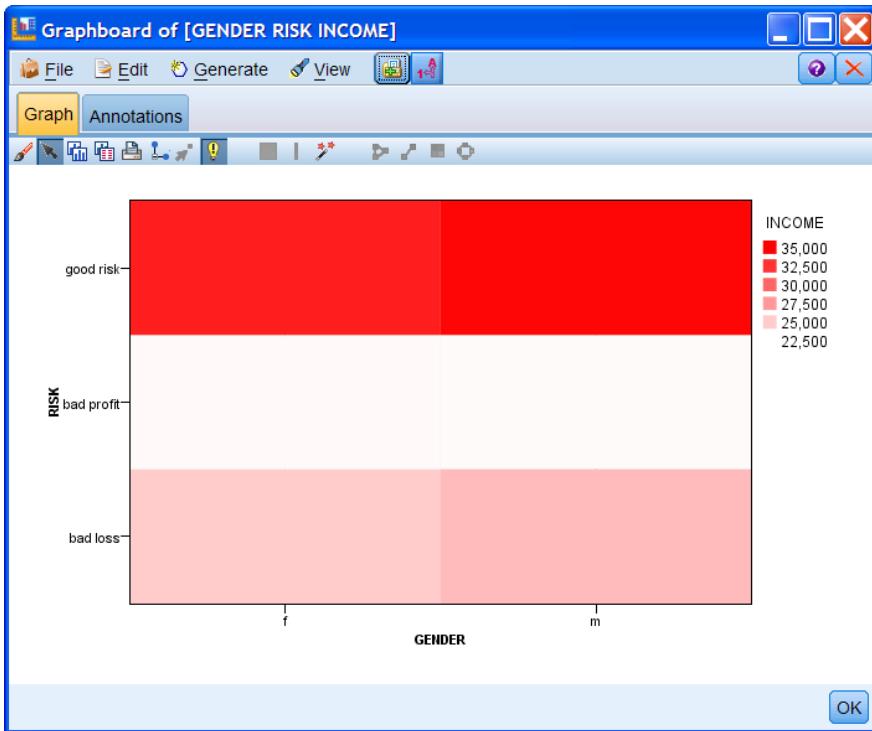
A heat map calculates a summary statistic for a continuous field for the joint distribution of two categorical fields, as with a 3-D barchart. A Heat Map, though, is like a table that uses colors instead of numbers to represent the values for the cells. Bright, deep red indicates the highest value, while gray indicates a low value. The value of each cell, in this instance, will be the mean of *INCOME* by *GENDER* and *RISK*.

Figure 7.33 Selecting a Heat Map Graph

Click **Run**

We find a very intriguing relationship in the heat map. Not surprisingly, the highest mean income (deepest red) is associated with the *good risk* category. But the midrange mean income is associated with the *bad loss* group, and the lowest mean incomes with *bad profit* customers.

Again, there doesn't seem much difference by gender, although in the *bad loss* and *good risk* groups, males have somewhat higher incomes. As with the 3-D bar chart, you can hover over a square in the graph and information on field values will pop up.

Figure 7.34 Heat Map Graph of GENDER, RISK, and INCOME

We conclude this lesson with a brief discussion of how to edit PASW Modeler graphs to introduce this capability.

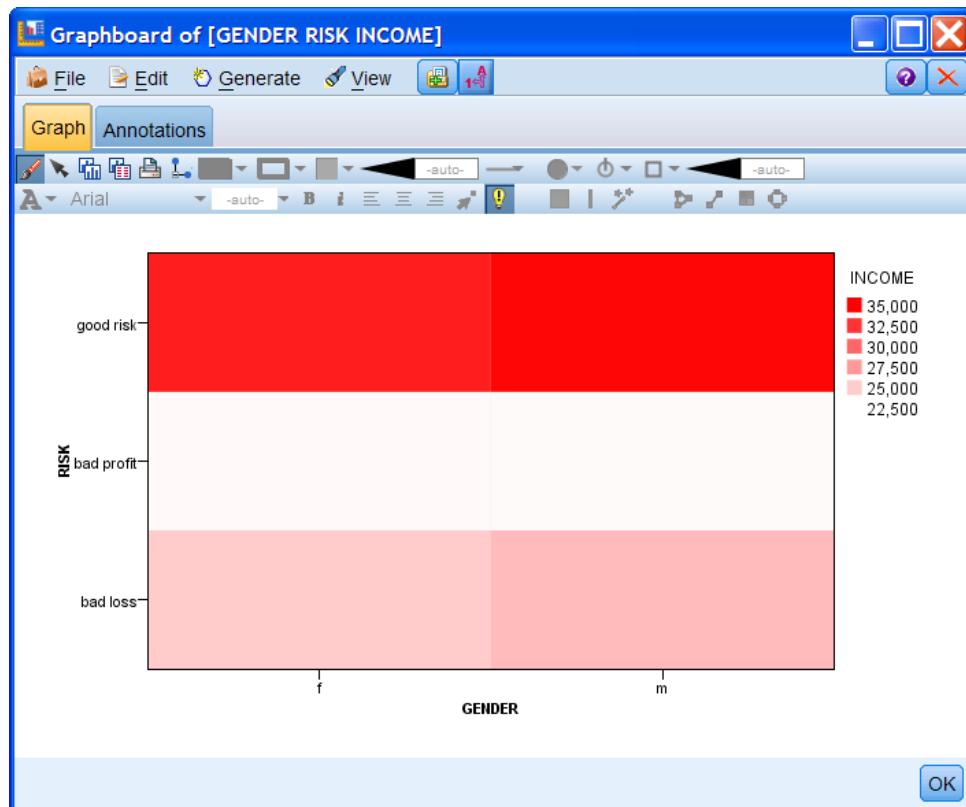
Editing PASW Modeler Graphs

We saw how PASW Modeler graphs can be modified to change the data displayed with a Web graph by using Interaction mode. If you would like to change the look of a graph, you use Edit mode. This mode can be selected from the menus (View....Edit Mode), or by clicking on the Edit mode button . Graphs can be edited to:

- Format text
- Change the fill color and pattern of frames and graphic elements
- Change the color and dashing of borders and lines
- Change the size of graphic elements (such as bars and points)
- Change the axis and scale settings
- Sort, exclude, and collapse categories on a categorical axis

We'll use Edit mode to make a few simple changes to the heat map.

Click View...Edit Mode

Figure 7.35 Editing Mode for the Heat Map

Notice how the tool bar changes to display tools for changing the borders, fonts, and colors. To change an item you click on it to select it. Let's change the basic color used to display intensity and the font size for the gender categories.

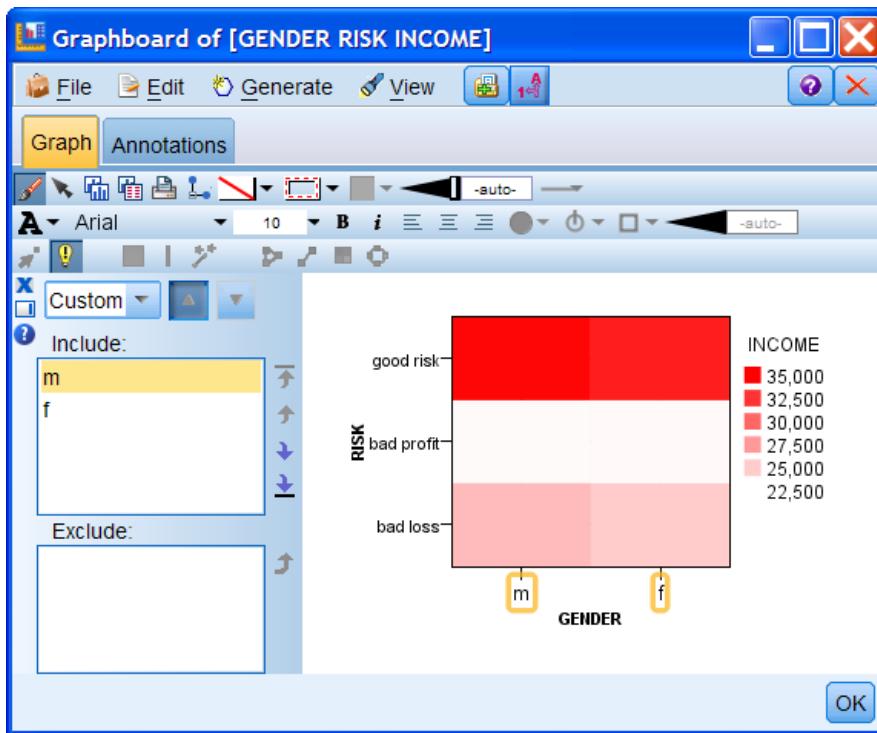
- Click on one of the bars so that all of these are selected (there will be a rectangle around them)
- Click on the **color** tool (the first tool in the toolbar) and select a **blue shade**
- Click on any of the **horizontal axis label values** (e.g., f) for **GENDER** to select these
- Click in the **Font size box** on the tool bar and change the size to **10**

Once a graph is being edited, additional tabs appear below the graph allowing you to change the scale of the axis and the display of ticks and gridlines, as appropriate for the graph type.

Next we'll change the order of the *GENDER* categories.

- Click **View** and select **Categories** checkbox
- Click the label **GENDER** on the **horizontal** axis
- In the **Include** box on the left, click **m**, and then click the **up arrow**

The order of the categories has now been switched along with their font size.

Figure 7.36 Edited Heat Map

We encourage you to investigate the various editing choices available for PASW Modeler graphs to tailor their appearance to your preferences.

Summary

In this lesson you have been introduced to a number of methods to explore relationships in data. You should now be able to:

- Produce a matrix table to investigate a relationship between two categorical fields
- Use a Web graph to visualize associations between categorical fields
- Use correlations to quantify the linear relationship between two continuous fields
- Use means to quantify the relationship between continuous and categorical fields
- Use the Graphboard node to create graphs that are appropriate for the fields selected
- Edit graphs to modify their appearance

Exercises

In this session we will use the stream created in the previous exercises and investigate whether there are any simple relationships in the data. In future lessons we will attempt to predict the field *Response to campaign*, and so we will focus on relationships between this field and others in the data.

1. Import data from *charity.sav* (Read names and labels; Read labels as data)
2. Connect a Matrix node from the Output palette to the source file node. We'll use this node to look at the relationship between *RESPONSE* and *AGEBAND*. Place *RESPONSE* in the row and *AGEBAND* in the column, request counts and column % be displayed in the matrix table. How would you characterize the relationship of these two fields?
3. Connect a Web node from the Graphs palette to the source file node.
4. Edit the Web node to produce a web plot showing the relationships between the following fields:
Response to campaign (RESPONSE)
Pre-campaign spend category (SPENDB)
Pre-campaign visit category (VISITB)
Gender (SEX)
Age category (AGEBAND)
5. Due to the substantial number of records in the data, set *Show only links above* to 200, set *Weak links below* to 300, and set *Strong links above* to 400. Run the node.
6. Edit the web plot by hiding irrelevant connections. What are the three strongest connections with the responder value of the response to campaign field? Which age groups are most associated with the non-responders?
7. Next we will investigate the relationship between the continuous fields of ORISPEND (*Pre-campaign expenditure*) and ORIVISIT (*Pre-campaign visits*). Use the Plot node to display a graph of the two fields (ORIVISIT on the X-axis; ORISPEND on the Y-axis) and the Statistics node to request summary statistics and the correlation coefficient. Does there appear to be a relationship in this data between these two fields?
8. Finally, we will investigate whether there is a relationship between the ORISPEND (*Pre-campaign expenditure*) and the RESPONSE (*Response to campaign*) field. Use the Means node to display all of the statistics for the Responder and Non-responder groups on pre-campaign expenditure.
9. Use the Graphboard node to create an appropriate graph to study the relationship between the same two fields as in Question 8. Which graph did you select? What is the relationship between the two fields? If so, is this consistent with your previous conclusions?
10. Add a third field to the graph, such as *SEX*, using Graphboard to help select the graph. Which graph type did you choose? What is the relationship between these three fields?

11. Edit one of the graphs you have created above to modify its appearance, including adding a title, changing font type and size for the axes, and the color of the graph elements.
12. Save a copy of the stream under the name *ExerLesson7.str*.

Lesson 8: Combining Data Files

Objectives

- Introduce the Append node as a method of joining sets of records together
- Introduce the Merge node as a method of combining multiple records from two or more files to create a single record
- Use SuperNodes to simplify streams

Data

In this lesson we will merge three data files from a financial organization. The data files contain information on customers and the accounts they hold. Table 8.1 lists information about these files.

Table 8.1 Data Files Used in This Lesson

File Name	File Type	Description	Fields
Customer.dat	Tab delimited text file	Information on each individual customer	Id number, age, gender, region, income, marital status, number of children, car ownership, mortgage, initial contact date.
Acct97.txt	Comma delimited text file	Information on accounts opened in 1997	Id number, account type, opening and current balance, date of opening account and account number (old style).
Accounts98.sav	Statistics data file	Information on accounts opened in 1998	Id number, account type, date of opening account, opening and current balance, account type, customer and account reference number.

8.1 Introduction

Often one of the early tasks in the Data Preparation phase of the CRISP-DM methodology is to merge data files from two or more sources. This is quite common in a variety of projects when using customer data, or in any situation where multiple databases are maintained on the persons or organizations of interest. For example, patient data may need to be merged from accounting, in-patient and out-patient databases to provide a complete record of information.

Sometimes, though, files are combined after data exploration and understanding is completed. This order of operations is followed because it can be more natural and efficient to look at outliers and distributions of fields within each separate file before creating a larger file.

In any case, before data mining modeling can begin, all of the different data sources must be pulled together to form a single data file in a PASW Modeler stream. In this lesson we introduce methods that can be used to perform such data manipulation within PASW Modeler.

We will first introduce the Append node as a method of joining data files that contain similar fields for separate groups of records. The Merge node will then be introduced as a method of joining data files that contain different information for the same records.

Once we have successfully joined the three data files, we will introduce the SuperNode as a method of condensing and simplifying the contents of the Stream Canvas.

8.2 **Using the Append Node to Combine Data Files**

Similar pieces of information for different groups of records may be stored in different data files. Examples of this include:

- Bank account information for different financial years
- Examination results for different academic years
- Fraud information for different local offices
- Transaction data for different weeks

There is often a need to collectively analyze such data, possibly to compare performance over subsequent years or to discover group differences. To analyze such information within PASW Modeler, the data files must be combined into one single file. The Append node joins two or more data sources together so that information held for different groups of records can be analyzed and compared.

The Append node is found in the Record Ops palette and can take multiple inputs; that is, a number of separate streams may enter the node. It joins two or more data files by reading and passing downstream all of the records from the first source, then reading and passing the records from the next source, and so on.

The order in which the sources are read is originally defined by the order in which they are attached to the Append node. The first attached source is named the *main dataset* and, by default, its format is used to define the structure of the data leaving the Append node. As we will see, you can specify the order in which inputs are processed by reordering the list of input nodes in the Append dialog box, and you have options when the same fields are not present in the input files.

The Append node assumes that the data entering via the other inputs have a similar, if not identical, structure to that of the primary input source, and it joins data by field name in the data file.

When there are different numbers of fields contained within the sources (inputs) the Append node uses the following rules:

- If an input has fewer fields than the main dataset, then the records from the input source are padded with the undefined value (\$null\$) for the missing fields.
- If an input has more fields than the main dataset, then the extra fields are, by default, filtered from the stream, but an option allows all fields to be input from all datasets. If fields are input from all datasets, then records from sources not containing these fields are padded with the undefined value (\$null\$).

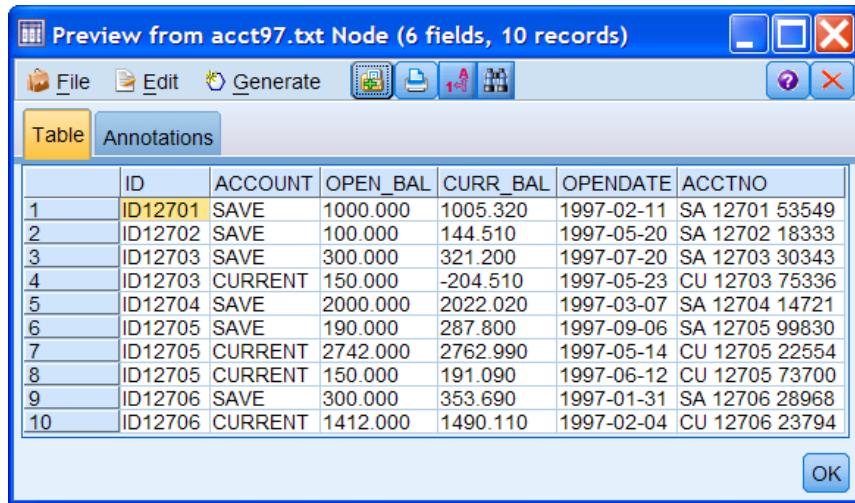
In this example we will illustrate the action of the Append node by using it to join the two files containing information on accounts opened in 1997 and 1998. The files, *acct97.txt* (comma-delimited) and *accounts98.sav* (Statistics data file), are detailed in the Data section above.

We will first read both data files into PASW Modeler. To read in the comma-delimited file, *acct97.txt*, use the Var. File node from the Sources palette:

Place a **Var. File** source node on the Stream Canvas

Edit the source node and set the file to **acct97.txt** stored in **c:\Train\ModelerIntro**
 Click the **Preview** button

Figure 8.1 Table for acct97.txt File Showing Accounts Opened in 1997



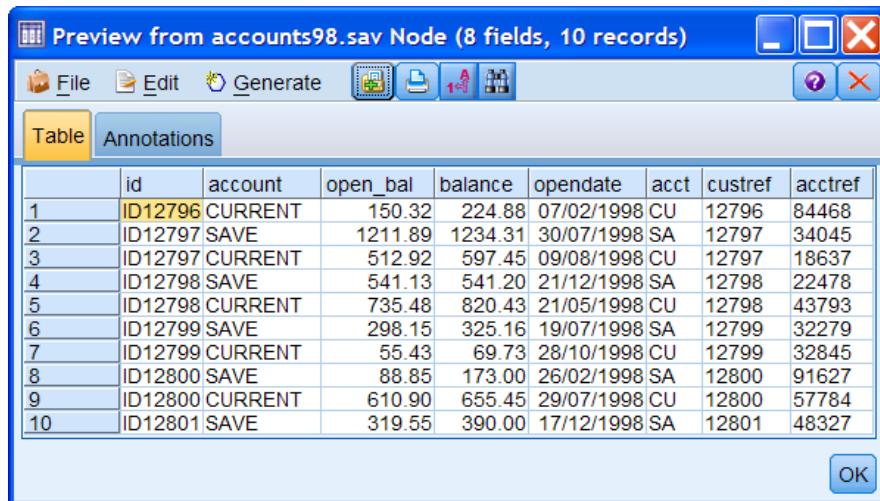
	ID	ACCOUNT	OPEN_BAL	CURR_BAL	OPENDATE	ACCTNO
1	ID12701	SAVE	1000.000	1005.320	1997-02-11	SA 12701 53549
2	ID12702	SAVE	100.000	144.510	1997-05-20	SA 12702 18333
3	ID12703	SAVE	300.000	321.200	1997-07-20	SA 12703 30343
4	ID12703	CURRENT	150.000	-204.510	1997-05-23	CU 12703 75336
5	ID12704	SAVE	2000.000	2022.020	1997-03-07	SA 12704 14721
6	ID12705	SAVE	190.000	287.800	1997-09-06	SA 12705 99830
7	ID12705	CURRENT	2742.000	2762.990	1997-05-14	CU 12705 22554
8	ID12705	CURRENT	150.000	191.090	1997-06-12	CU 12705 73700
9	ID12706	SAVE	300.000	353.690	1997-01-31	SA 12706 28968
10	ID12706	CURRENT	1412.000	1490.110	1997-02-04	CU 12706 23794

Notice, field names are uppercase.

To read in the Statistics file, *accounts98.sav*, use the Statistics File node in the Sources palette.

Close the Preview window
 Click **OK** to close the Var. File node
 Place a **Statistics File** node on the Stream Canvas
 Edit the node and set the file to **accounts98.sav** in **c:\Train\ModelerIntro**
 Make sure **Read names and labels** and **Read data and labels** are selected
 Click the **Preview** button

Figure 8.2 Table for accounts98.sav Showing Accounts Opened in 1998



	id	account	open_bal	balance	opendate	acct	custref	acctref
1	ID12796	CURRENT	150.32	224.88	07/02/1998	CU	12796	84468
2	ID12797	SAVE	1211.89	1234.31	30/07/1998	SA	12797	34045
3	ID12797	CURRENT	512.92	597.45	09/08/1998	CU	12797	18637
4	ID12798	SAVE	541.13	541.20	21/12/1998	SA	12798	22478
5	ID12798	CURRENT	735.48	820.43	21/05/1998	CU	12798	43793
6	ID12799	SAVE	298.15	325.16	19/07/1998	SA	12799	32279
7	ID12799	CURRENT	55.43	69.73	28/10/1998	CU	12799	32845
8	ID12800	SAVE	88.85	173.00	26/02/1998	SA	12800	91627
9	ID12800	CURRENT	610.90	655.45	29/07/1998	CU	12800	57784
10	ID12801	SAVE	319.55	390.00	17/12/1998	SA	12801	48327

The tables show that the account reference number was stored in a slightly different format across the two years. In 1997 the account number was held in one field called **ACCTNO**. Alternatively, in 1998, this number was split into three components, the account type (**ACCT**), the customer reference

(*CUSTREF*) and the account reference number (*ACCTREF*). Furthermore, the field *CURR_BAL* contained in the first file is not present in the second. Notice, field names are lowercase.

We will see how to deal with these inconsistencies in the following examples.

We now join these two data sources together so that the two years may be compared.

Close the Preview window

Click **OK** to close the Statistics File node

Place an **Append** node from the Record Ops palette to the right of the two source nodes

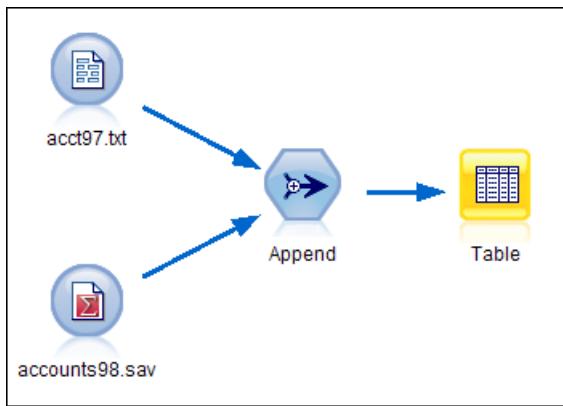
Connect the **source** node labeled *acct97.txt* to the **Append** node

Connect the **source** node labeled *accounts98.sav* to the **Append** node

Place a **Table** node to the right of the **Append** node

Connect the **Append** node to the **Table** node

Figure 8.3 Stream Joining Files Containing Accounts Opened in 1997 and 1998



Because we first connected *acct97.txt* to the Append node, *acct97.txt* serves as the main dataset. To see what this means, run the stream.

Run the **Table** node attached to the Append node

Scroll down to **row 163** in the Table output

Figure 8.4 Data Table Produced with acct97.txt as the Main Dataset

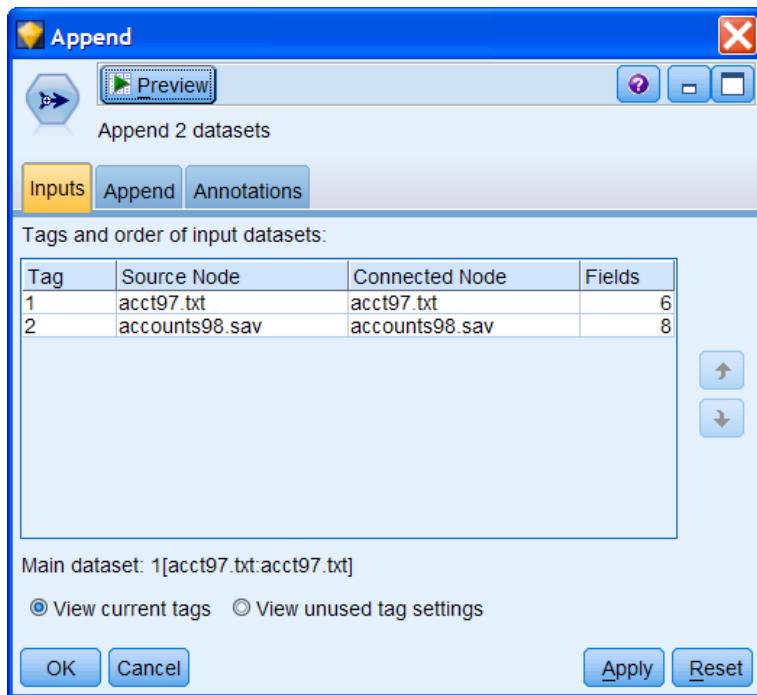
Table (6 fields, 358 records)

ID	ACCOUNT	OPEN_BAL	CURR_BAL	OPENDATE	ACCTNO
156	ID12792	SAVE	1024.970	1043.920	1997-09-26 SA 12792 96005
157	ID12792	CURRENT	1096.870	1112.110	1997-10-03 CU 12792 44630
158	ID12793	SAVE	1239.000	1337.200	1997-05-13 SA 12793 17564
159	ID12793	CURRENT	156.486	184.100	1997-11-18 CU 12793 36234
160	ID12794	SAVE	156.001	230.290	1997-08-26 SA 12794 70288
161	ID12795	SAVE	119.696	158.900	1997-03-07 SA 12795 80051
162	ID12795	CURRENT	85.980	185.460	1997-10-09 CU 12795 85444
163	ID12796	CURRENT	150.320	\$null\$	1998-02-07 \$null\$
164	ID12797	SAVE	1211.890	\$null\$	1998-07-30 \$null\$
165	ID12797	CURRENT	512.920	\$null\$	1998-08-09 \$null\$
166	ID12798	SAVE	541.130	\$null\$	1998-12-21 \$null\$
167	ID12798	CURRENT	735.480	\$null\$	1998-05-21 \$null\$
168	ID12799	SAVE	298.150	\$null\$	1998-07-19 \$null\$
169	ID12799	CURRENT	55.430	\$null\$	1998-10-28 \$null\$
170	ID12800	SAVE	88.850	\$null\$	1998-02-26 \$null\$
171	ID12800	CURRENT	610.900	\$null\$	1998-07-29 \$null\$
172	ID12801	SAVE	319.550	\$null\$	1998-12-17 \$null\$
173	ID12802	CURRENT	1034.090	\$null\$	1998-05-27 \$null\$
174	ID12802	CURRENT	1311.470	\$null\$	1998-07-22 \$null\$
175	ID12803	SAVE	1471.710	\$null\$	1998-07-13 \$null\$

It is clear that the main dataset (*acct97.txt*) defines the fields (and field names) that will be contained in the combined file. Fields unique to the second file (*accounts98.sav*) are left out of the combined file. Note that undefined values are set to \$null\$. For instance, *CURR_BAL* was not included in *accounts98.sav*, so the records from *accounts98.sav* have the \$null\$ value on *CURR_BAL*.

You may force a different source to be defined as the main dataset by reordering the sources listed in the Inputs tab of the Append node:

- Close the Table window
- Edit the **Append** node
- Click the **Inputs** tab

Figure 8.5 Append Node Dialog Box: Inputs Tab

The order of the inputs, given in *Tags and order of input datasets*, controls the order in which the records are read into the Append node. To promote or demote an individual input, select it and use the Up or Down buttons, respectively.

Select **acct97.txt** in the column **Source Node**

Click to demote this file

Click **OK**

Run the **Table** node attached to the **Append** node

Scroll down to **row 197** in the Table output

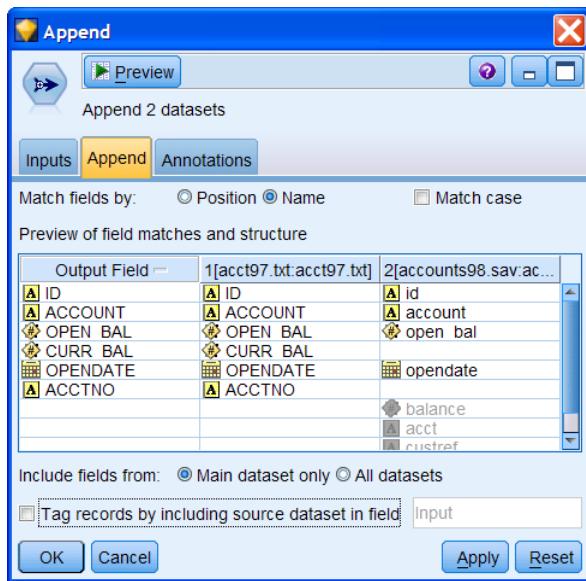
Figure 8.6 Data Table Produced with accounts98.sav as the Main Dataset

The screenshot shows a software interface for viewing a dataset. The title bar says "Table (8 fields, 358 records)". The menu bar includes "File", "Edit", "Generate", and various icons. Below the menu is a tab bar with "Table" selected. The main area is a grid of data with 8 columns and approximately 358 rows. The columns are labeled: id, account, open_bal, balance, opendate, acct, custref, and acctref. Some data values are clearly visible (e.g., 250.00, 321.98, 19/07/1998), while others are represented by "\$null\$". An "OK" button is at the bottom right of the grid.

Now, fields unique to *accounts98.sav* are undefined (\$null\$) for records coming from *acct97.txt*. Note that the records from *acct97.txt* are appended to *accounts98.sav*, so records from *accounts98.sav* will precede records from *acct97.txt* in the combined data file. Field names now are lowercase.

To overcome the problems raised by specifying one of the files as the main dataset, let's have both files play an equal role.

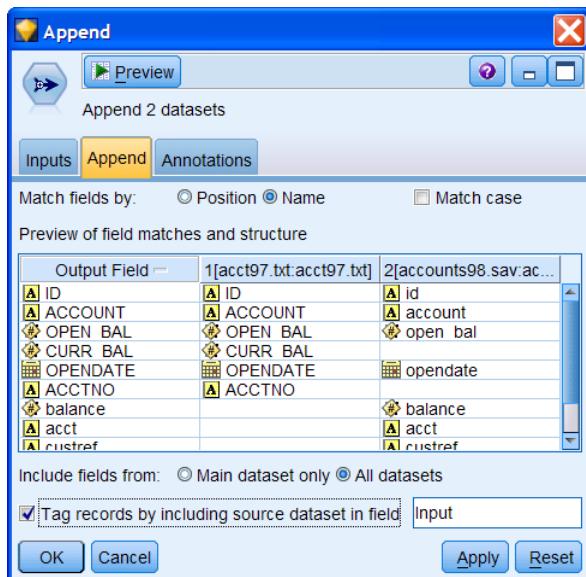
- Close the Table window
- Edit the **Append** node
- Click the **Inputs** tab
- Select **acct97.txt**
- Click to bring **acct97.txt** to the top of the **Source Node** list (it's a matter of taste, but we want records from **acct97.txt** be the first in the combined dataset)
- Click the **Append** tab

Figure 8.7 Append Node: Append Tab

Fields to be included in the new data file downstream are listed in the Output Field column. The *Include Fields from* option gives you the ability to change the way the fields within the data files will be retained. By selecting the *All datasets* option, fields found in any of the data sources will be retained in the stream. This is the option we need in order to retain the *ACCTNO* field from the *acct97.txt* source and the *ACCT*, *CUSTREF*, and *ACCTREF* fields from the *accounts98.sav* data source. Note that we also can create a field recording the data source of a record.

Click **All datasets** option button

Check the **Tag records by including source dataset in field** check box

Figure 8.8 Append: Append Tab: All Datasets Option Selected

Notice, the option *Match case* is not checked. This is the reason why fields could be matched even though field names were uppercase in one dataset and lowercase in the other.

Click **OK**

Run the **Table node** attached to the Append node

Scroll to **row 163** in the Table window

Figure 8.9 Appending Datasets: Including Fields from All Datasets

The screenshot shows a Windows-style application window titled "Table (11 fields, 358 records) #1". The window has a menu bar with "File", "Edit", "Generate", and "Help" options. Below the menu is a toolbar with icons for file operations. The main area is a grid table with 11 columns labeled: ID, ACCOUNT, OPEN_BAL, CURR_BAL, OPENDATE, ACCTNO, balance, acct, cust..., acctref, and Input. The rows contain data entries such as ID12792, ACCOUNT SAVE, OPEN_BAL 1024.970, CURR_BAL 1043.920, etc. Row 163 is highlighted with a yellow background. The bottom right corner of the window contains an "OK" button.

The new field *input* records the source file for each case. Having retained all fields from both files, it is no surprise that we have many \$null\$ values. However, there is good news. Notice, that account number is contained in *ACCTNO* for records coming from *acct97.txt* and in *ACCT*, *CUSTREF*, and *ACCTREF* for records coming from *accounts98.sav*. So, we have the information readily available to fill in the \$null\$ values.

There are two ways to do this:

- Using Derive nodes and the substring function from the CLEM language, the account reference number in the 1997 data file can be split into the three-component format used in 1998. The Derive nodes would be placed between the source node labeled *acct97.txt* and the Append node.
- Using a Derive node and the >< (concatenate) function from the CLEM language, the account type, customer reference and account reference numbers in the 1998 data file can be concatenated into the account reference number format used in 1997. This Derive node would be placed between the Statistics File node labeled *accounts98.sav* and the Append node.

Since the most recent format (used in 1998) is to have the account number in the three components we will use the first solution:

Close the Table window

Move the Append node to the right to make some space

Select a **Derive** node from the Fields Ops. palette and connect it between the **Var. File** source node and the **Append** node

Edit the **Derive** node

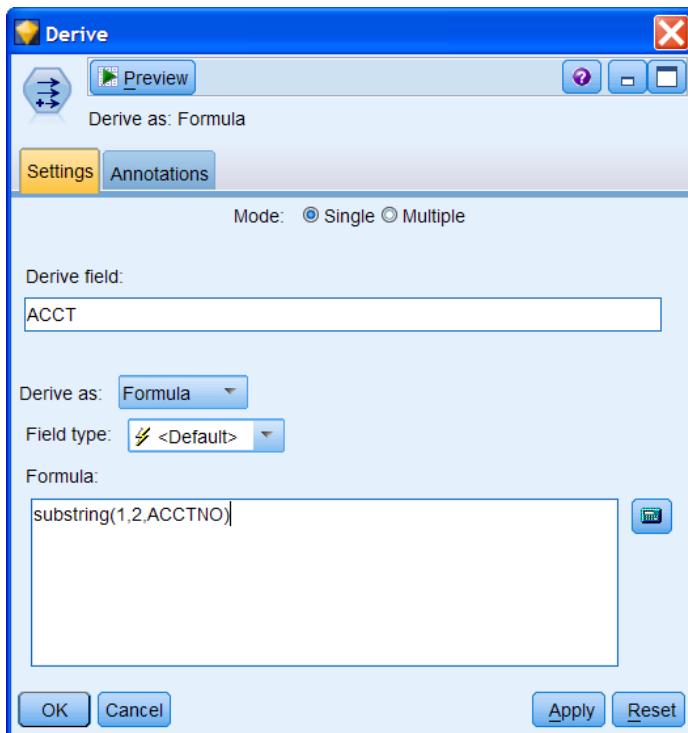
The function for extracting sections from a field is **substring** and takes the format:

```
substring(start,length,string)
```

where *start* and *length* denote the starting position and number of characters you wish to extract from the field *string* (you would use the name of the field of interest)

To extract the first two characters to form the account type reference code, complete the Derive node dialog, as shown in Figure 8.10. Remember that the CLEM language is case sensitive. You can always use the Expression Builder dialog as an aid.

Figure 8.10 Completed Derive Node to Extract the Account Type Reference Code



Click **OK**

Add a **Derive** node to create the **CUSTREF** field, using the substring function on **ACCTNO**, now beginning in **4** with length **5**. The formula is **substring(4, 5, ACCTNO)**

Add a **Derive** node to create the **ACCTREF** field, again taking a substring on **ACCTNO**, this time beginning in **10** with length **5**. The formula is **substring(10,5,ACCTNO)**

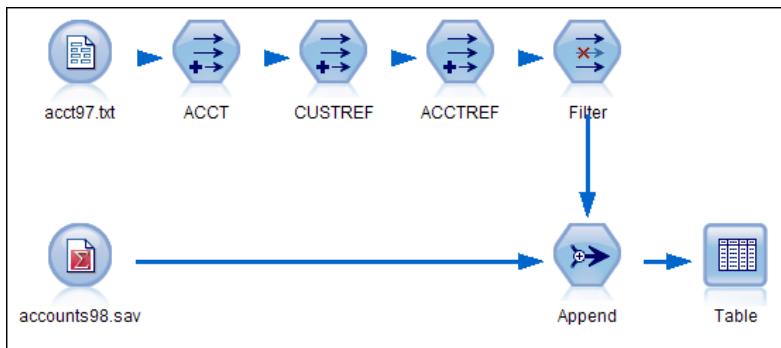
Finally, before the stream can be run, the now redundant **ACCTNO** field should be filtered out of the stream before the data passes through to the Append node.

Select a **Filter** node from the Field Ops palette

Connect it between the last **Derive** node and the **Append** node

Edit the **Filter** node and click once on the arrow pointing from the **ACCTNO** field

Click **OK**

Figure 8.11 Stream to Extract Account Number Information Before Appending

Run the **Table** node attached to the **Append** node
Scroll down to **row 163**

Figure 8.12 Append Streams Plus the Combined File

Table (10 fields, 358 records)											
File Edit Generate 											
Table Annotations											
ID	ACCOUNT	OPEN_BAL	CURR_BAL	OPENDATE	ACCT	CUSTREF	ACCTREF	balance	Input		
156	ID12792	SAVE	1024.970	1043.920	1997-09-26	SA	12792	96005	\$null\$ 1		
157	ID12792	CURRENT	1096.870	1112.110	1997-10-03	CU	12792	44630	\$null\$ 1		
158	ID12793	SAVE	1239.000	1337.200	1997-05-13	SA	12793	17564	\$null\$ 1		
159	ID12793	CURRENT	156.486	184.100	1997-11-18	CU	12793	36234	\$null\$ 1		
160	ID12794	SAVE	156.001	230.290	1997-08-26	SA	12794	70288	\$null\$ 1		
161	ID12795	SAVE	119.696	158.900	1997-03-07	SA	12795	80051	\$null\$ 1		
162	ID12795	CURRENT	85.980	185.460	1997-10-09	CU	12795	85444	\$null\$ 1		
163	ID12796	CURRENT	150.320	\$null\$	1998-02-07	CU	12796	84468	224.88 2		
164	ID12797	SAVE	1211.890	\$null\$	1998-07-30	SA	12797	34045	1234.31 2		
165	ID12797	CURRENT	512.920	\$null\$	1998-08-09	CU	12797	18637	597.45 2		
166	ID12798	SAVE	541.130	\$null\$	1998-12-21	SA	12798	22478	541.20 2		
167	ID12798	CURRENT	735.480	\$null\$	1998-05-21	CU	12798	43793	820.43 2		
168	ID12799	SAVE	298.150	\$null\$	1998-07-19	SA	12799	32279	325.16 2		
169	ID12799	CURRENT	55.430	\$null\$	1998-10-28	CU	12799	32845	69.73 2		
170	ID12800	SAVE	88.850	\$null\$	1998-02-26	SA	12800	91627	173.00 2		
171	ID12800	CURRENT	610.900	\$null\$	1998-07-29	CU	12800	57784	655.45 2		
172	ID12801	SAVE	319.550	\$null\$	1998-12-17	SA	12801	48327	390.00 2		
173	ID12802	CURRENT	1034.090	\$null\$	1998-05-27	CU	12802	25045	1056.82 2		
174	ID12802	CURRENT	1311.470	\$null\$	1998-07-22	CU	12802	93554	1409.95 2		
175	ID12803	SAVE	1471.710	\$null\$	1998-07-13	SA	12803	10897	1564.60 2		
176	ID12804	SAVE	2209.380	\$null\$	1998-04-16	SA	12804	47633	2245.98 2		

We have successfully joined the files from 1997 and 1998. Note that year can be extracted from the *OPENDATE* field. If this were not available, then adding a year field to each stream, before appending, would be useful.

Close the Table window

Note

We could align the *CURR_BAL* and *BALANCE* fields by renaming one of them to the other field name within a Filter node upstream of the Append node. If time permits, try this as an exercise.

Our next task is to use the Merge node to attach the customer information database, *customer.dat*, to this combined account data stream.

8.3 **Using a Merge Node to Combine Data Files**

In many organizations, different pieces of information for individuals are held in separate locations. Examples of this include:

- Customer information held separately from purchase information
- Account details held in a database separate from transactions
- Information in a housing organization may be held at an individual and property level
- Panel surveys where information is collected on the same individuals at regular intervals.

There is often a need to collectively analyze such data; for example, one could attempt to find patterns in purchase behavior and link them back to demographics. To be able to analyze such information within PASW Modeler, the data files must be combined into one single file. The Merge node joins two or more data sources together so that information held for an individual in different locations can be analyzed collectively.

Like the Append node, the Merge node is found in the Record Ops palette and can take multiple inputs, as a number of separate streams may enter the node. It works in a slightly different way than other nodes in PASW Modeler in that it reads multiple input records and creates a single output record, containing some or all of the input fields.

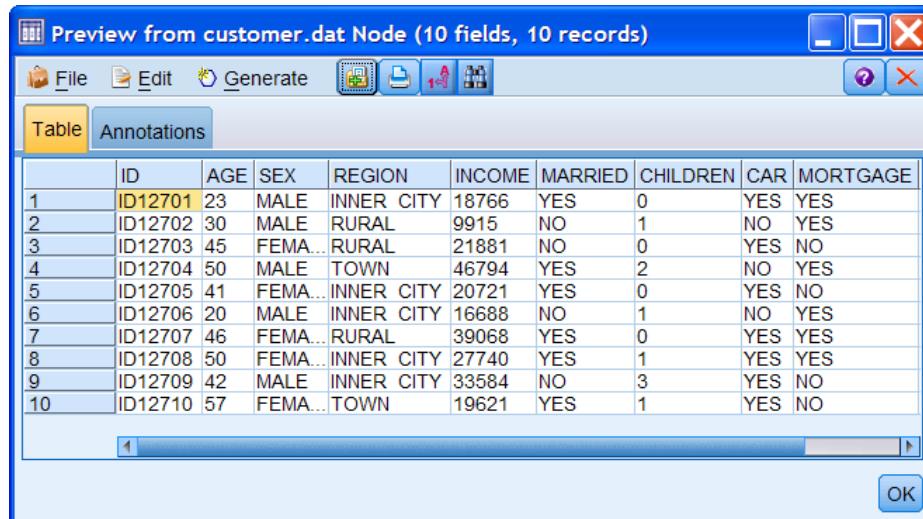
The Merge node contains four major tabs. The Merge tab allows you to specify the method of merging, since the Merge node can cope with a variety of different merging situations. The Filter tab allows you to drop unwanted fields. The Inputs node lists the data sources involved in the merge; the ordering of these sources determines the field ordering in the merged data. The Optimization tab provides two options that allow you to merge data more efficiently when one input dataset is significantly larger than the other datasets or when the data is already presorted by all or some of the key fields that you are using to merge.

In this section we will illustrate the Merge node by attaching customer information held in the file *customer.dat* to the combined account information created in the previous section using the Append node.

First we must read the customer account information into PASW Modeler and view the data:

- Select a **Var. File** node and place it at the bottom of the Stream Canvas near the **Append** node
- Edit the **Var. file** node
- Set the file to **customer.dat** held in the **c:\Train\ModelerIntro** directory
- As delimiter, check **Tab**
- Deselect the **Comma Delimiter** check box (not shown)
- Click the **Preview** button

Figure 8.13 Customer Information Held in customer.dat



The screenshot shows a software interface titled "Preview from customer.dat Node (10 fields, 10 records)". The window has a toolbar with icons for File, Edit, Generate, and various data manipulation tools. Below the toolbar, there are two tabs: "Table" (which is selected) and "Annotations". The main area displays a table with 10 columns and 10 rows of data. The columns are labeled: ID, AGE, SEX, REGION, INCOME, MARRIED, CHILDREN, CAR, and MORTGAGE. The data rows are as follows:

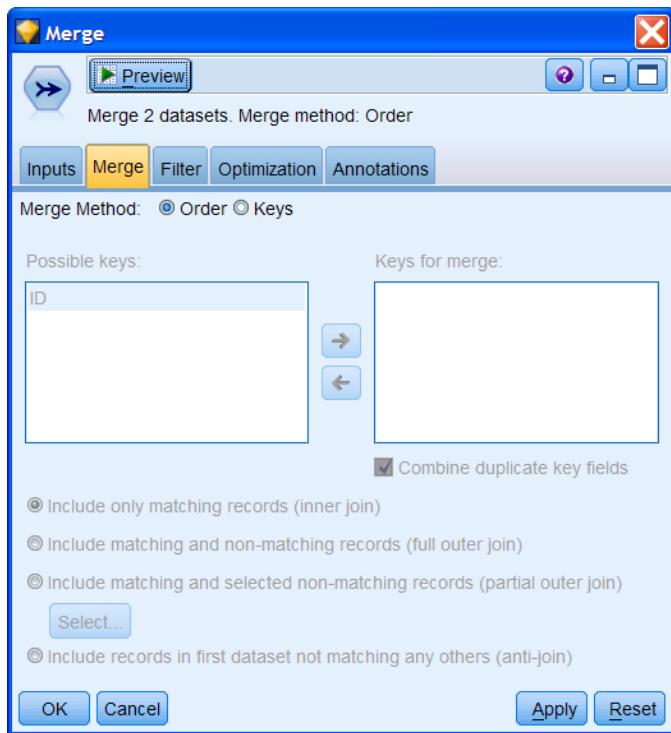
ID	AGE	SEX	REGION	INCOME	MARRIED	CHILDREN	CAR	MORTGAGE
1	ID12701	23	MALE	INNER CITY	18766	YES	0	YES
2	ID12702	30	MALE	RURAL	9915	NO	1	NO
3	ID12703	45	FEMA...	RURAL	21881	NO	0	YES
4	ID12704	50	MALE	TOWN	46794	YES	2	NO
5	ID12705	41	FEMA...	INNER CITY	20721	YES	0	YES
6	ID12706	20	MALE	INNER CITY	16688	NO	1	NO
7	ID12707	46	FEMA...	RURAL	39068	YES	0	YES
8	ID12708	50	FEMA...	INNER CITY	27740	YES	1	YES
9	ID12709	42	MALE	INNER CITY	33584	NO	3	YES
10	ID12710	57	FEMA...	TOWN	19621	YES	1	NO

At the bottom right of the preview window is an "OK" button.

The file contains various demographics on customers and the date at which they first opened an account at the bank.

We will now use the Merge node to attach the information held in this file to the previously created data stream.

- Close the Preview window
- Click **OK** to return to the Stream Canvas
- Select a **Merge** node and place it below the **Append** node
- Connect the **Var. file** node labeled **customer.dat** to the **Merge** node
- Connect the **Append** node to the **Merge** node
- Edit the **Merge** node

Figure 8.14 Merge Dialog: Method of Merging (Merge Tab)

There are two methods of merging:

Order. This constructs the n^{th} output record from the n^{th} record from each input file in turn. Once any of the inputs runs out of records, no further output records are produced.

Keys. This is often referred to as an “equi-join” or “keyed table match,” where records that have the same value in the field(s) defined as the “key” are merged. If multiple records contain the key field, all possible merges are returned. For example, two data sources containing:

KEY	Date	Purchase Type	Amount
1	23 April 1999	Check	39.99
1	4 June 1999	Visa	120.00
2	30 July 1999	Cash	25.99
2	7 August 1999	Cash	30.00

KEY	Age	Income
1	24	20,000
2	47	42,500

would produce this merged file:

KEY	Date	Purchase Type	Amount	Age	Income
1	23 April 1999	Check	39.99	24	20,000
1	4 June 1999	Visa	120.00	24	20,000
2	30 July 1999	Cash	25.99	47	42,500
2	7 August 1999	Cash	30.00	47	42,500

In this example the customer data file contains one record per customer and the account data files contain one record per account that a customer holds. Each customer may have more than one

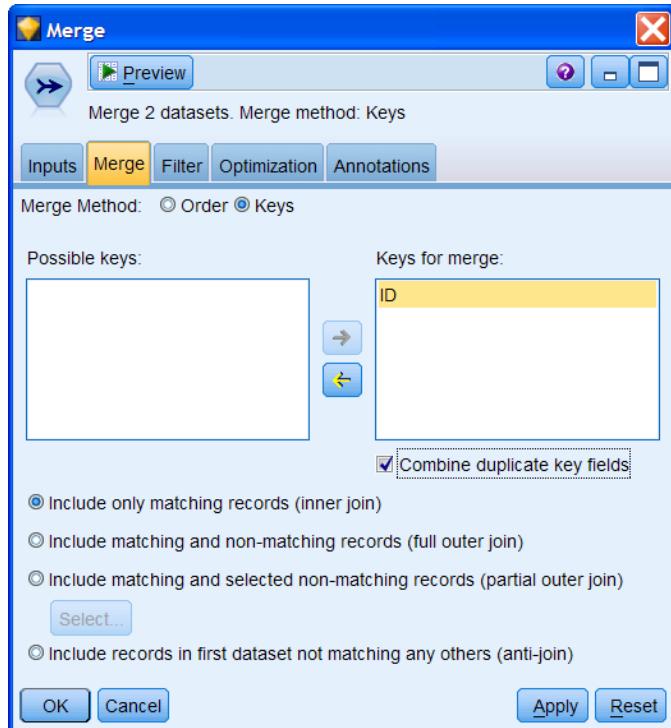
account and both files contain a unique ID number that identifies each individual customer. We will therefore perform a key merge and use *ID* as the key field.

Set Merge Method to Keys

Select **ID** in the Possible keys: list

Click to move **ID** into the Keys for merge list

Figure 8.15 Merging by a Key Field



Fields contained in all input sources appear in the *Possible keys* list. To identify one or more fields as the key field(s), click once on the field and then click the right arrow to move the selected field into the *Keys for merge* list.

There are four major methods of merging using a key field:

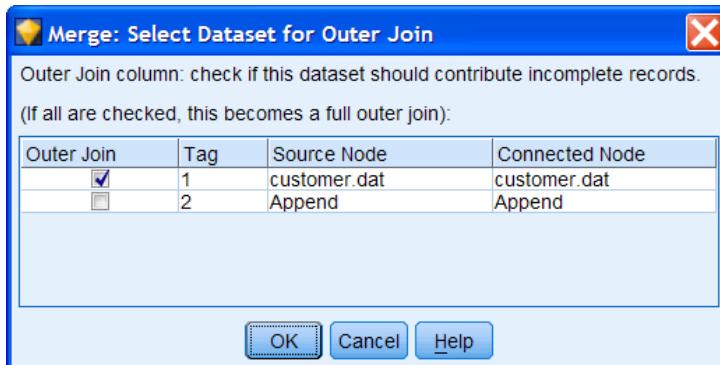
- **Include only matching records (inner join).** Select to merge only complete records, that is, records that are available in both source files.
- **Include matching and non-matching records (full outer join).** This means that if a key field(s) value is present in one data source but not present in others, the incomplete records are still retained. The undefined value (\$null\$) is added to the missing fields and included in the output.
- **Include matching and selected non-matching records (partial outer join).** This method performs what are called left and right outer joins. All records from the specified file are retained, along with only those records from the other file(s) that match records in the specified file on the key field(s). The *Select* button allows you to designate which file is to contribute incomplete records. So if we wished to retain all records from the *customer.dat* file and only those from the Append node that matched on *ID*, then *customer.dat* would be checked in the Merge: Select Dataset for Outer Join dialog.

- **Include records in first dataset not matching any others (anti-join).** This option provides an easy way of identifying records in a dataset that do not have records with the same key values in any of the other datasets involved in the merge. Thus only records from the dataset that match with no other records will be retained. They might be examined for incorrect key values.

To illustrate the setup for a partial (left or right) outer join:

Click the **Include matching and selected non-matching records (partial outer join)** option button
Click **Select** button

Figure 8.16 Merge: Select Dataset for Outer Join Dialog



Here the *customer.dat* data source is checked in the *Outer Join* column, so all records from this source, including records that don't match the other file(s) on the key field(s), will be retained. In this way, partial outer joins (left outer join and right outer join) can be performed within the Merge node.

Click **Cancel**

Combine duplicate key fields deals with the problem of duplicate field names (one from each source) when key fields are used. This option ensures that there is only one output field with a given name and is enabled by default, except in the case when streams have been imported from earlier versions of PASW Modeler (Clementine). When this option is disabled, duplicate key fields must be renamed or excluded using the Filter tab in the Merge node dialog.

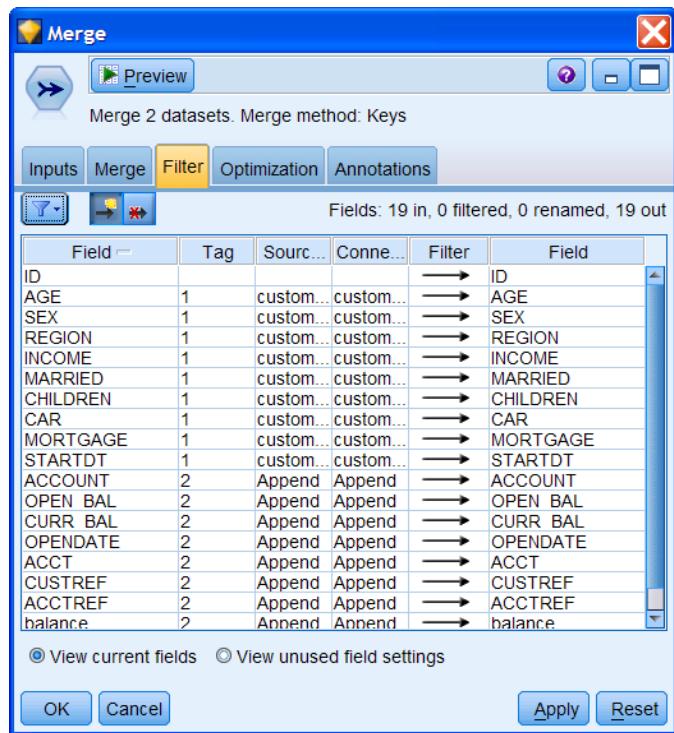
For this example we will use *ID* as the key field and choose a full outer join.

Select the **Include matching and non-matching records (full outer join)** option (not shown)

The Inputs and Annotations tabs serve the same purposes as before: the Inputs tab controls the files to be merged; the Annotations tab allows you to add comments describing the operation.

Click the **Filter** tab

The Filter tab is similar to the Filter node and performs the same operations as that node. The only difference in this node is that the Source Node and Connected Node names provide source information for the input fields.

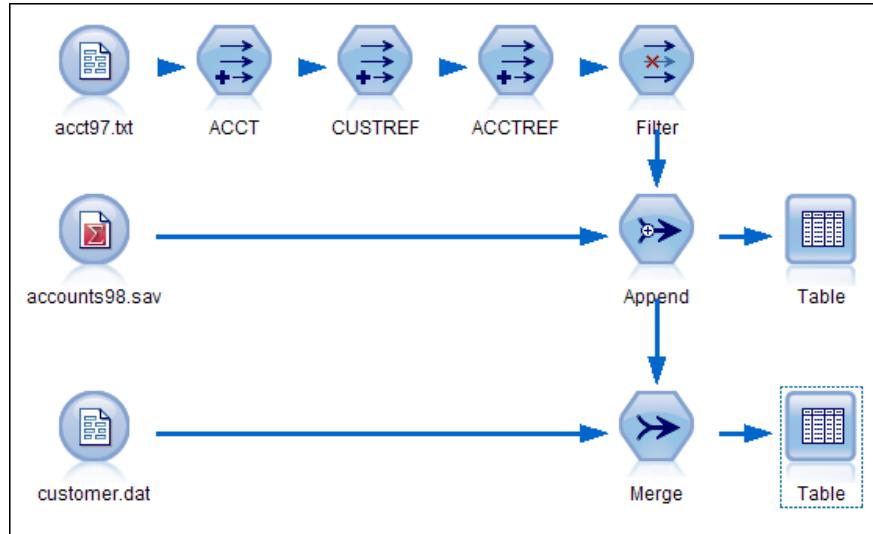
Figure 8.17 Merge Node: Filter Tab

The Filter tab can be used to remove duplicate (or unwanted) fields by clicking on the arrow, which will become disabled (red and crossed out). Alternatively, you may wish to rename a duplicate field name by clicking on the name in the right-hand *Field* column and entering a new name. These fields can then be compared further downstream as an integrity check.

In this example we will use the current settings.

Click **OK**

Connect a **Table** node to the right of the **Merge** node

Figure 8.18 Complete Stream Joining the Two Account Files and the Customer Database

Run the **Table** node attached to the **Merge** node

Figure 8.19 Combined Data File

The screenshot shows the 'Table' node interface in IBM SPSS Modeler. The title bar reads 'Table (19 fields, 358 records)'. The menu bar includes 'File', 'Edit', 'Generate', and various icons. The main area is a grid showing data from row 1 to 20. The columns are labeled: ID, AGE, SEX, REGION, INCOME, MARRIED, CHILDREN, CAR, MORTGAGE, STARTDT, ACCOUNT, OPEN_BAL, and CURR_BAL. The data includes various demographic and financial details for different individuals.

ID	AGE	SEX	REGION	INCOME	MARRIED	CHILDREN	CAR	MORTGAGE	STARTDT	ACCOUNT	OPEN_BAL	CURR_BAL
1	ID12701	23	MALE	INNER CITY	18766	YES	0	YES YES	1995-01-05	SAVE	1000 000	1005 320
2	ID12702	30	MALE	RURAL	9915	NO	1	NO YES	1994-12-13	SAVE	100 000	144 510
3	ID12703	45	FEMA...	RURAL	21881	NO	0	YES NO	1994-02-18	SAVE	300 000	321 200
4	ID12703	45	FEMA...	RURAL	21881	NO	0	YES NO	1994-02-18	CURRENT	150 000	-204 510
5	ID12704	50	MALE	TOWN	46794	YES	2	NO YES	1994-09-08	SAVE	2000 000	2022 020
6	ID12705	41	FEMA...	INNER CITY	20721	YES	0	YES NO	1995-03-05	CURRENT	2742 000	2762 990
7	ID12705	41	FEMA...	INNER CITY	20721	YES	0	YES NO	1995-03-05	SAVE	190 000	287 800
8	ID12705	41	FEMA...	INNER CITY	20721	YES	0	YES NO	1995-03-05	CURRENT	150 000	191 090
9	ID12706	20	MALE	INNER CITY	16688	NO	1	NO YES	1994-02-21	SAVE	300 000	353 690
10	ID12706	20	MALE	INNER CITY	16688	NO	1	NO YES	1994-02-21	CURRENT	1412 000	1490 110
11	ID12706	20	MALE	INNER CITY	16688	NO	1	NO YES	1994-02-21	CURRENT	132 000	230 640
12	ID12707	46	FEMA...	RURAL	39068	YES	0	YES YES	1995-04-11	SAVE	10 000	55 030
13	ID12708	50	FEMA...	INNER CITY	27740	YES	1	YES YES	1994-06-13	SAVE	5 000	10 550
14	ID12709	42	MALE	INNER CITY	33584	NO	3	YES NO	1996-08-02	SAVE	50 000	57 210
15	ID12710	57	FEMA...	TOWN	19621	YES	1	YES NO	1995-11-03	CURRENT	43 000	121 610
16	ID12710	57	FEMA...	TOWN	19621	YES	1	YES NO	1995-11-03	CURRENT	980 000	1052 070
17	ID12711	63	FEMA...	INNER CITY	47630	YES	0	NO YES	1995-09-05	SAVE	1000 000	1082 270
18	ID12712	26	FEMA...	INNER CITY	22378	NO	0	YES YES	1996-03-02	CURRENT	1100 000	1123 280
19	ID12713	62	FEMA...	RURAL	20837	YES	0	YES NO	1996-12-19	SAVE	50 000	122 110
20	ID12713	62	FEMA...	RURAL	20837	YES	0	YES NO	1996-12-19	CURRENT	1 000	35 920

OK

It appears that the merge has been successful and, for individuals with more than one account (for example ID12703), there are multiple records containing duplicate customer information.

We are now in a position to begin exploring and studying the account information. However, before we can do this we will introduce the SuperNode as a way to simplify the Stream Canvas, thus creating more space to perform our analysis.

8.4 SuperNode

A SuperNode allows a section of a stream, consisting of a number of nodes, to be condensed into a single node, represented by a star icon. This process is called encapsulation.

There are three types of SuperNode, depending on where the encapsulation begins and ends:

- Source SuperNode
- Process SuperNode
- Terminal SuperNode

The Source SuperNode

Source SuperNodes contain a data source and can be used wherever a data source node can be used. The left-hand side of the SuperNode icon is shaded, indicating that connections can only be made *from* this node and not to the node.

Figure 8.20 Source SuperNode



SuperNode

The Terminal SuperNode

Terminal SuperNodes contain one or more terminal nodes and can be used in the same way as a terminal node. The right-hand side of the SuperNode icon is shaded, indicating that connections can only be made *to* this node and not from the node.

Figure 8.21 Terminal SuperNode



The Process SuperNode

Process SuperNodes represent a section of a stream that contains only manipulation nodes. The Process SuperNode is not shaded, indicating that connections can be made to and from the node.

Figure 8.22 Process SuperNode



SuperNode Rules

There are certain limitations on applying encapsulation:

- There must be a path between the two selected nodes
- An entire stream cannot be included in a SuperNode
- Sections of streams to be encapsulated cannot contain forked paths (i.e. nodes with multiple output connections).

This last rule does not apply to terminal SuperNodes that contain terminal nodes in every one of the forked paths. The selected node and all nodes downstream from it will be condensed into a single terminal SuperNode.

Creating SuperNodes

There are several ways to create a SuperNode. One method is the following:

- Right click a node to reveal the context menu
- Select the Create SuperNode option
- Choose the Select... option. This causes the node to become highlighted and the cursor to become a star
- Move the cursor to another node, either upstream or downstream from the first selected node
- Click using any mouse button on this node

A SuperNode will replace the two selected nodes and any nodes between them. A source SuperNode will be created if the selected node furthest upstream is a source node. A terminal SuperNode will be created if the selected node furthest downstream is a terminal node. If neither of the above situations applies then a process SuperNode will be created.

If nodes are selected in advance, then a SuperNode can be created by right-clicking on one of the selected nodes and then clicking Create SuperNode. A SuperNode can also be created from a node and continuing downstream by right-clicking the node and then clicking Create SuperNode...From Here.

SuperNodes can also be nested within SuperNodes, greatly reducing stream size.

We will illustrate simplifying the stream by creating a SuperNode to include the Derive nodes used to correct the ACCT fields. Then we will create a SuperNode that includes all the data processing nodes.

Close the Table window

Right click the Derive node named ACCT

From the context menu, select Create SuperNode...Select

Place the cursor over the Derive node named ACCTREF and click once

Right-click the new SuperNode, and select Rename and Annotate from the context menu

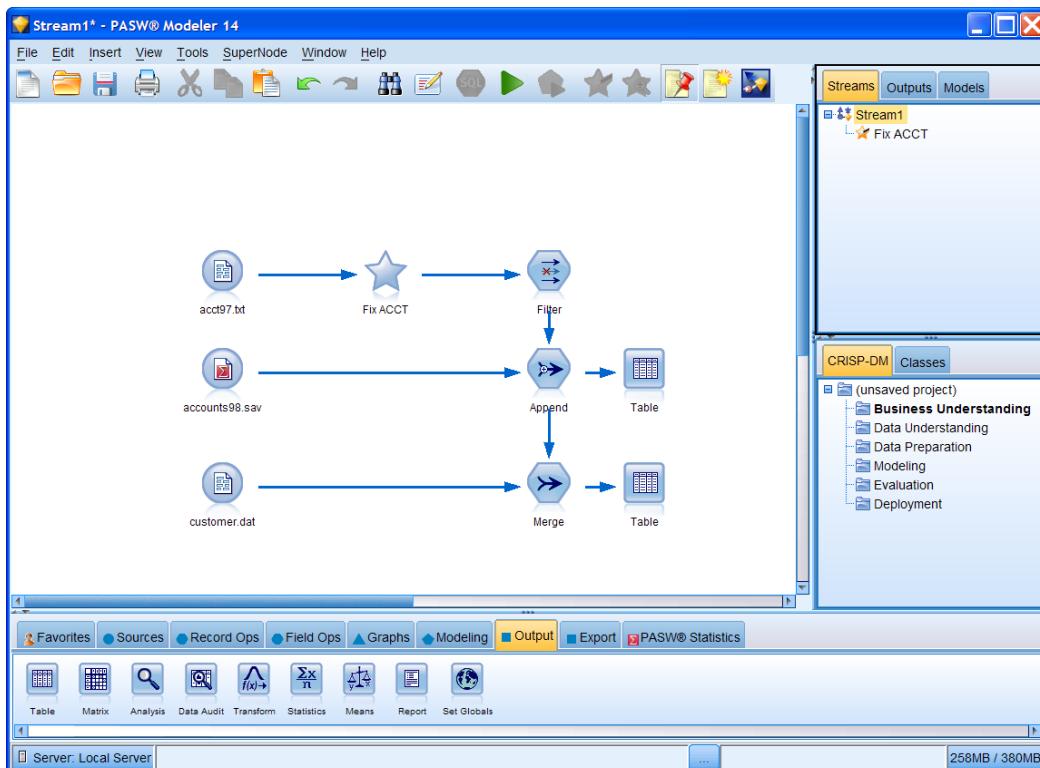
Click the Custom option button, type Fix ACCT in the Name text box, and then click OK

A SuperNode should replace the Derive nodes.

Click the Streams manager tab

Expand Stream1 in the Streams manager

Figure 8.23 Simplified Stream Containing a Process SuperNode

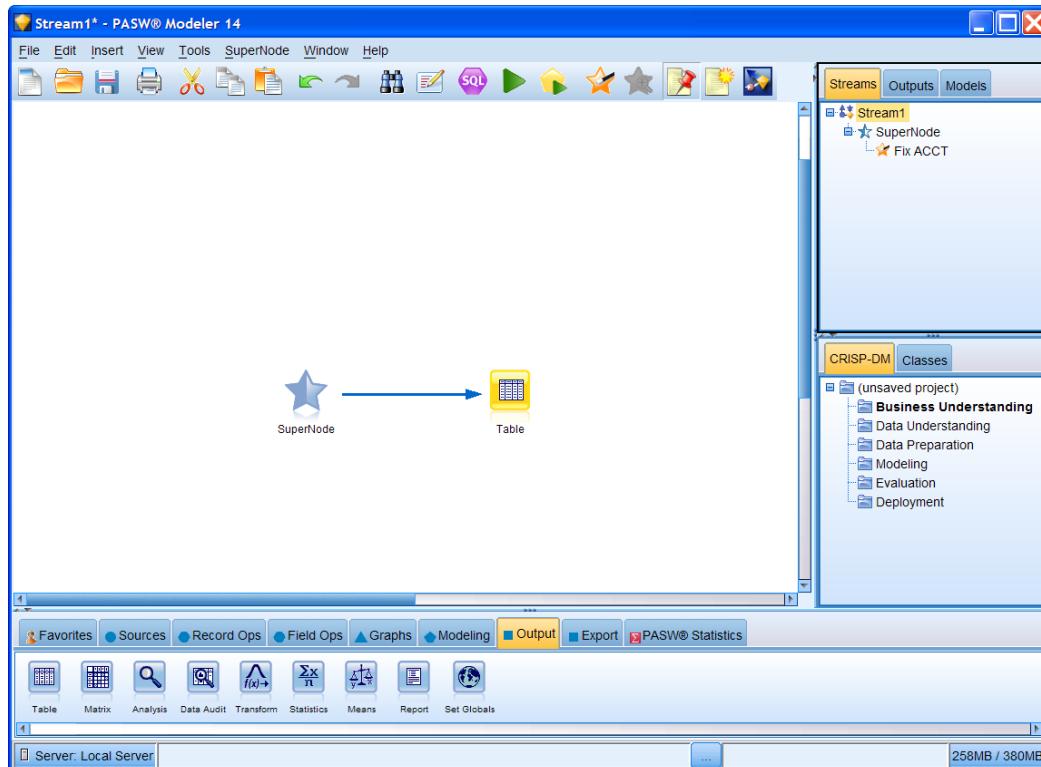


The Fix ACCT SuperNode encapsulates the three Derive nodes. Notice that the presence of a SuperNode is also reflected in the Streams Manager.

Next we will create a SuperNode that includes all the data processing steps (source nodes, data manipulation nodes, Append and Merge nodes)

- Delete all **Table** nodes except for the one attached to the Merge node
- Select **all nodes except the Table** node (click Edit...Select All, then Ctrl-click the Table node)
- Right-click any of the selected nodes, and then select **Create SuperNode**
- Expand **Stream1** completely in the Streams manager

Figure 8.24 SuperNode Nested within a SuperNode



The data source and processing nodes are now encapsulated within a single SuperNode that includes a SuperNode containing Derive nodes. This hierarchy is reflected in the Streams manager. By judicious use of SuperNodes, complex streams can be simplified.

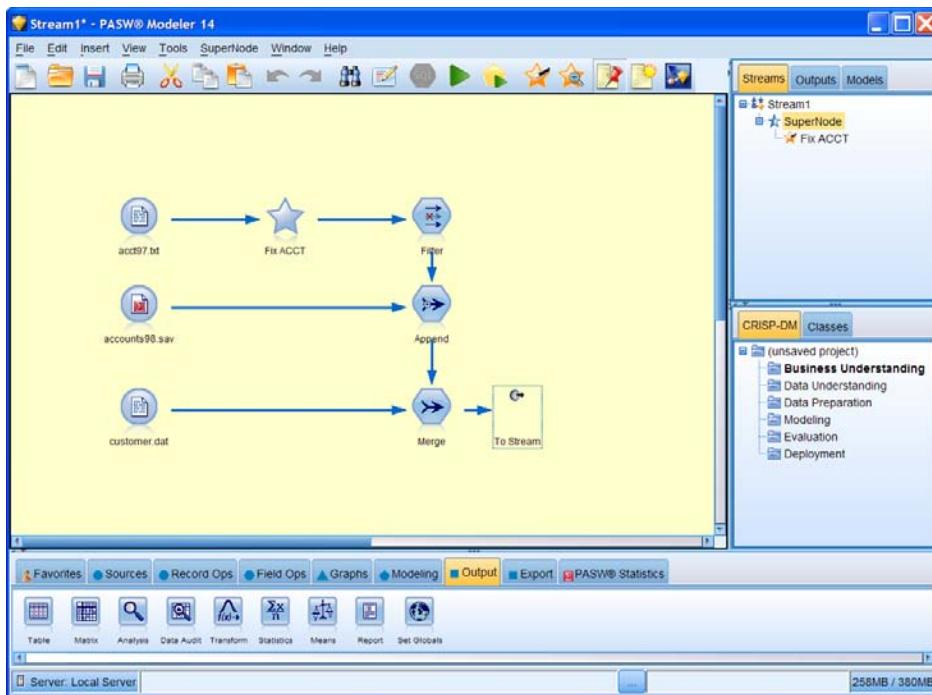
8.5 Editing SuperNodes

The nodes within a SuperNode may be edited in a variety of ways:

Zoom In

SuperNodes can be edited by clicking with the right mouse button on the SuperNode and selecting the Zoom In from the Context menu. The section of stream condensed within the SuperNode will be displayed in the Stream Canvas (the background color will change as a cue).

- Right-click the **SuperNode**
- From the context menu, select **Zoom in**
- (Alternatively, click SuperNode under Stream1 in the Streams manager)

Figure 8.25 A SuperNode Zoomed In

Changes can now be made to the nodes contained in the SuperNode. The connection between the SuperNode and the main stream is marked (see Merge node).

To zoom out, use the SuperNode menu, or click the button in the Toolbar, or click on Stream1 in the Streams Manager.

Expanding

Alternatively the contents of a SuperNode can be put back into the Stream Canvas by expanding the SuperNode. To expand a SuperNode, right-click the SuperNode and click Expand from the Context menu.

8.6 Saving and Inserting SuperNodes

SuperNodes can be saved and then inserted into other streams. This allows you to retain and easily reuse PASW Modeler nodes that perform frequently needed functions (purchase date and account field manipulation, reclassification of product codes, etc.)

To Save a SuperNode

- Right-click on the SuperNode in the Stream Canvas, and then click Save SuperNode

Alternatively:

- Zoom in on the SuperNode
- From the SuperNode menu, choose Save SuperNode
- Specify a filename and directory in the dialog box (SuperNodes have the extension .slb)
- Select whether to add the saved SuperNode to the current project
- Click Save

To Insert a SuperNode

- From the Insert menu in the PASW Modeler window, select SuperNode from File
- Select a SuperNode (.slb) file from the current directory or browse to a different one
- Click Insert

Advantages of Using SuperNodes

In addition to simplifying streams, SuperNodes have a number of benefits:

- Applications passed on to other users show less detail and are more easily followed
- PASW Modeler nodes can be combined to create SuperNodes which are more specific and directly related to your business application
- SuperNodes can be saved and re-used in other streams

Summary

In this lesson we have introduced a number of ways to combine data files from different sources. You should now be able to:

- Use the Append node to join groups of records with similar fields
- Use the Merge node to combine information linked to an individual, but held in different sources, into one record
- Use SuperNodes to simplify your stream

Exercises

We will use data from the travel company that we used previously for the exercises in Lesson 5. In that lesson the data file had already been combined; here our job is to combine separate files to create the merged holidays file.

custtravel1.dat and **custtravel2.dat**

These two files contain information on a subset of trips taken by the company's customers. The files contain the following fields:

CUSTID	Customer reference number
NAME	Customer name
DOB	Date of birth
GENDER	Gender
REGION	Home location
NUMPARTY	Number in party
HOLCOST	Total cost of holiday
NIGHTS	Number of nights away
TRAVDATE	Departure date
HOLCODE	Holiday reference code

holtravel.dat

This file contains information on different holiday travel programs the company offers. The file contains the following fields:

HOLCODE	Holiday code
COUNTRY	Country
POOL	Usage of a pool
ACCOM	Type of accommodation
DIST_TO_BEECH	Distance to beach in kms

1. Begin a new stream (File...New Stream).
2. Select the Var. File node from the Sources palette and place it on the stream canvas. Edit the node and set the file to *custtravel1.dat* held in the *c:\Train\ModelerIntro* directory. Make sure PASW Modeler reads field names from the first line of the file.
3. Repeat step 2 for the other two files, *custtravel2.dat* and *holtravel.dat*. Both files contain field names.
4. Select three Table nodes from the Output palette and place them next to each of the Var. File nodes. Connect each of the source nodes to a Table node and run each stream. Check each table to ensure that the data files are being read into PASW Modeler correctly.

First join the two customer data files using the Append node.

5. Connect the two customer source nodes to an Append node. Edit the Append node and ensure that *custtravel1.dat* is the first file in the list.

6. Connect a Table node to the Append node and run this section of the stream. Have the two files been successfully joined?

Now merge the holiday information onto the combined customer files.

7. Connect the Append node and the source node labeled *holtravel.dat* to a Merge node. Edit the Merge node.
8. In the Merge tab, select the keys method of merging. Choose the appropriate field as the merge key. Select the *Include matching and non-matching records (full outer join)* option.
9. Connect a Table node to the Merge node and run this section of the stream. Have the files been successfully merged?
10. Save the stream as *ExerLesson8.str* in the *c:\Train\ModelerIntro* directory.
11. *For those with extra time:* Create a supernode containing the three Var. File nodes. Which type of supernode have you created? Zoom in on the Supernode and expand it again.

Lesson 9: Aggregating Data

Objectives

- Introduce the Aggregate node to create summary records
- Introduce the SetToFlag node to transform a categorical field into a collection of flag fields
- Use the Merge node to combine the output from the Aggregate and SetToFlag nodes
- Introduce the Restructure node

Data

In this lesson we use the data file *fulldata.txt*. The file contains one record per account held by customers of a financial organization. The file contains demographic details on the customer and individual account information.

9.1 Introduction

As the previous lessons have illustrated, the general file structure of the data source may not be in the correct format for your analysis.

For example, purchase data is often held in a file containing one record per purchase. To analyze this at a customer level, the data must be restructured so that there is one record per customer, containing summary information. Using the Aggregate node, a new file can be created so that each record contains information such as total amount spent, average amount spent on certain goods, and total number of purchases.

Alternatively, it is often useful to have purchase data arranged so that each field represents a particular product and each record stores whether or not a customer has purchased that product. The SetToFlag node may be used to transform a single field containing a set of products into a collection of product flag fields within a record (or, generically, any categorical field into a set of multiple fields).

In the following sections we introduce the Aggregate and SetToFlag nodes to perform such data file manipulation. We also introduce the Restructure node to transform each category of a categorical field into a separate field, but this time each new field will take on the value from another continuous field. And finally, we will show how aggregation can be used in conjunction with each node.

9.2 Summarizing Data Using the Aggregate Node

It is often necessary to replace data with summary or aggregated information. For example:

- A data file, containing information on individual purchases can be replaced by a file containing summary information on individual customers, such as total amount spent.
- A data file in which a record contains end-of-month account balances for different accounts can be replaced by a file with one record per account containing average end-of-month balances for the year.
- Information held on individuals, such as age, can be summarized into information representing each household, such as average age.

The Aggregate node, located in the Record Ops palette, replaces a collection of input records with a summary, aggregated output record. During aggregation, the overall file structure normally changes because the case or record definition is altered.

During aggregation at least one key field must be specified. The key field defines the group of records that are to be replaced by one record, and multiple key fields may be used. For example, if summarizing purchase data into customer data, the key field will be a unique customer reference number/ID or a combination of fields that are used to uniquely identify each customer.

For each unique combination of the key field(s), values within an aggregate field (fields from which aggregate summaries are calculated) will be replaced by a single summary value. Summaries available include the mean, minimum, maximum, sum, and standard deviation. The aggregate fields must be continuous.

For example, Table 9.1 gives a set of possible input records taken from a personnel database:

Table 9.1 Input Records to the Aggregate Node

Employee Number	Department	Current Age	Initial Salary	Current Salary
10123	Sales	24	17,000	18,000
10124	Sales	29	16,500	20,000
10125	Sales	32	13,000	30,000
10126	Accounts	30	14,000	20,000
10127	Accounts	41	12,000	35,000
10128	Accounts	35	16,000	18,000
10129	R&D	29	10,000	14,000
10130	R&D	37	13,000	17,000
10131	R&D	45	12,500	27,500
10132	Marketing	35	25,000	26,000
10133	Marketing	21	12,000	14,000

Aggregating this information, using department as the key field, with aggregate fields age (with a mean summary), initial salary (with a minimum summary), and current salary (with a sum summary), will produce the data file in Table 9.2.

Table 9.2 Aggregated Data Using Department as the Key Field

Department	Mean Age	Minimum Initial Salary	Total Current Salary
Sales	28.33	13,000	68,000
Accounts	35.33	12,000	73,000
R&D	37.00	10,000	58,500
Marketing	28.00	12,000	40,000

To illustrate the use of the Aggregate node, we will reduce the data file *fulldata.txt* from a set of records containing account information into a data stream containing one record per customer.

It is best to aggregate data after missing values have been removed or modified. Records with missing values (non-numeric or \$null\$) on aggregate fields will not contribute to the aggregate summaries for those fields. However, blanks (user defined missing values) do contribute to the aggregate summaries

and these values should be replaced with \$null\$ (you can use the Filler node) before aggregation. We will begin with an existing stream.

- Open the Stream file **Aggregate.str**, located in the **c:\Train\ModelerIntro** directory
- Run the **Data Audit** node
- Click the **Quality** tab

Figure 9.1 Quality Tab in Data Audit Node

Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method	% Complete	Valid Records
ID	Nominal	--	--	Never	Fixed		100	358
AGE	Continuous	0	0 None	Never	Fixed		100	358
SEX	Flag	--	--	Never	Fixed		100	358
REGION	Nominal	--	--	Never	Fixed		100	358
INCOME	Continuous	0	0 None	Never	Fixed		100	358
MARRIED	Flag	--	--	Never	Fixed		100	358
CHILDREN	Continuous	0	0 None	Never	Fixed		100	358
CAR	Flag	--	--	Never	Fixed		100	358
MORTGAGE	Flag	--	--	Never	Fixed		100	358
STARTDT	Nominal	--	--	Never	Fixed		100	358
ACCOUNT	Nominal	--	--	Never	Fixed		100	358
OPEN BAL	Continuous	8	2 None	Never	Fixed		100	358
CURR BAL	Continuous	8	2 None	Never	Fixed		100	358
OPENDATE	Nominal	--	--	Never	Fixed		100	358
ACCT	Flag	--	--	Never	Fixed		100	358
CUSTREF	Continuous	0	0 None	Never	Fixed		100	358
ACCTREF	Continuous	0	0 None	Never	Fixed		100	358

Note that the data are 100% complete, which means there are no missing data.

- Close the Data Audit output window
- Run the **Table** node

Figure 9.2 Transaction Data (fulldata.txt)

The screenshot shows the 'Table' window in IBM SPSS Modeler. The title bar reads 'Table (17 fields, 358 records)'. The menu bar includes 'File', 'Edit', 'Generate', and various icons. Below the menu is a tab bar with 'Table' selected. The main area displays a grid of data with 20 rows and 9 columns. The columns are labeled: ID, AGE, SEX, REGION, INCOME, MARRIED, CHILDREN, and CAR. The first column contains IDs ranging from 1 to 20. The second column contains ages (e.g., 23, 30, 45, 50, 41, 41, 41, 20, 20, 20, 46, 50, 50, 50, 50, 57, 57, 63, 26, 62, 62). The third column contains sex (MALE or FEMALE). The fourth column contains region (INNER CITY or RURAL). The fifth column contains income values. The sixth column contains marital status (YES or NO). The seventh column contains number of children (0, 1, 0, 0, 2, 0, 0, 1, 1, 1, 0, 3, 1, 1, 0, 0, 0, 0, 0, 0). The eighth column contains car ownership (YES or NO). The 'OK' button is visible at the bottom right of the window.

	ID	AGE	SEX	REGION	INCOME	MARRIED	CHILDREN	CAR
1	ID12701	23	MALE	INNER CITY	18766	YES	0	YES
2	ID12702	30	MALE	RURAL	9915	NO	1	NO
3	ID12703	45	FEMA...	RURAL	21881	NO	0	YES
4	ID12703	45	FEMA...	RURAL	21881	NO	0	YES
5	ID12704	50	MALE	TOWN	46794	YES	2	NO
6	ID12705	41	FEMA...	INNER CITY	20721	YES	0	YES
7	ID12705	41	FEMA...	INNER CITY	20721	YES	0	YES
8	ID12705	41	FEMA...	INNER CITY	20721	YES	0	YES
9	ID12706	20	MALE	INNER CITY	16688	NO	1	NO
10	ID12706	20	MALE	INNER CITY	16688	NO	1	NO
11	ID12706	20	MALE	INNER CITY	16688	NO	1	NO
12	ID12707	46	FEMA...	RURAL	39068	YES	0	YES
13	ID12708	50	FEMA...	INNER CITY	27740	YES	1	YES
14	ID12709	42	MALE	INNER CITY	33584	NO	3	YES
15	ID12710	57	FEMA...	TOWN	19621	YES	1	YES
16	ID12710	57	FEMA...	TOWN	19621	YES	1	YES
17	ID12711	63	FEMA...	INNER CITY	47630	YES	0	NO
18	ID12712	26	FEMA...	INNER CITY	22378	NO	0	YES
19	ID12713	62	FEMA...	RURAL	20837	YES	0	YES
20	ID12713	62	FEMA...	RURAL	20837	YES	0	YES

Note that *ID* is not unique, since a single customer can hold multiple accounts. This is a common phenomenon when dealing with transaction data.

We will aggregate the data using *ID* as the key field and request several summaries, including the total amount originally deposited and the number of accounts for each individual. We will also sort the data to make the data aggregation more efficient.

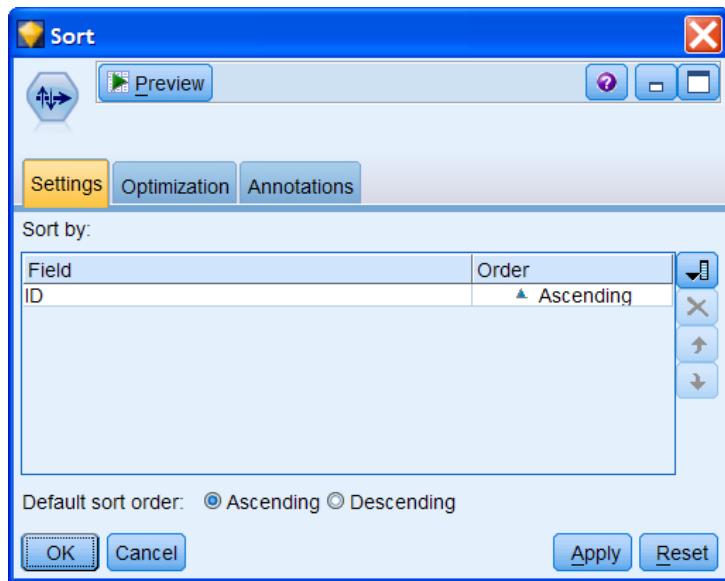
Close the Table window

Place a **Sort** node from the Record Ops palette to the right of the Type node

Connect the **Type** node to the **Sort** node

Edit the **Sort** node

Select **ID** in the **Sort by:** field box

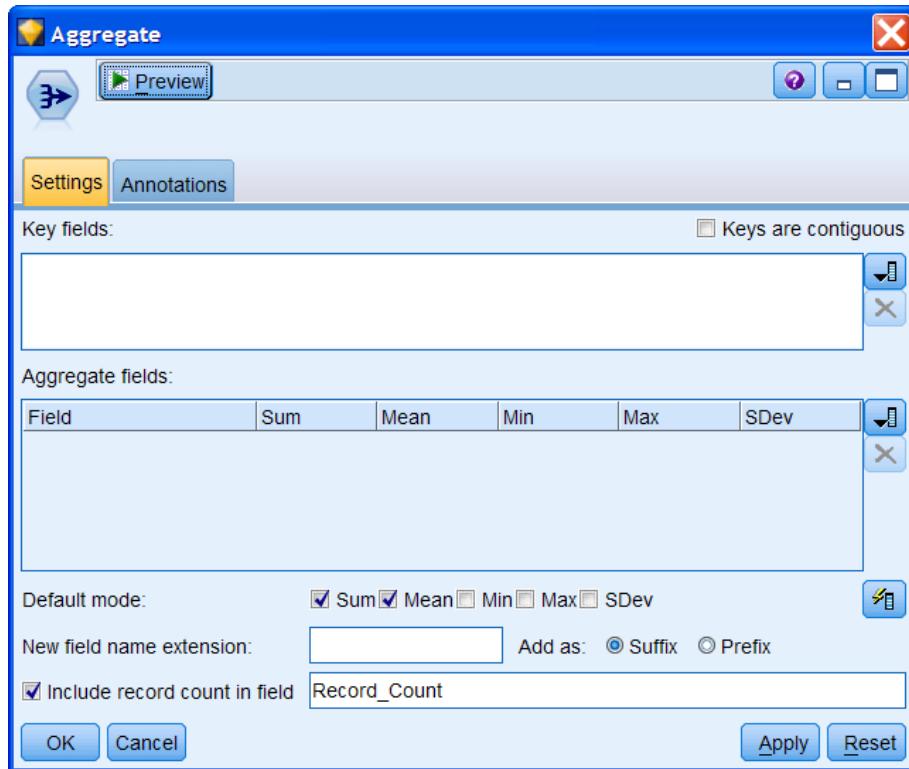
Figure 9.3 Sort Node Dialog

Click **OK** to return to the Stream Canvas

Place an **Aggregate** node from the Record Ops palette to the right of the **Sort** node

Connect the **Sort** node to the **Aggregate** node

Edit the **Aggregate** node

Figure 9.4 Aggregate Node Dialog

As shown in the dialog box in Figure 9.4, the **Key fields** box is used to specify the key(s) for aggregation. If you select multiple key fields, the values of the fields are combined to produce a key value.

If the **Keys are contiguous** check box is selected, values for the key fields will only be treated as equal if they occur in adjacent records. If the data stream is already sorted in order of the key fields, checking **Keys are contiguous** will make the aggregation more efficient.

Fields whose values are to be aggregated are selected on the **Aggregate fields** list. To create aggregate summaries for the selected aggregate field, check one or more of the aggregate mode check boxes. Summaries will be created for each unique combination of key field values. The five available summaries are the Sum, Mean, Min (minimum), Max (maximum), and Sdev (standard deviation).

As a reminder, when aggregating a field, if a null value is found, the aggregation will ignore the null value. However, blanks (user defined missing values) will be included in the calculations and so should be excluded beforehand or converted to nulls.

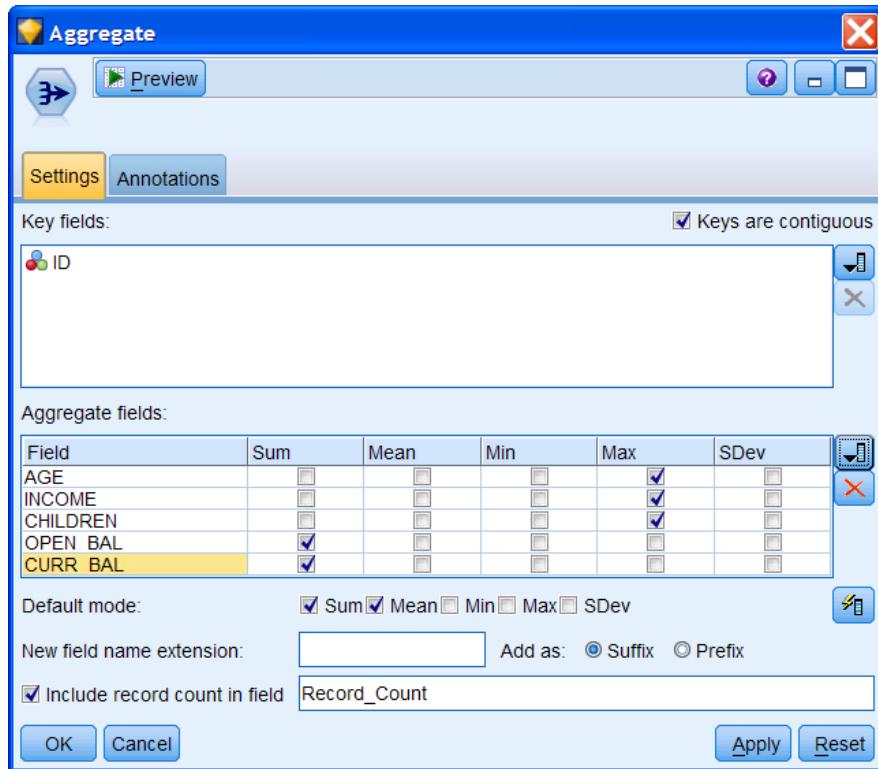
Checking the **Include record count in field** check box will create a new field containing the number of records aggregated to form each output record. This field will be named *Record_Count* by default, but the name can be changed.

On execution, the Aggregate node reads all input records, extracting summary information. Each aggregate record is created with the aggregation key fields and summary values stored in new fields. Each new aggregate field is assigned a name that consists of the aggregate mode name appended (by default) to the original field name (for example, *BALANCE_Sum*). These aggregate records are then passed downstream in no defined order.

If you wish to retain a field value that is constant for all records in an aggregate group (for example, customer's age when aggregating to the customer level from multiple customer account records), simply request the Min, Max, or Mean as the Aggregate mode for that field.

To obtain records representing ID-level aggregates:

- Place **ID** in the **Key fields** list
- Check the **Keys are contiguous** check box
- Select **AGE**, **INCOME** and **CHILDREN** from the Aggregate fields: list
- Check **Max** statistic (deselect **Sum** and **Mean**) for AGE, INCOME and CHILDREN
- Select **OPEN_BAL** and **CURR_BAL** from the Aggregate fields: list
- Check **Sum** statistic (deselect **Mean**) for OPEN_BAL and CURR_BAL
- Check the **Include record count in field** check box (if necessary)

Figure 9.5 Completed Aggregate Node Dialog

Click the **Preview** button

Figure 9.6 Aggregated Data Containing One Record per Customer

Preview from Aggregate Node (7 fields, 10 records)							
	ID	AGE_Max	INCOME_Max	CHILDREN_Max	OPEN_BAL_Sum	CURR_BAL_Sum	Record_Count
1	ID12701	23	18766	0	1000.000	1005.320	1
2	ID12702	30	9915	1	100.000	144.510	1
3	ID12703	45	21881	0	450.000	116.690	2
4	ID12704	50	46794	2	2000.000	2022.020	1
5	ID12705	41	20721	0	3082.000	3241.880	3
6	ID12706	20	16688	1	1844.000	2074.440	3
7	ID12707	46	39068	0	10.000	55.030	1
8	ID12708	50	27740	1	5.000	10.550	1
9	ID12709	42	33584	3	50.000	57.210	1
10	ID12710	57	19621	1	1023.000	1173.680	2

The data have now been restructured so that one record contains the age, income, and number of children for each customer. The sums of the opening and current balance are given, together with the total number of accounts the customer holds. The name of the aggregate mode (summary) function applied to each aggregate field has been appended to the original field name (for example, *AGE_Max*). These names can be changed with a Filter node. Fields in the original data stream that were neither key nor aggregate fields have been dropped from the stream.

These data can now be used for modeling and analysis, or the aggregate data can be merged back into the original file, as we show below.

One limitation of the Aggregate node is that aggregate fields must be continuous. If, for example, the stream contains a field defining purchase type that is categorical, this information will be dropped when performing aggregation. In the next section we will introduce the SetToFlag node as a method of restructuring data held in a categorical field.

9.3 ***Restructuring Set Fields Using the SetToFlag Node***

It may be necessary to convert information held in a set field into a collection of flag fields. For example:

- A file containing one record per purchase may need to be analyzed at a customer level. A field containing a product identifier can be expanded across a number of flag fields indicating whether or not each product was purchased.
- To use a categorical field in regression or discriminant analysis, flags based on the categories (so-called dummy variables) must be used as inputs, as opposed to the original categorical field.

The SetToFlag node located in the Field Ops palette generates flag fields based on set members. On execution, each record received is checked for each value of the selected set field. If found, the appropriate flag field is set to true. The record is then passed down the stream.

The SetToFlag node has an optional aggregate setting that groups the records according to aggregate key field(s). Those flags with at least one true value within the group are set to true, and all others are set to false.

For example, if the data shown in Table 9.3 were passed through a SetToFlag node, with the product field as the set field,

Table 9.4 shows the resulting data if aggregating were not used, while Table 9.5 shows the resulting data if aggregating were used.

Table 9.3 Input Data to SetToFlag Node

ID	AGE	PRODUCT
101	24	Bread
101	24	Milk
101	24	Cheese
102	32	Bread
102	32	Fruit
103	44	Milk

Table 9.4 Output Data from SetToFlag Node Without Aggregate

ID	AGE	PRODUCT	Bread	Milk	Cheese	Fruit
101	24	Bread	T	F	F	F
101	24	Milk	F	T	F	F
101	24	Cheese	F	F	T	F
102	32	Bread	T	F	F	F
102	32	Fruit	F	F	F	T
103	44	Milk	F	T	F	F

Table 9.5 Output Data from SetToFlag Node With Aggregation

ID	Bread	Milk	Cheese	Fruit
101	T	T	T	F
102	T	F	F	T
103	F	T	F	F

We will first demonstrate the SetToFlag node without aggregation, using the set field *ACCOUNT*, which contains four different account types, *CURRENT*, *ISA*, *PEP* and *SAVE*.

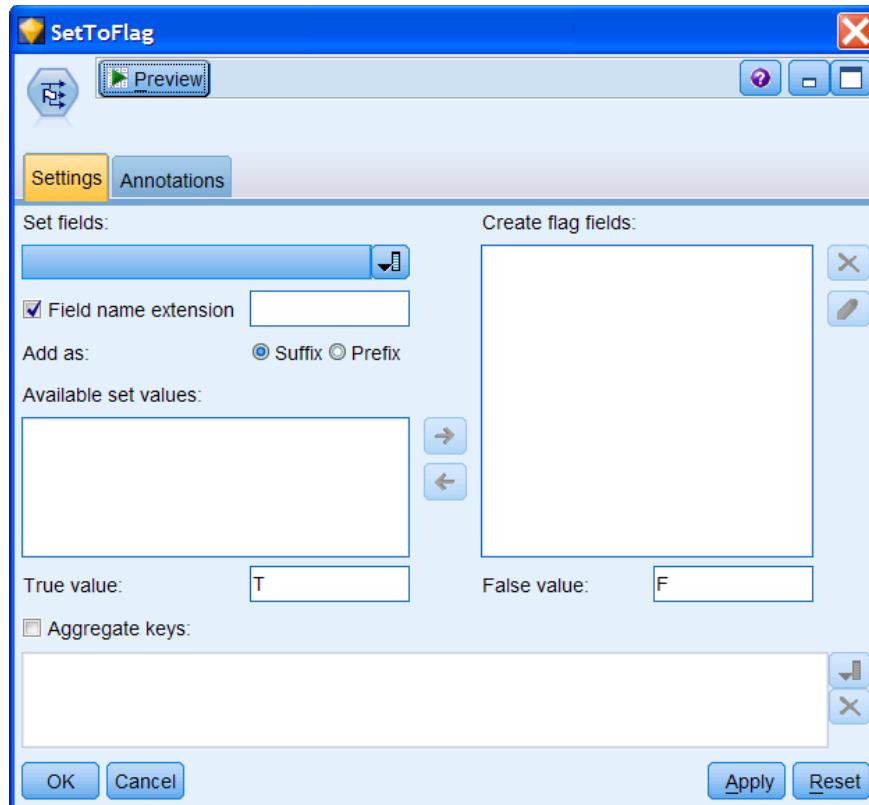
Close the Preview window

Click **OK** to return to the Stream Canvas

Place the **SetToFlag** node from the Field Ops palette to the right of the **Sort** node

Connect the **Sort** node to the **SetToFlag** node

Edit the **SetToFlag** node

Figure 9.7 SetToFlag Dialog

The field to be expanded into a number of flag fields is selected from the *Set fields* list. The *Field name extension* text box allows you to specify an extension that will be added as a suffix or prefix to the new flag field names. By default, new field names are automatically created by combining the original field name with the field value as a suffix; for example, *ACCOUNT_CURRENT*, *ACCOUNT_ISA*, etc. When a field is selected, its members will appear in the *Available set values* list box.

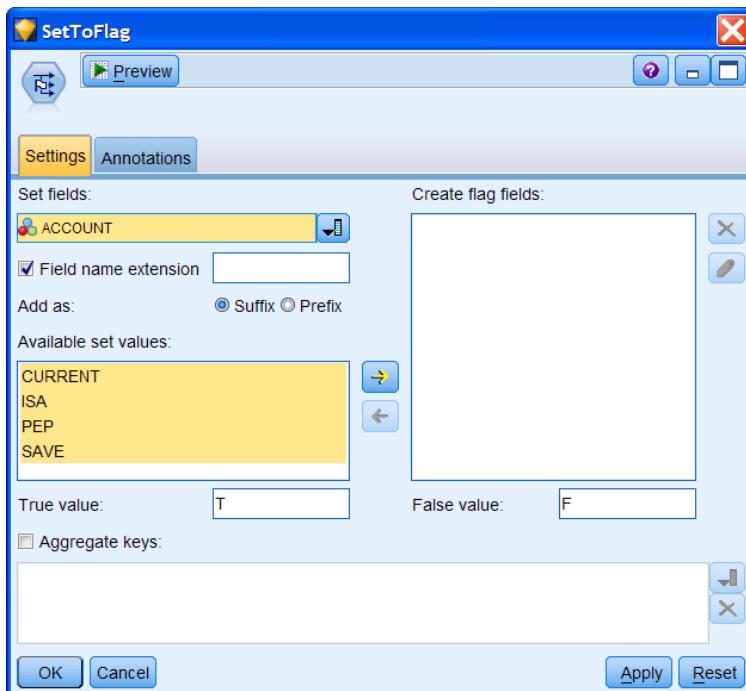
You then place the members for which you wish to create new flag fields in the *Create flag fields* list box. Not all possible flags need to be created.

By default, if the set value is present in a record, the corresponding flag field will be set to the True value (T), otherwise it will be set to the False value (F). These can be changed if desired.

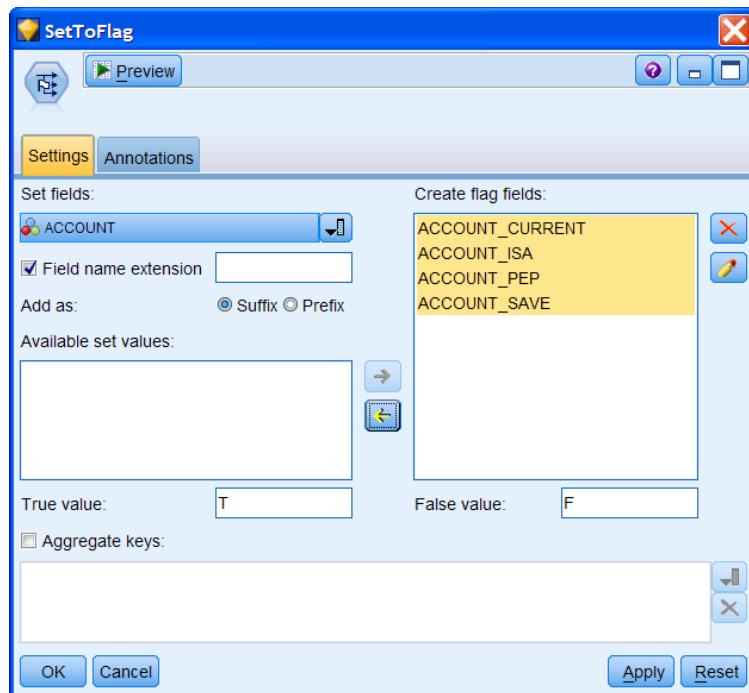
As explained earlier, output records can be aggregated by checking the *Aggregate keys* check box and selecting the appropriate key(s).

Select **ACCOUNT** in the **Set fields:** list. The list of values will appear in the **Available set values** list (Note that all values listed in the Type node are included, whether they exist in the current data file or not)

Figure 9.8 SetToFlag Dialog (Displaying Set Values for ACCOUNT)



Click the button

Figure 9.9 Completed SetToFlag Dialog (Without Aggregation)

Click **OK** to return to the Stream Canvas
 Connect the **SetToFlag** node to a **Filter** node
 Edit the **Filter** node and filter all fields, except **ID**, **ACCOUNT**, and **ACCOUNT_CURRENT** to **ACCOUNT_SAVE** (not shown)
 Click the **Preview** button

Figure 9.10 Data Showing Flag Fields Created by SetToFlag Node without Aggregation

Preview from Filter Node (6 fields, 10 records)

	ID	ACCOUNT	ACCOUNT_CURRENT	ACCOUNT_ISA	ACCOUNT_PEP	ACCOUNT_SAVE
1	ID12701	SAVE	F	F	F	T
2	ID12702	SAVE	F	F	F	T
3	ID12703	SAVE	F	F	F	T
4	ID12703	CURRENT	T	F	F	F
5	ID12704	SAVE	F	F	F	T
6	ID12705	CURRENT	T	F	F	F
7	ID12705	SAVE	F	F	F	T
8	ID12705	CURRENT	T	F	F	F
9	ID12706	SAVE	F	F	F	T
10	ID12706	CURRENT	T	F	F	F

Four flag fields were created. In this file, the **ACCOUNT** field only takes on values of **SAVE** and **CURRENT**, so only the flags representing those values are coded **T** for some records. We now

demonstrate the effect of setting the Aggregate option by returning to the dialog box of the SetToFlag node.

- Close the Preview window
- Click **OK** to return to the Stream Canvas
- Edit the **SetToFlag** node
- Click the **Aggregate keys** check box (not shown)
- Specify **ID** in the **Aggregate keys** field box, and then click **OK**
- Connect a **Sort** node between the **SetToFlag** and **Filter** nodes
- Edit the **Sort** node
- Select **ID** in the **Sort by:** field box
- Click the **Preview** button

Figure 9.11 Data Showing Flag Fields Created by SetToFlag Node with Aggregation

The screenshot shows a software interface titled "Preview from Sort Node (5 fields, 10 records)". The window has a menu bar with File, Edit, Generate, and several icons. Below the menu is a toolbar with icons for Table, Annotations, and other preview options. The main area is a grid table with 10 rows and 5 columns. The columns are labeled: ID, ACCOUNT_CURRENT, ACCOUNT_ISA, ACCOUNT_PEP, and ACCOUNT_SAVE. The data shows multiple rows for each ID, indicating aggregation. The first row has the ID "ID12701" and values F, F, F, T. Subsequent rows show different combinations of IDs and values, such as ID12702 (F, F, F, T), ID12703 (T, F, F, T), etc. An "OK" button is visible at the bottom right of the dialog.

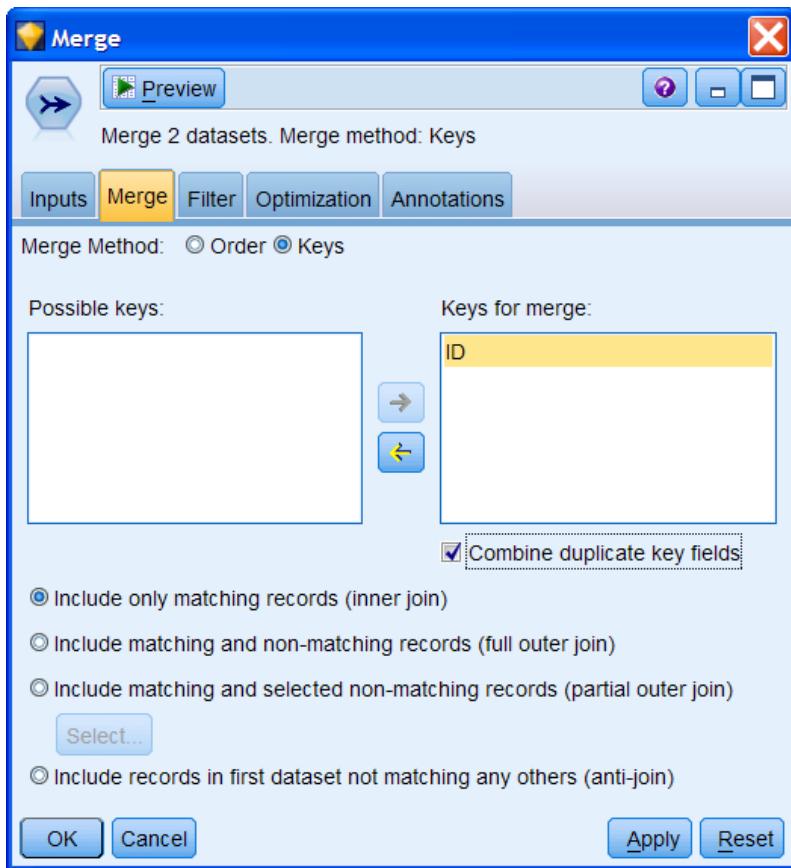
	ID	ACCOUNT_CURRENT	ACCOUNT_ISA	ACCOUNT_PEP	ACCOUNT_SAVE
1	ID12701	F	F	F	T
2	ID12702	F	F	F	T
3	ID12703	T	F	F	T
4	ID12704	F	F	F	T
5	ID12705	T	F	F	T
6	ID12706	T	F	F	T
7	ID12707	F	F	F	T
8	ID12708	F	F	F	T
9	ID12709	F	F	F	T
10	ID12710	T	F	F	F

Now there is only one record per customer. One limitation of using the Aggregate option within the SetToFlag node is that all other fields are dropped from the output stream. If you wish to have the data in this case structure, but also have other fields of interest available to analyze, then the branches of the stream created in the previous two sections must be merged using the Merge node.

9.4 Combining Aggregation and SetToFlag Output

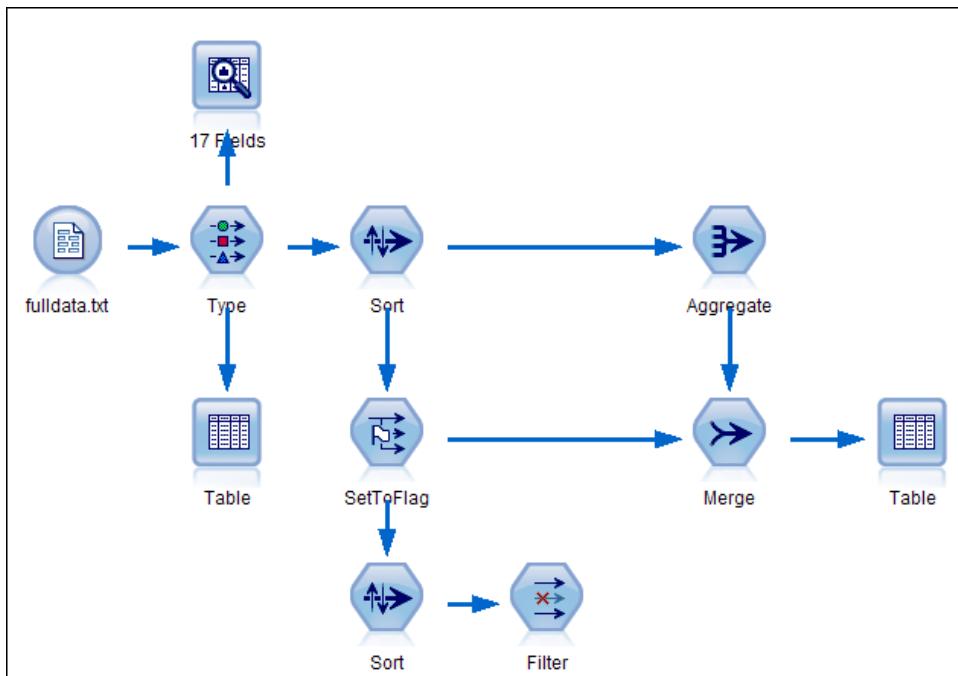
We now demonstrate how the two previous data manipulation exercises may be combined to form a data file that contains not only the output fields from the SetToFlag node with aggregation, but also summary information for the aggregated records.

- Close the Preview window
- Click **OK** to return to the Stream Canvas
- Place a **Merge** node from the Record Ops Palette on the Stream Canvas
- Connect the **Aggregate** node to the **Merge** node
- Connect the **SetToFlag** node to the **Merge** node
- Edit the **Merge** node
- Set the Merge Method to **Keys**
- Select **ID** from the **Possible keys** list and move it to **Keys for merge:**

Figure 9.12 Merge Specifications

Click **OK**

Connect the **Merge** node to a **Table** node

Figure 9.13 Completed Stream (Merging Aggregate and SetToFlag Results)

Run the **Table** node

Figure 9.14 Output Data after Aggregate and SetToFlag Streams Merged

Table (11 fields, 198 records)										
	ID	AGE_Max	INCOME_Max	CHILDREN_Max	OPEN_BAL_Sum	CURR_BAL_Sum	Record_Count	ACCOUNT_CURRENT		
1	ID12701	23	18766	0	1000.000	1005.320	1	F		
2	ID12702	30	9915	1	100.000	144.510	1	F		
3	ID12703	45	21881	0	450.000	116.690	2	T		
4	ID12704	50	46794	2	2000.000	2022.020	1	F		
5	ID12705	41	20721	0	3082.000	3241.880	3	T		
6	ID12706	20	16688	1	1844.000	2074.440	3	T		
7	ID12707	46	39068	0	10.000	55.030	1	F		
8	ID12708	50	27740	1	5.000	10.550	1	F		
9	ID12709	42	33584	3	50.000	57.210	1	F		
10	ID12710	57	19621	1	1023.000	1173.680	2	T		
11	ID12711	63	47630	0	1000.000	1082.270	1	F		
12	ID12712	26	22378	0	1100.000	1123.280	1	T		
13	ID12713	62	20837	0	51.000	158.030	2	T		
14	ID12714	26	23912	0	70.000	133.200	2	T		
15	ID12715	19	8005	1	1153.000	1223.760	2	T		
16	ID12716	44	34961	1	425.000	435.270	1	F		
17	ID12717	32	24627	0	1100.000	1196.720	2	T		
18	ID12718	56	47315	3	210.000	294.390	1	F		
19	ID12719	26	13196	3	506.000	704.080	3	T		
20	ID12720	43	20528	3	1210.000	1349.480	2	T		

9.5 Restructuring Data Using the Restructure Node

The Restructure node functions in much the same way as the SetToFlag node except that it is not limited to just creating flags; you can also create continuous fields using values from other fields. Also, Restructure can be used to restructure Flag as well as Set fields. However, unlike with Set To Flag, you cannot restructure your fields and aggregate all in one step. This is because the Restructure

node does not have its own aggregation option. For that reason, SetToFlag node may be more convenient than Restructure if you are creating Flag fields.

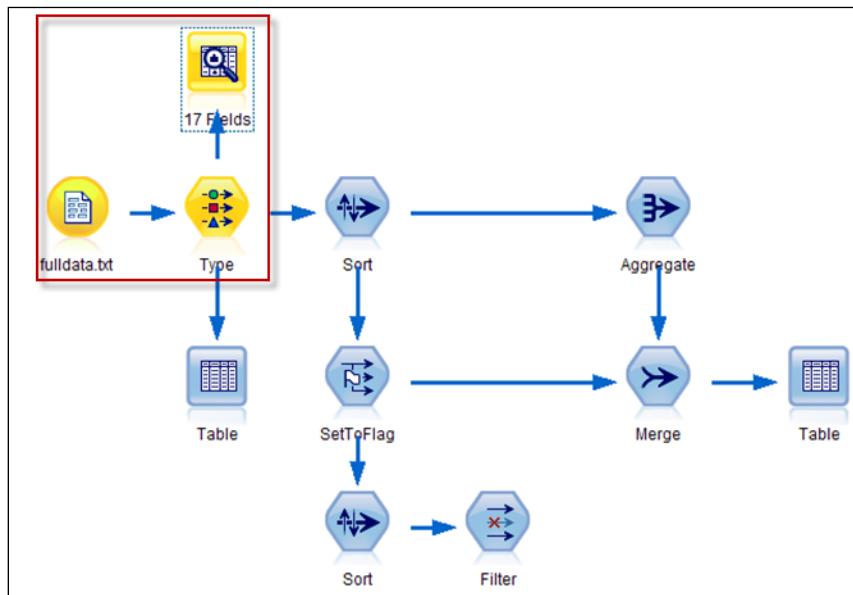
In this example, we will create a field for two of the accounts, *CURRENT* and *SAVE*, but instead of creating flag fields to indicate whether or not a person has these accounts, we will use the value from *CURR_BAL* to show how much money they have in each account.

We will use a subset of the data that has already been modified with the History node so that each customer does not have the same number of monthly balances the History node is discussed in the *Advanced Data manipulation with PASW Modeler* course; one limitation of the History node is that you can specify only one span and offset value, which would not work in this situation because you would end up spanning across observations). We will restructure the file so that the monthly balance for each individual is stored on the same record.

We will also use some of the nodes from the existing stream since we will use the same data file.

Close the Table window
 Use the mouse to **draw a rectangle** around the **three nodes** shown in Figure 9.15 to select them
 Click **Edit...Copy**

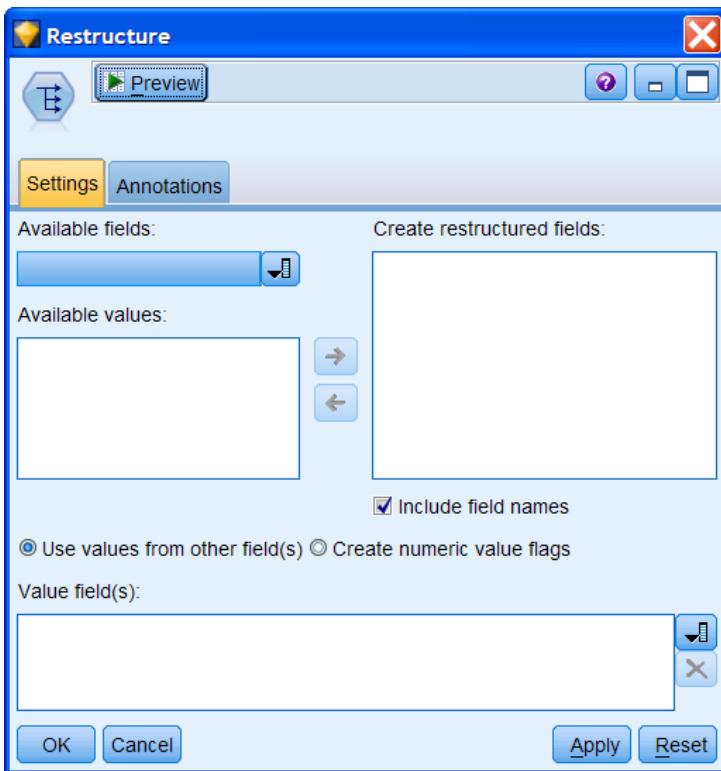
Figure 9.15 Nodes to Select for Copy Operation



Click **File...New Stream**
 Click **Edit...Paste**

These actions will place the three nodes into a new stream.

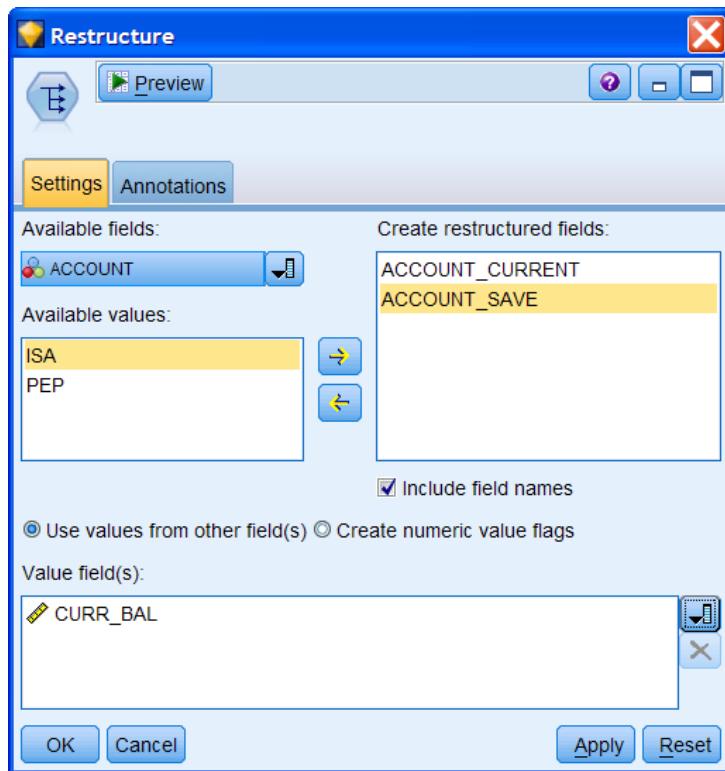
Place a **Restructure** node from the Field ops palette to the right of the Type node
 Connect the **Type** node to the **Restructure** node
 Edit the **Restructure** node

Figure 9.16 Restructure Dialog

The *Available fields:* option lists all the fields that are either *Set* or *Flag*. When a field is selected, the values will be displayed in the *Available values:* box. You can create separate fields for a subset of these values or all of them by moving them into the *Create restructured fields* box. Note that the data must be fully instantiated using an upstream Type node before you can see a list of the available fields and their values. The *Include field names* box should be checked if you want to include the original field name as a prefix for the new fields.

The *Create numeric value flags* option should be checked if you want to create a numeric Flag for each field (0 for false and 1 for true) to indicate in this instance whether or not the person had a current balance that particular month. The *Use values from other field(s)* option should be used instead if you want the person's balance for that month to become the value of the newly restructured field.

- Select **ACCOUNT** from the **Available fields:** list. The list of values will appear in the *Available values:* box
- Move **CURRENT** and **SAVE** into the **Create restructured fields** box
- Check to be sure that the **Use values from other field(s)** box is checked
- Select **CURR_BAL** from the **Value field(s)** list

Figure 9.17 Completed Restructure Dialog

To see how the Restructure node operates we will look at the data in a preview window.

Click the **Preview** button

Scroll to the right to locate the new fields in the preview window

Figure 9.18 Two New Fields Created by the Restructure Node

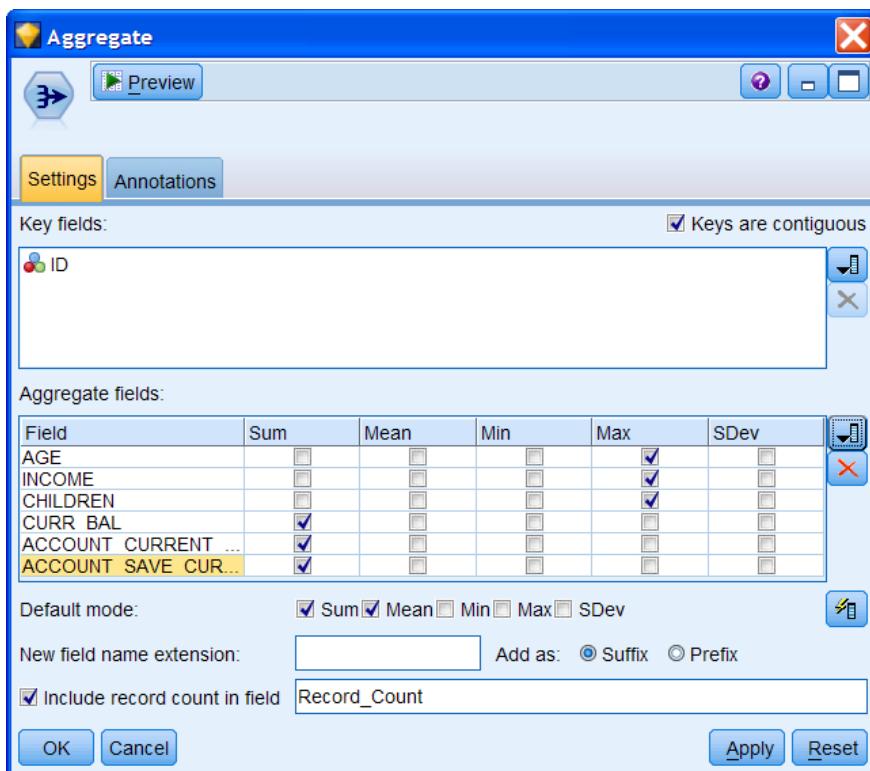
Preview from Restructure Node (19 fields, 10 records)										
File Edit Generate										
Table Annotations										
	ACCOUNT	OPEN_BAL	CURR_BAL	OPEND...	ACCT	CUSTREF	ACCTREF	ACCOUNT_CURRENT_CURR_BAL	ACCOUNT_SAVE_CURR_BAL	
1	SAVE	1000.000	1005.320	11/2/1997	SA	12701	53549	\$null\$	1005.320	
2	SAVE	100.000	144.510	20/5/1997	SA	12702	18333	\$null\$	144.510	
3	SAVE	300.000	321.200	20/7/1997	SA	12703	30343	\$null\$	321.200	
4	CURRENT	150.000	-204.510	23/5/1997	CU	12703	75336	-204.510	\$null\$	
5	SAVE	2000.000	2022.020	7/3/1997	SA	12704	14721	\$null\$	2022.020	
6	SAVE	190.000	287.800	6/9/1997	SA	12705	99830	\$null\$	287.800	
7	CURRENT	2742.000	2762.990	14/5/1997	CU	12705	22554	2762.990	\$null\$	
8	CURRENT	150.000	191.090	12/6/1997	CU	12705	73700	191.090	\$null\$	
9	SAVE	300.000	353.690	31/1/1997	SA	12706	28968	\$null\$	353.690	
10	CURRENT	1412.000	1490.110	4/2/1997	CU	12706	23794	1490.110	\$null\$	

The Restructure node has created two new fields because there were two values (“CURRENT” and “SAVE”) of *ACCOUNT* that we specified. Their names are a combination of the fields and values used to create the field values. In each of these new fields, the value of *CURR_BAL* is placed in the appropriate field, depending on whether the value of *ACCOUNT* for that case was “CURRENT” or “SAVE.” The other field gets the value \$null\$, the missing value for continuous fields.

We can now proceed to the aggregate.

Close the Preview window
Click **OK** to return to the Stream Canvas
Add an **Aggregate** node to the stream
Connect the **Restructure** node to the **Aggregate** node
Edit the **Aggregate** node
Select the **ID** field in the **Key fields** list
Check the **Keys are contiguous** check box
Select **AGE**, **INCOME** and **CHILDREN** from the Aggregate fields: list
Check **Max** statistic (deselect **Sum** and **Mean**) for AGE, INCOME and CHILDREN
Select **CURR_BAL**, **ACCOUNT_CURRENT_CURR_BAL**, and
ACCOUNT_SAVE_CURR_BAL from the Aggregate fields: list
Check **Sum** statistic (deselect **Mean**) for these three balance fields

Figure 9.19 Completed Aggregate Node Dialog



Click the **Preview** button

Figure 9.20 Data Showing New Fields Created by Restructure With Aggregation

Preview from Aggregate Node (8 fields, 10 records)

ID	AGE_Max	INCOME_Max	CHILDREN_Max	CURR_BAL_Sum	ACCOUNT_CURRENT_CURR_BAL_Sum	ACCOUNT_SAVE_CURR_BAL_Sum
1	ID12701	23	18766	0	1005.320	\$null\$
2	ID12702	30	9915	1	144.510	\$null\$
3	ID12703	45	21881	0	116.690	-204.510
4	ID12704	50	46794	2	2022.020	\$null\$
5	ID12705	41	20721	0	3241.880	2954.080
6	ID12706	20	16688	1	2074.440	1720.750
7	ID12707	46	39068	0	55.030	\$null\$
8	ID12708	50	27740	1	10.550	\$null\$
9	ID12709	42	33584	3	57.210	55.030

We now see only one record for each customer. The two fields created by the Restructure node sum to the value of *CURR_BAL_Sum*.

Summary

In this lesson we introduced methods to manipulate the data structure.

You should now be able to:

- Use the Aggregate node to create summary records
- Use the SetToFlag node to expand a set field into a collection of flag fields
- Combine the outputs from the SetToFlag and Aggregate nodes using the Merge node
- Use the Restructure node to as an alternative to the SetToFlag node

Exercises

In this exercise you will restructure the data that was merged in the exercises for Lesson 9 to create a file that has one record per holiday code. Each new record will contain information including total number of holidaymakers, total cost, average length of stay etc.

1. If it is not already loaded, load the stream created in the previous lesson exercise, *ExerLesson8.str*. (If that is unavailable, load the *Backup_ExerChapter08.str* from the course folder.)
2. First, check to see if the data are sorted by *HOLCODE*. Hint: Add a Table node to the Merge node. Are the data sorted?
3. Connect an Aggregate node to the Merge node. Edit the Aggregate node. Use *HOLCODE* as the key field, and create the following aggregate fields:
 - Mean number of people in party (*NUMPARTY*)
 - Total holiday cost (*HOLCOST*)
 - Mean number of nights stayed (*NIGHTS*)
 - Retain the distance to the beach (*DIST_TO_BEECH*)—use Min mode
 - Total number of bookings for this holiday (Include record count).

Use the check box that tells Aggregate that the key values are contiguous.

4. Connect a Table node to the Aggregate node and run this section of the stream. Has the data aggregation worked? How much, in total, has been spent by individuals for their holidays in code CAF3108? How many records/holidays were aggregated to create this information?
5. Use a SetToFlag node (connect to Sort node) to create three new fields in the original merged data that represent whether each customer requested Full board (FB), Half board (HB) or self-catering (SC) in field *ACCOM*. Connect a Table node to verify the results.
6. Edit the SetToFlag node to create these three fields aggregated on *HOLCODE*.
7. Merge the Aggregate and SetToFlag fields in one data stream and review the data with a Table.
8. Save the stream as *ExerLesson9.str*.

Lesson 10: Selecting, Sampling and Partitioning Records

Objectives

- Remove Duplicate records with the Distinct node
- Demonstrate the Sort node
- Use the Select node for data selection
- Demonstrate how to automatically generate a Select node
- Sample records randomly with the Sample node
- Use the Balance node to increase the number of records in certain categories of a field
- Use the Partition node to split the data into separate subsets for the training and testing of models
- Use data caching to speed up data processing and freeze samples

Data

In this lesson we continue to work with the credit risk data (*Risk.txt*) used in previous lessons. The data file contains information concerning the credit rating and financial position of 4117 individuals, along with basic demographic information, such as marital status and gender. We also use the data file *fulldata.txt* which contains one record per account held by customers of a financial organization. The file contains demographic details on the customer, such as income, gender, and marital status; it also has account information, including account type, opening date, and current balance. A third file used is *Credit.sav*, also containing information from a financial institution on its customers.

10.1 Introduction

Before data mining can begin, it may be necessary to drop a number of records from the data file(s), or create subsets of the data. Reasons for this are varied and include:

- Removing duplicate records
- Selecting a sample of records to reduce file size
- Splitting a dataset into two samples, the first of which is used to build a predictive model (training data) and the second is used to validate the model (test data).

In the following sections we show how PASW Modeler may be used to perform these types of data selection and subsetting operations using the Distinct, Select, Sample, and Partition nodes. At the end of the lesson, we show how to use a data cache to store data in memory that is attached to a node. This will enable you to improve speed of processing and to freeze any selected samples.

10.2 Using the Distinct Node to Remove Duplicates

Databases may contain duplicate records that often must be removed before data mining can begin. An example of this is a marketing database in which individuals may appear multiple times with different address or company information. The process of removing duplicate records is often referred to as de-duping the data.

The Distinct node, located in the Record Ops palette, checks for duplicate records and either passes the first distinct record, or all but the first record (the duplicates), along the stream. A duplicate is

defined by the data values on one or more fields that you specify. A record with the same values for all of the selected fields as another is treated as a duplicate. Any number or combination of fields may be used to specify a duplicate.

In this section we illustrate the use of the Distinct node by removing all duplicates of customers from a data file. The result will be a flow of data that contains one record per customer. We begin by reading the data into PASW Modeler.

If the Stream Canvas is not empty, click **File...New Stream**

Select the **Var. File** node and place it on the Stream Canvas

Edit the **Var. File** node and set the file to **fulldata.txt** in the **c:\Train\ModelerIntro** directory

Click the **Preview** button

Figure 10.1 Data with Duplicates

The screenshot shows the 'Preview' window titled 'Preview from fulldata.txt Node (17 fields, 10 records)'. The window has a toolbar with icons for File, Edit, Generate, and various preview options. Below the toolbar is a tab bar with 'Table' selected. The main area displays a table with 10 rows of data. The columns are labeled ID, AGE, SEX, REGION, INCOME, MARRIED, CHILDREN, and CAR. Several records show multiple occurrences of the same ID (e.g., ID12701, ID12702, ID12703, ID12704, ID12705, ID12706). The last row shows two entries for ID12706. At the bottom right of the preview window is an 'OK' button.

ID	AGE	SEX	REGION	INCOME	MARRIED	CHILDREN	CAR
1	ID12701	23	MALE	INNER CITY	18766	YES	0
2	ID12702	30	MALE	RURAL	9915	NO	1
3	ID12703	45	FEMA...	RURAL	21881	NO	0
4	ID12703	45	FEMA...	RURAL	21881	NO	0
5	ID12704	50	MALE	TOWN	46794	YES	2
6	ID12705	41	FEMA...	INNER CITY	20721	YES	0
7	ID12705	41	FEMA...	INNER CITY	20721	YES	0
8	ID12705	41	FEMA...	INNER CITY	20721	YES	0
9	ID12706	20	MALE	INNER CITY	16688	NO	1
10	ID12706	20	MALE	INNER CITY	16688	NO	1

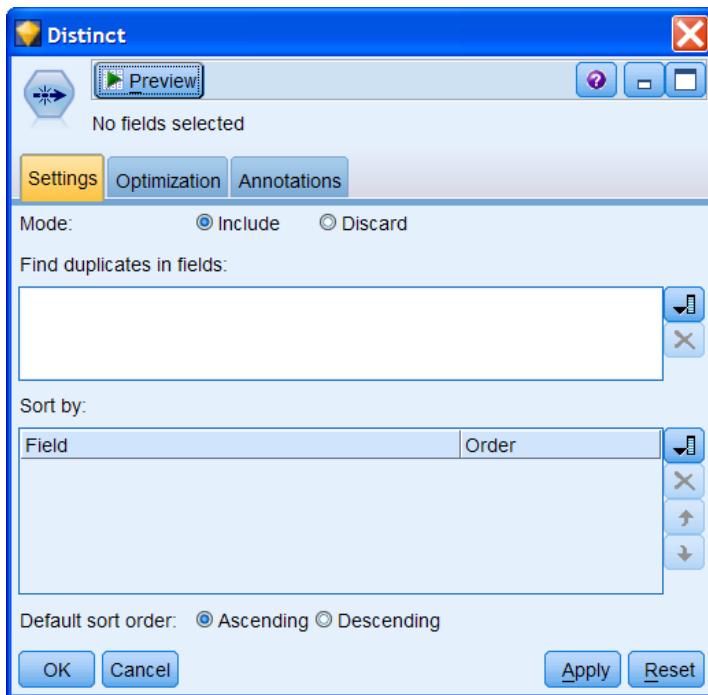
Several records (ID12703, ID12705, ID12706, etc.) occur more than once. We now remove the duplicates and include the first occurrence only of a record (customer).

Close the Preview window

Click **OK** to return to the Stream Canvas

Place a **Distinct** node from the Record Ops palette to the right of the Var. File node, and connect the **Var. File** node to it

Edit the **Distinct** node

Figure 10.2 Distinct Node Dialog

The *Include* and *Discard* options control whether the first distinct record is passed (Include) or all but the first distinct record are passed (Discard). To remove duplicates in a database set the Mode to *Include*; to identify duplicates set the mode to *Discard*. Fields that provide the basis for identification of duplicates are selected in the *Find duplicates in fields* list.

It should be noted that the Distinct node will still function properly even if the data are not sorted by the fields used to determine duplication. That is, the Distinct node will identify the first distinct record it encounters, reading down the records, and either pass this record, or pass all but this first distinct record. However, it is probably best to sort the data beforehand, either by using the Sort node found in the Record Ops palette or by specifying the sort field in the Distinct dialog itself, in the *Sort by:* area. Sorting makes checking the operation of the Distinct node more straightforward. And clearly, if there is a natural ordering to the data such that the first record in a group of duplicates is to be preferred, sorting beforehand is a necessity.

In this example, any record that has the same value in the *ID* field defines a duplicate.

- Select **ID** in the **Find duplicates in fields** field box
- Make certain the Mode: **Include** button is selected
- Click **OK** to return to the Stream Canvas
- Connect the **Distinct** node to a **Table** node
- Run the **Table** node

Scrolling down the resulting data table, it can be seen that ID numbers now are distinct.

Figure 10.3 De-duplicated Data Using the Distinct Node

The screenshot shows a Windows-style application window titled "Table (17 fields, 198 records)". The window has a menu bar with "File", "Edit", "Generate", and other icons. Below the menu is a tab bar with "Table" selected. The main area is a data grid with 20 rows of data. The columns are labeled: ID, AGE, SEX, REGION, INCOME, MARRIED, CHILDREN, and CAR. The first row (ID 12701) is highlighted with a yellow background. The data includes various demographic and socioeconomic information. At the bottom right of the grid is an "OK" button.

ID	AGE	SEX	REGION	INCOME	MARRIED	CHILDREN	CAR	
1	ID12701	23	MALE	INNER CITY	18766	YES	0	YES
2	ID12702	30	MALE	RURAL	9915	NO	1	NO
3	ID12703	45	FEMA...	RURAL	21881	NO	0	YES
4	ID12704	50	MALE	TOWN	46794	YES	2	NO
5	ID12705	41	FEMA...	INNER CITY	20721	YES	0	YES
6	ID12706	20	MALE	INNER CITY	16688	NO	1	NO
7	ID12707	46	FEMA...	RURAL	39068	YES	0	YES
8	ID12708	50	FEMA...	INNER CITY	27740	YES	1	YES
9	ID12709	42	MALE	INNER CITY	33584	NO	3	YES
10	ID12710	57	FEMA...	TOWN	19621	YES	1	YES
11	ID12711	63	FEMA...	INNER CITY	47630	YES	0	NO
12	ID12712	26	FEMA...	INNER CITY	22378	NO	0	YES
13	ID12713	62	FEMA...	RURAL	20837	YES	0	YES
14	ID12714	26	FEMA...	SUBURBAN	23912	YES	0	YES
15	ID12715	19	MALE	RURAL	8005	YES	1	NO
16	ID12716	44	MALE	TOWN	34961	YES	1	NO
17	ID12717	32	FEMA...	INNER CITY	24627	YES	0	YES
18	ID12718	56	FEMA...	RURAL	47315	YES	3	YES
19	ID12719	26	MALE	TOWN	13196	YES	3	NO
20	ID12720	43	FEMA...	TOWN	20528	NO	3	YES

Once the data stream has been de-duped you are ready to work further with the data, including possibly selecting a sample of records to analyze. In the following sections we will discuss a number of ways of selecting or creating subgroups of records from the data.

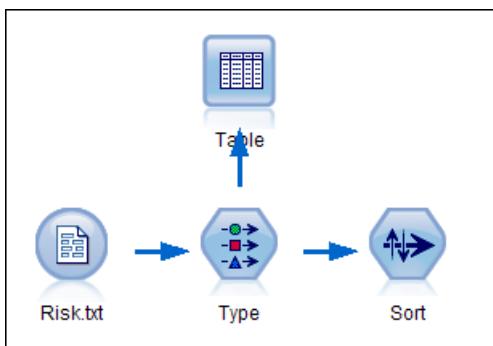
10.3 Sorting Records

Sorting the data records in a file is often done when you have more than one record per unit of analysis (e.g., a customer, student, or organization). You might, for example, receive files from several sources, such as customer transactions over time, and wish to place all the records for a single customer in a contiguous block. This will facilitate reviewing the records and perhaps creating reports.

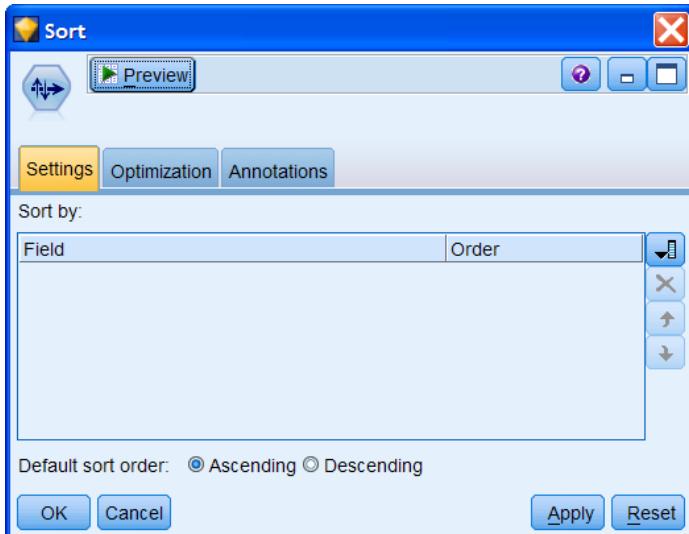
Even when there is only one record per customer or student, you may still wish to sort the file on other fields, such as the date(s) of events.

We will illustrate sorting with the *Risk.txt* data file by sorting the file on the fields *INCOME* and *RISK*. This will place all customers with about the same income together in the file and let us see whether risk seems to be related to income.

- Close the Table window
- Click **File...Open Stream**
- Double-click on **Riskdef.str**
- Place a **Sort** node from the Record Ops palette to the right of the **Type** node
- Connect the **Type** node to the **Sort** node

Figure 10.4 Sort Node Added to Stream

Double-click on the **Sort** node

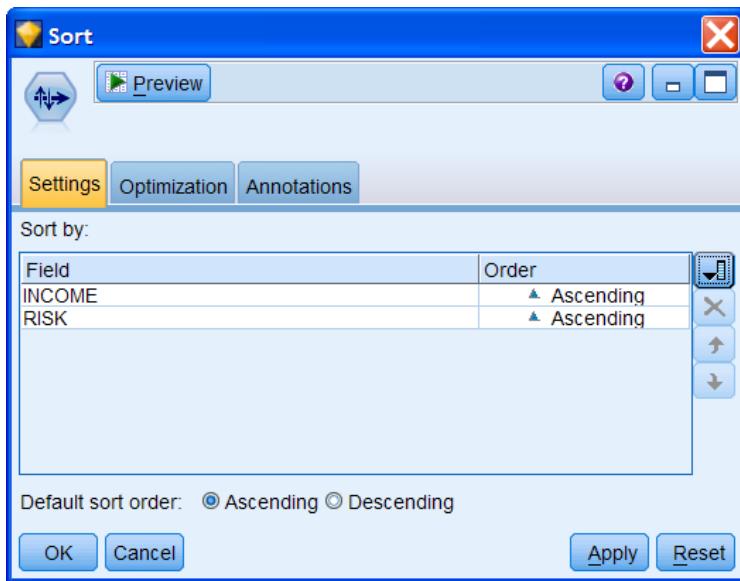
Figure 10.5 Sort Node Settings Tab

There are only a few choices to make in this node. You select one or more fields to sort by, with the order of fields determining the sort order. The first field selected takes precedence, so all data are sorted by that field's values. If a second sort field is selected, the data are sorted by the second field's values *within* values of the first field, and so forth.

Each field can be sorted in either ascending (the default) or descending order, using the arrow in the Order column.

Click the **Field Chooser** button
Select both **INCOME** and **RISK**

We will retain the default ascending order. Since *RISK* is a categorical field with string values rather than numeric values, it will be sorted in alphabetic order. This is functional for our purposes, since the order will be *bad loss*, *bad profit*, and then *good risk*.

Figure 10.6 Specification to Sort File by Income and Risk

The *Optimization* tab allows you to tell PASW Modeler that some of the data are pre-sorted. In large files, this can make an appreciable difference.

Click **OK**

To see the effect of sorting, we can use a Table node to view the records.

Place a **Table** node from the Output palette to the right of the **Sort** node

Connect the **Sort** node to the **Table** node

Run the **Table** node

Increase the width of the Table window so that you can see all the fields

As you can observe in Figure 10.7, the lowest value of *INCOME* in the file is 15005. We can also see that there are very few people with exactly the same value of *INCOME*; as a consequence, sorting the file by *RISK* turns out not to be so critical (but there was no harm in doing so in this small file).

Figure 10.7 Table Displaying Records Sorted by Income and Risk

The screenshot shows a software interface for managing a dataset. The title bar reads "Table (12 fields, 4,117 records) #1". Below the title bar is a menu bar with "File", "Edit", "Generate", and several icons. There are two tabs at the top: "Table" (which is selected) and "Annotations". The main area is a grid of data with 12 columns and approximately 4,117 rows. The columns are labeled: ID, AGE, INCOME, GENDER, MARITAL, NUMKIDS, NUMCARDS, HOWPAID, MORTGAGE, STORECAR, LOANS, and RISK. The first row of data is highlighted in yellow, showing ID 103676, AGE 50, INCOME 15005, GENDER f, MARITAL divsepwid 2, NUMKIDS 6, NUMCARDS weekly, HOWPAID y, MORTGAGE 5, STORECAR 3, LOANS bad profit, and RISK. The "OK" button is visible in the bottom right corner of the grid area.

Looking at the first few dozen records, do you see a pattern in the *RISK* values? That is, are individuals with the lowest incomes more likely to have a specific risk? The answer appears to be yes, that values of *bad profit* are quite common among this group. You can also determine whether these people tend to share other characteristics, such as age or marital status.

Now let's compare them to people who make the highest incomes.

Scroll to the **bottom** of the Table window

Those with the highest incomes definitely have a different distribution of values on *RISK*. Many of these people are categorized as *good risk*, which makes logical sense, given the fact that their incomes are almost 4 times as great.

This pattern comports with our findings in Lesson 7 about the relationship between financial risk and income. And, obviously, using a more rigorous or graphical technique is preferable to simple data sorting, but the intent of this example was solely to introduce you to the Sort node and its features.

Figure 10.8 Records with Highest Incomes

The screenshot shows a Windows-style window titled "Table (12 fields, 4,117 records) #1". The window has a menu bar with "File", "Edit", "Generate", and other options. Below the menu is a toolbar with icons for "Table" (selected), "Annotations", "New", "Open", "Save", "Print", and "Exit". The main area is a grid table with 12 columns labeled: ID, AGE, INCOME, GENDER, MARITAL, NUMKIDS, NUMCARDS, HOWPAID, MORTGAGE, STORECAR, LOANS, and RISK. The data consists of approximately 4,117 rows of individual records. The first few rows are as follows:

ID	AGE	INCOME	GENDER	MARITAL	NUMKIDS	NUMCARDS	HOWPAID	MORTGAGE	STORECAR	LOANS	RISK	
4097	100728	28	57623	m	married	1	1	monthly	y	1	1	bad loss
4098	100567	38	57683	f	married	1	1	monthly	y	2	1	bad loss
4099	100452	35	57689	m	married	1	1	monthly	y	2	1	good risk
4100	100354	32	57718	m	married	1	2	monthly	y	1	1	bad profit
4101	100414	34	58026	m	married	0	1	monthly	y	2	0	good risk
4102	100796	45	58381	m	married	1	1	monthly	y	1	0	good risk
4103	100376	33	58505	f	married	0	2	monthly	y	1	0	good risk
4104	100769	44	58529	m	married	0	1	monthly	y	1	0	bad loss
4105	100698	42	58785	f	married	0	2	monthly	y	1	0	good risk
4106	100695	36	58878	f	married	1	1	monthly	y	1	0	bad profit
4107	100758	38	58914	m	married	0	1	monthly	y	1	1	bad profit
4108	100389	30	59036	m	married	1	1	monthly	y	2	1	good risk
4109	100666	28	59179	m	married	1	1	monthly	y	2	1	bad loss
4110	100319	31	59193	f	married	1	2	monthly	y	1	1	good risk
4111	100702	42	59201	m	married	0	1	monthly	y	2	0	good risk
4112	100657	41	59276	m	married	1	2	monthly	y	1	1	good risk
4113	100590	39	59393	f	married	0	2	monthly	y	1	0	good risk
4114	100416	34	59463	m	married	0	2	monthly	y	1	1	bad loss
4115	100418	34	59508	m	married	1	1	monthly	y	2	1	good risk
4116	100668	35	59692	m	married	1	1	monthly	y	1	0	bad loss
4117	100756	44	59944	m	married	1	2	monthly	y	2	0	good risk

OK

We turn next to the Select node.

10.4 Selecting Records

We select records with the Select node. Data selection can be specified manually by the user; alternatively, a data selection statement can be automatically derived from the Generate menu in an output window. We'll try both methods in this section.

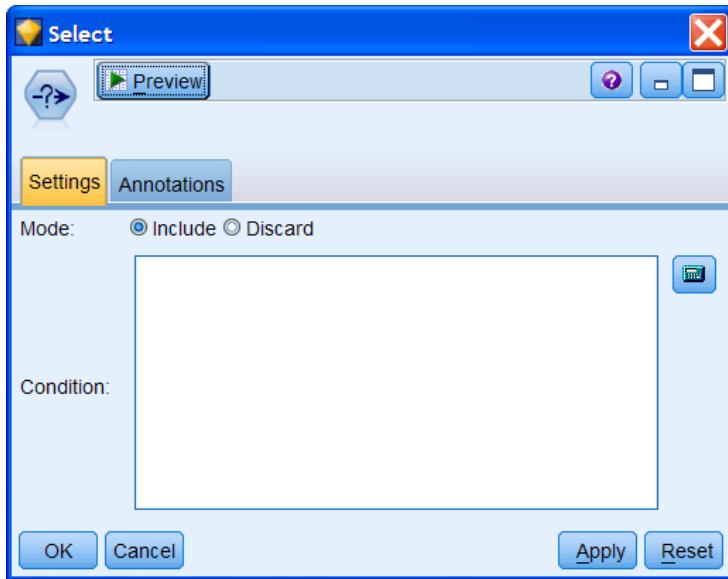
Assume that we are interested in selecting records for which *INCOME* is below 20,000 so that we can examine this group more closely. As we have seen, this group may not be as good financial risks.

Place a **Select** node from the Record Ops palette to the right of the **Type** node
 Connect the **Type** node to the **Select** node
 Right-click the **Select** node, then click **Edit**

Selection is done by specifying a CLEM expression that sets the conditions for record selection or deletion. For example, an expression could be:

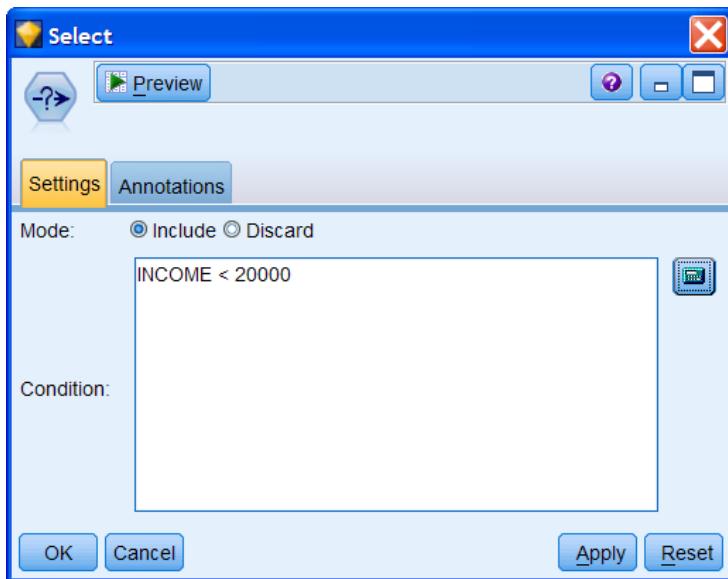
MARITAL="divsepwid"

and this expression would select those individuals who are divorced, separated, or widowed. Expressions can be as complex as necessary. You can either enter a CLEM expression directly in the Condition text box or use the Expression Builder to create the expression.

Figure 10.9 Select Node Setting Tab

The Mode option allows you to choose whether to select (*Include*) or delete (*Discard*) records that satisfy the condition.

Type **INCOME < 20000** in the **Condition** text box (remember that CLEM is case sensitive!)
Check that **Mode:** is set to **Include**

Figure 10.10 Select Expression to Select Records with INCOME < 20000

Click **OK** to return to the Stream Canvas

Tip

If you type the expression in the Condition text box, PASW Modeler doesn't check the CLEM syntax when you click OK. But if you use the Expression builder (even if you type the expression there), PASW Modeler can check the CLEM expression before proceeding.

At this point, nodes downstream of the Select node will analyze only records for which income is below 20,000.

To ensure that the select statement is correct, it is a good idea to connect a Table node to the Select node. Only those records that meet the condition will appear in the table. In this instance the Data Audit or Statistics nodes, which would display the maximum value for *INCOME*, could be used as well.

Place a **Table** node from the Output palette to the right of the **Select** node (or use one of the existing Table nodes)

Connect the **Select** node to the new **Table** node

Run the **Table** node

Figure 10.11 Table Report of Records Where Income is Less than 20,000

The screenshot shows a 'Table' node window titled 'Table (12 fields, 869 records)'. The window has a toolbar with icons for File, Edit, Generate, and various data manipulation tools. Below the toolbar is a tab bar with 'Table' selected. The main area is a grid displaying 20 rows of data. The columns are labeled ID, AGE, INCOME, GENDER, MARITAL, NUMKIDS, NUMCARDS, and HOWPAID. The data shows various demographic and financial details for individuals. An 'OK' button is visible at the bottom right of the window.

ID	AGE	INCOME	GENDER	MARITAL	NUMKIDS	NUMCARDS	HOWPAID
1	102428	23	19990	f	single	0	1
2	103352	29	19950	m	divsepwid	4	6
3	102870	40	19945	f	divsepwid	2	6
4	102129	21	19876	f	single	0	2
5	103601	49	19863	f	divsepwid	2	5
6	100808	18	19847	f	married	2	0
7	102589	36	19811	f	single	1	3
8	102279	22	19809	m	single	0	3
9	103020	42	19779	f	divsepwid	2	5
10	103680	34	19767	m	divsepwid	4	6
11	103775	41	19705	m	married	2	4
12	103027	42	19697	f	divsepwid	4	5
13	103838	43	19696	f	married	3	3
14	103351	46	19688	m	divsepwid	4	6
15	101714	28	19683	m	married	2	2
16	101276	21	19640	f	married	1	1
17	103191	27	19624	m	divsepwid	4	6
18	101572	23	19617	f	married	1	2
19	103675	50	19606	m	divsepwid	2	6
20	103836	43	19599	f	married	2	3

The resulting table contains only 869 records—those with *INCOME* below 20,000, so the Select node was successful (the original *Risk.txt* file was already sorted on *INCOME*).

Next, using a Distribution node, we will compare the distribution of risk for the entire sample to the subgroup with *INCOME* under 20,000.

Close the Table window

Disconnect the **Select** node from the **Table** node

Place a **Distribution** node from the Graphs palette to the right of the **Type** node

Connect the **Type** node to the **Distribution** node

Double-click the **Distribution** node

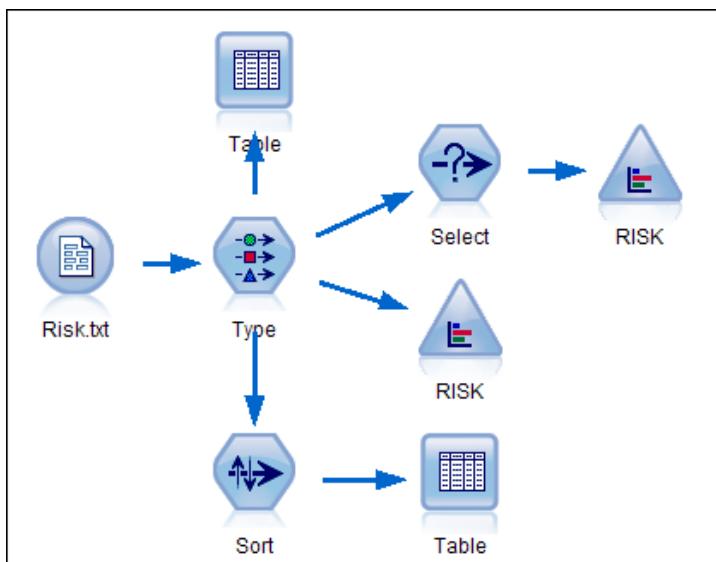
Click the field list button and select **RISK** (not shown)

Click **OK**

Copy the **Distribution** node and **Paste** it to the right of the **Select** node

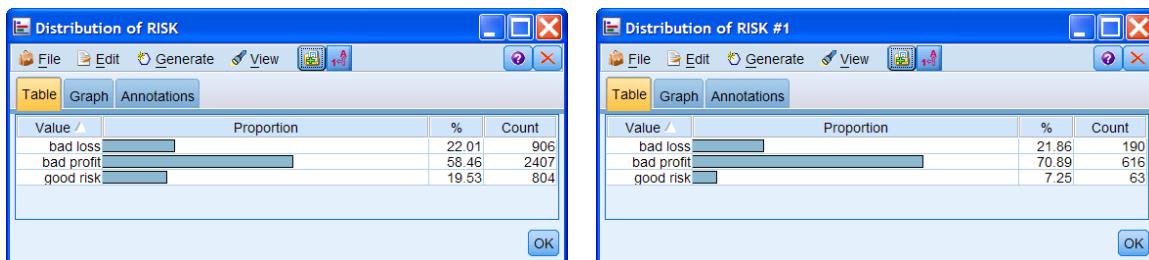
Connect the **Select** node to the pasted **Distribution** node

Figure 10.12 Comparing Credit Risk for the Complete Sample to a Selected Group



Run the two **Distribution** nodes

Figure 10.13 Distribution of Risk for Complete Sample and for those Earning Under 20,000



The distribution plots indicate that in comparison to the complete sample, the subgroup of those who earn below 20,000 contains a smaller proportion of good credit risks and a higher proportion of bad profit risks.

Alternatively, we could have used a second Select node to select those with incomes greater than or equal to 20,000, and used that as a comparison.

Hint

Because it can sometimes be difficult to keep track of which piece of output you are viewing when they are labeled with generic names of "Distribution of RISK" and "Distribution of RISK #1", it can sometimes be helpful to click on the Annotations tab and custom name the output. For example, if we had custom named the Distribution node for all the cases to "Entire Sample," the Distribution graph would have been labeled "Distribution of Entire Sample" instead of "Distribution of RISK."

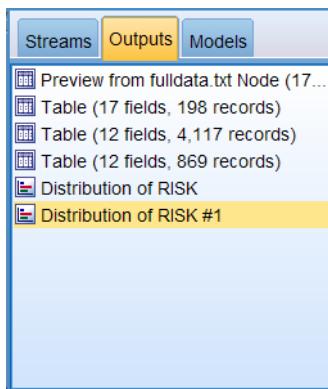
10.5 Automatically Generating a Select Node

As we have just learned, a subset of records can be selected by constructing an appropriate CLEM expression in a Select node. PASW Modeler offers many techniques to automate your work, and several nodes allow you to select sections of output and then generate a Select node directly with no additional effort.

To illustrate this option, let's suppose that we'd like to examine females who are married and have a mortgage. We'll use the Table output to generate the Select node, and we don't need to rerun the Table node because the output we have created has been stored in the Outputs manager.

Click **Outputs** tab in Manager

Figure 10.14 Outputs Manager



All the output that has been created in this PASW Modeler session is stored in the Outputs manager. We can open any output again from here.

Double-click on the **Table (12 fields, 4,117 records)** output

The table opens up again (not shown).

The Generate menu has three options to select records:

- **Select Node ("Records")**. Generates a Select node that selects the specific records for which any cell in the table is selected.
- **Select Node ("And")**. Generates a Select node that selects records containing *all* of the values selected in the table.
- **Select Node ("Or")**. Generates a Select node that selects records containing *any* of the values selected in the table.

We want to use the *Select (And)* option. To do so, we need to click on cells with the appropriate values.

Hold down the Ctrl key, and then click on the value of **MARITAL** in **Record 2**, the value of **MORTGAGE** in **Record 3**, and the value of **GENDER** in **Record 4** (you don't have to pick separate records)

Figure 10.15 Selection of Field Values to Generate a Select Node

The screenshot shows a software interface for managing a dataset. At the top, there's a menu bar with 'File', 'Edit', 'Generate', and other icons. Below the menu is a toolbar with various buttons. The main area is a table titled 'Table (12 fields, 4,117 records)'. It has two tabs: 'Table' and 'Annotations', with 'Annotations' currently selected. The table contains 21 rows of data with 12 columns each. The columns are labeled: ID, AGE, INCOME, GENDER, MARITAL, NUMKIDS, NUMCARDS, HOWPAID, MORTGAGE, STORECAR, LOANS, and RISK. In the highlighted row, the values for MARITAL ('married'), MORTGAGE ('y'), and GENDER ('f') are highlighted with yellow boxes. The 'OK' button is located at the bottom right of the table window.

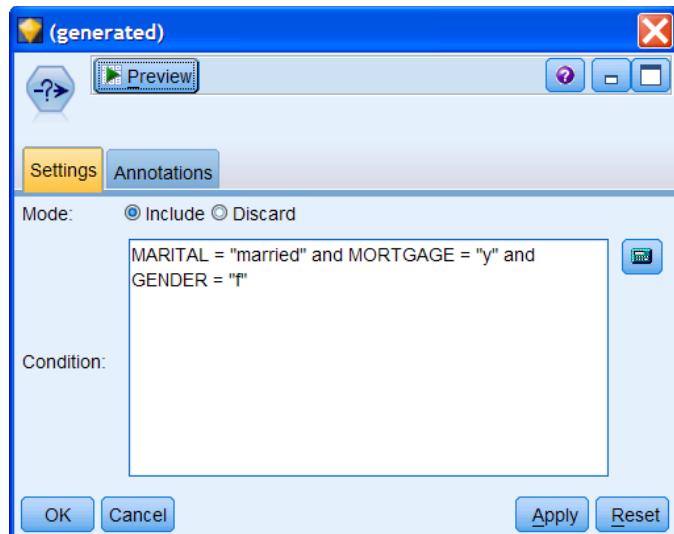
These actions select cells with values of *married* for *MARITAL*, *y* for *MORTGAGE*, and *f* for *GENDER*.

Click **Generate...Select Node ("And")**

Close the Table window

Double-click the new Select node named **(generated)**

The Select node has the correct expression and the correct CLEM syntax. You will probably agree that this was easier than creating the expression in the Select node yourself, and the more complicated the expression, the more time you can save and errors you can avoid.

Figure 10.16 Generated Select Node for Married Females with a Mortgage

At this point, we could attach this node to the stream and study this subset of individuals.

10.6 Using the Sample Node to Select Records

When mining your data, it is often not necessary to use all of the records stored in a data file. It is common in data mining to have hundreds of thousands, if not millions, of records available for processing. Building a successful predictive model, or discovering the majority of associations between fields, can be accomplished quite well with a moderate number of records. In these situations, using all the records can be quite inefficient in terms of processing time and memory. In this section we introduce the Sample node as a way of selecting samples of records from full datasets. The Sample node is also contained in the Record Ops palette.

Data selection and sampling can occur at multiple points in the data mining process. Data selection or sampling often occur at the data collection stage very early in the process, even before any data exploration, so that an overly large dataset is not created from the original data sources.

To illustrate the use of the Sample node, we use it to select a random sample of records from the complete data used in the beginning section, *fulldata.txt*.

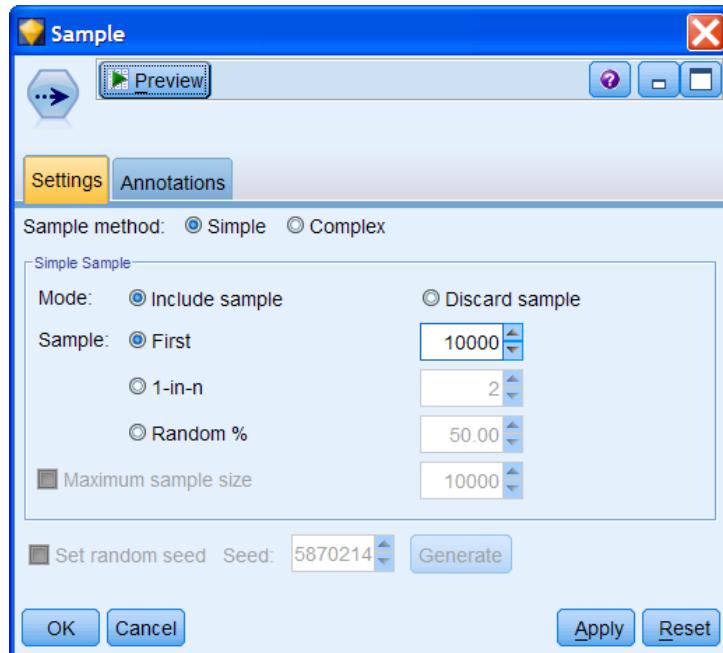
Close the Table window

Click the **Streams** tab in the manager area, and click on **Stream1**

Select a **Sample** node from the Record Ops palette, place it to the right of the Var. File node in the Stream Canvas, and **connect** the nodes

Edit the **Sample** node

Figure 10.17 Sample Node Dialog



The Sample node *Mode* allows the user to either select records (*Include sample*) or eliminate records (*Discard sample*). The maximum size of the sample is entered in the *Maximum sample size* text box.

The node has three possible methods of doing simple sampling:

- *First*: The first n records will be selected (where n is the value in the *First* text box). The *Maximum sample size* option is disabled when *First* and *Include sample* are selected.
- *1-in-n*: every n^{th} record, where n is to be specified in the text box

- *Random %*: a random sample of size $r\%$; the percentage r is to be specified in the text box

Table 10.1 describes each of these three sample settings and their effect when used with the *Include sample* and *Discard sample* modes.

Table 10.1 Effects of Sample Settings on the Sample Node

Sample Type	Include Sample	Discard Sample
First m (m is the specified sample size)	The first m records will be passed along the stream	No records are passed along the stream until m records have been read and discarded
1-in-n (n is specified using the Sample 1 in: option)	Every n^{th} record is passed down the stream beginning with the n^{th} . Only the maximum, m , records will be passed along the stream	Every n^{th} record will be discarded. The maximum sample size setting is ignored.
Random % ($r\%$ is specified using the Random: option)	There is a $r\%$ chance of each record being passed along the stream, thus approximating to a sample of $r\%$ of the total records. Only the maximum, m , records will be passed along the stream.	There is $r\%$ chance of each record being discarded, thus approximating to a discarded sample of $r\%$ of the total records

When a *Random %* sample is requested, you can specify a random seed value (*Set random seed*) that will allow you to reproduce the same sample later. If no random seed is specified, then each time the Sample node is run with *Random %* selected a different random sample will be chosen, making it impossible to exactly replicate earlier analyses. The *Generate* button, when clicked, will generate a new random seed value.

In this example we will select a random sample of approximately 60% of the original data file.

Set Sample: to **Random %**
 Set the **Random %** value to **60**
 Click the **Set random seed** check box
 Type **54321** in the Set random seed text box (not shown)
 Click **OK** to return to the Stream Canvas
 Connect the **Sample** node to a **Table** node
 Run the **Table** node

Figure 10.18 Sampling Approximately 60% of All Records

The screenshot shows a Windows application window titled "Table (17 fields, 210 records)". The window has a menu bar with "File", "Edit", "Generate", and other icons. Below the menu is a toolbar with icons for file operations. The main area is a grid table with 210 rows and 10 columns. The columns are labeled: ID, AGE, SEX, REGION, INCOME, MARRIED, CHILDREN, CAR, and MORTGAGE. The first few rows of data are:

ID	AGE	SEX	REGION	INCOME	MARRIED	CHILDREN	CAR	MORTGAGE	
1	ID12703	45	FEMALE	RURAL	21881	NO	0	YES	NO
2	ID12704	50	MALE	TOWN	46794	YES	2	NO	YES
3	ID12705	41	FEMALE	INNER C...	20721	YES	0	YES	NO
4	ID12706	20	MALE	INNER C...	16688	NO	1	NO	YES
5	ID12706	20	MALE	INNER C...	16688	NO	1	NO	YES
6	ID12708	50	FEMALE	INNER C...	27740	YES	1	YES	YES
7	ID12709	42	MALE	INNER C...	33584	NO	3	YES	NO
8	ID12710	57	FEMALE	TOWN	19621	YES	1	YES	NO
9	ID12710	57	FEMALE	TOWN	19621	YES	1	YES	NO
10	ID12711	63	FEMALE	INNER C...	47630	YES	0	NO	YES
11	ID12714	26	FEMALE	SUBURB...	23912	YES	0	YES	NO
12	ID12715	19	MALE	RURAL	8005	YES	1	NO	NO
13	ID12716	44	MALE	TOWN	34961	YES	1	NO	YES
14	ID12717	32	FEMALE	INNER C...	24627	YES	0	YES	YES
15	ID12717	32	FEMALE	INNER C...	24627	YES	0	YES	YES
16	ID12718	56	FEMALE	RURAL	47315	YES	3	YES	NO
17	ID12719	26	MALE	TOWN	13196	YES	3	NO	NO
18	ID12719	26	MALE	TOWN	13196	YES	3	NO	NO
19	ID12720	43	FEMALE	TOWN	20528	NO	3	YES	NO
20	ID12721	40	MALE	TOWN	37227	NO	1	YES	NO

At the bottom right of the window is an "OK" button.

The node has selected, in this instance, 210 records from the original 358 records. At this stage you are now able to perform analysis on the sample of records by connecting additional nodes to the Sample node.

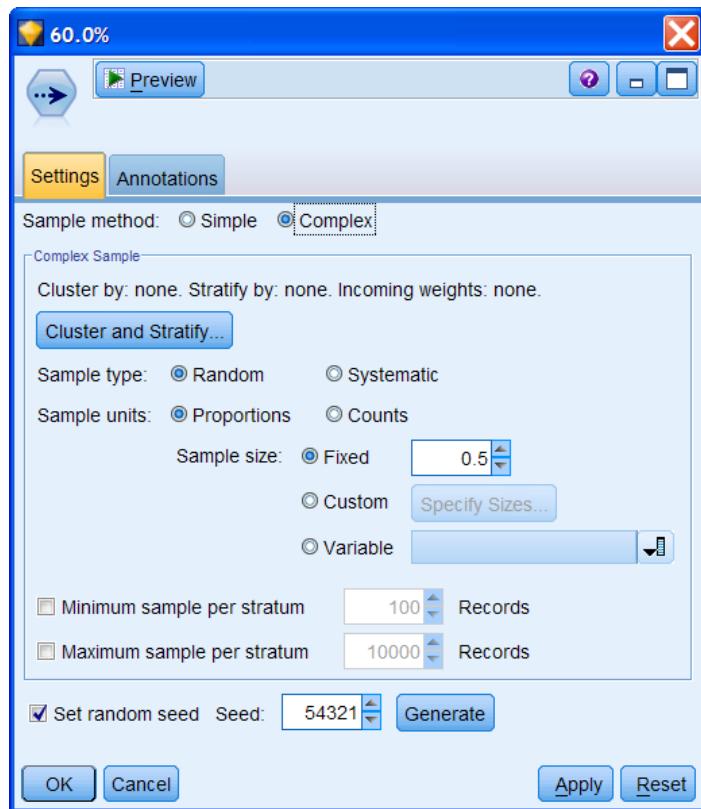
If the *Sample Mode* is set to *Random %*, every time data are passed along a stream, the Sample node reselects its records. You must therefore be aware that each time the stream is run from a point further along in the stream, unless a random seed value was specified, a new set of records is passed along the stream from the Sample node. This may be satisfactory in some circumstances. However, more often it is necessary to “freeze” your sample so that the same sample is used each time the stream is run. Setting the random seed to the same value will produce this result, as will creating a data cache on the node (see section below).

Note

When using the Set random seed option with records read from a database, a Sort node may be required prior to sampling in order to ensure the same result each time the node is run. This is because the random seed depends on the order of records, which is not guaranteed to stay the same in a relational database.

The Sample node can also perform complex sampling. To briefly review these options we'll reopen the node.

- Close the Table window
- Edit the **Sample** node
- Click the **Complex** option button under Sample method

Figure 10.19 Complex Sample Options

PASW Modeler can perform clustered or stratified sampling, or both, on a data source. These types of samples are often created for large surveys to insure proportional representation of elements within a target population, and also for efficiency in data collection. As an example, you can ensure that men and women are sampled in equal proportions, or that every region or socioeconomic group within a population is represented. You can also specify a different sample size for each stratum (if one or more groups have been under- or over-represented in the source data file).

A sample weight field is automatically created when doing complex sampling; it roughly corresponds to the frequency that each sampled unit (record) represents in the original data. The sum of the weight values gives an estimate of the total data file size. For example, if a random 10% sample is taken, the output weight will be 10 for all records, indicating that each sampled record represents roughly ten records in the original data.

Minimum and maximum sample sizes per stratum can be specified, and sample sizes can be listed by absolute number (count) or by proportion of the file.

Note

Many of the predictive modeling nodes in PASW Modeler allow the use of a weight field to indicate that records correspond to more, or less, than a single record or unit in the data. For some types of models, such as decision trees, the use of a weight field doesn't overly complicate model estimation. But for models based on classical statistics, including the Regression and Logistic nodes, the standard errors and significance values based on those standard errors will be incorrect. This is because the equations used in these models must be modified to take into account the complex sample properties,

but there is no option to do so in these nodes. This means that you must exercise caution when using models created with these nodes when you are using data from a complex sample.

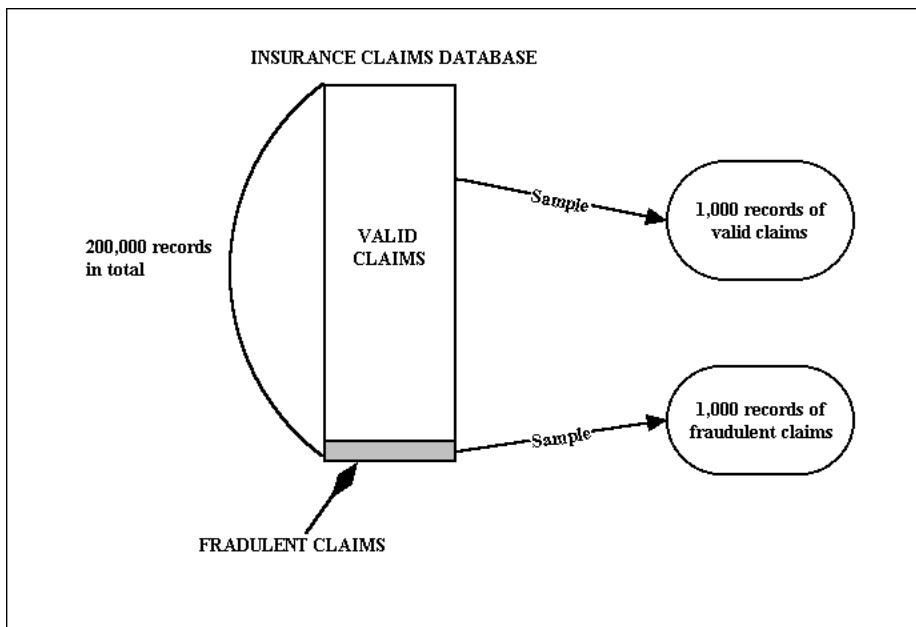
Over-Sampling

Data mining has frequently been used to study outcomes that are relatively rare, whether it be insurance fraud, responses to a direct mailing, or click sequences on a web page. Like any statistical technique, data mining methods typically perform better when there are a reasonable number of cases in all categories of interest. Consider serious insurance fraud for medicare claims from physicians. It may occur in only 1 in 200 claims, so in a database of 10,000 claims there will only be about 50 instances of fraud, but 9,950 valid claims. You don't need a deep knowledge of statistics, just common sense, to see that it will be hard to find good rules to predict cases of fraud, as this is literally searching for a needle in a haystack. Only if the fraud cases are extremely concentrated—all from older males in suburban practices with over 100 patients per day who have often moved their office and see mostly patients from one ethnic group—will this imbalance in the distribution not reduce the efficiency of data mining.

The solution is quite simple: over-sample cases from the rare categories so that in the training data there are proportionally more cases in the category of interest than in the population. Ideally, you might sample roughly equal numbers of cases from all categories to obtain greater precision.

For the insurance fraud example, to include 1,000 cases of fraud would require sampling from 200,000 past records. You would then, from those same 200,000 records, select 1,000 instances of valid claims. This procedure is illustrated in Figure 10.20. Stratified sampling would be used to sample 1,000 cases from each type of claim.

Figure 10.20 Over-Sampling to Increase Cases of Fraud



It isn't necessary that the number of cases in the categories of the target field be equal. In fact, in the situation just described, we might create a target field with a 75/25 split so that only 25% of the cases were fraudulent claims. In that way, more of the valid claims data could be used.

When you over-sample to create the training data, you must create the testing or validation data to match the population distribution of the target field. In the fraud example, the models should be tested on a data file where cases of fraud occur only in 1 in 200 records. Thus, the dataset created for validation will have only 0.5% cases of fraud.

Instead of over-sampling, some modeling techniques can adjust for the known population distribution on the target field. Some techniques do this directly; for others, the adjustment can be made through a matrix of costs of making an error. But in either case, don't expect these methods to necessarily overcome the problem of a small number of cases in one or more outcome categories. Only so much can be done with a limited amount of information, which is why we generally recommend over-sampling when predicting rare outcomes.

To over-sample a data file, you follow these steps:

1. Select records in the category that needs to be over-sampled, and then randomly sample from this group (or use all the records, if necessary)
2. Select records from the other categories, and sample the appropriate fraction of cases
3. Merge the two streams together to create the combined training sample.

Of course, remember that some cases must be left unused for the testing sample.

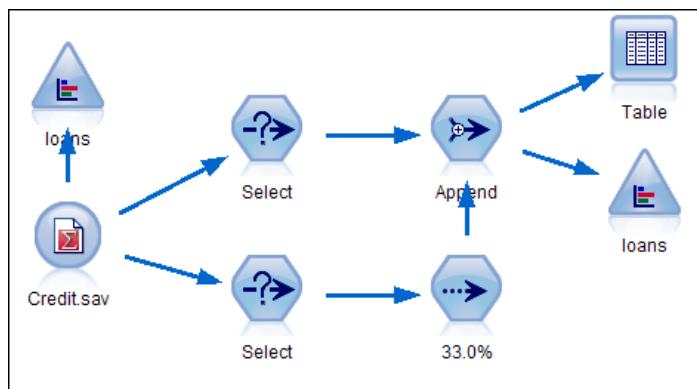
The existing stream *OverSample.str* illustrates this process using the file *Credit.sav*, which is an Statistics data file containing information from a financial institution on its customers. We are interested in understanding the factors that predict whether a customer has a loan or not.

Close the Sample node

Click **File...Open Stream**

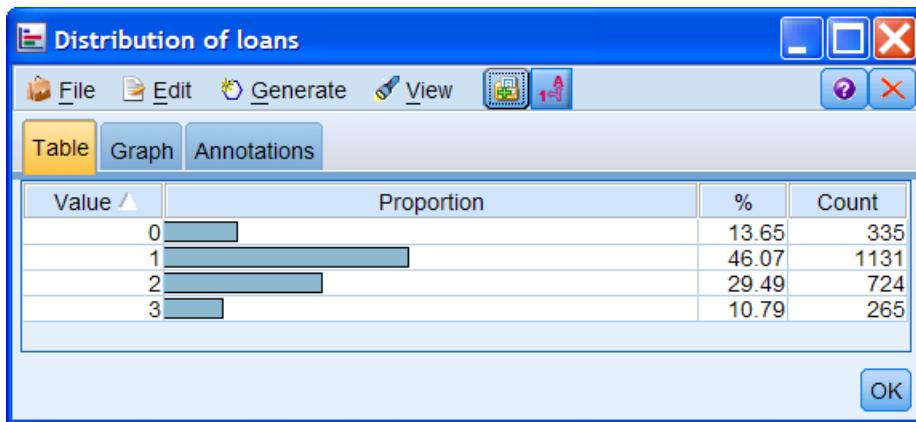
Double-click on the file **OverSample.str** in the c:\Train\ModelerIntro directory

Figure 10.21 OverSample Stream to Select More Customers Without a Loan



Let's review the distribution of the field *loan* that contains a record of the number of loans outstanding for each customer.

Run the Distribution node named **loans** attached to the Source node

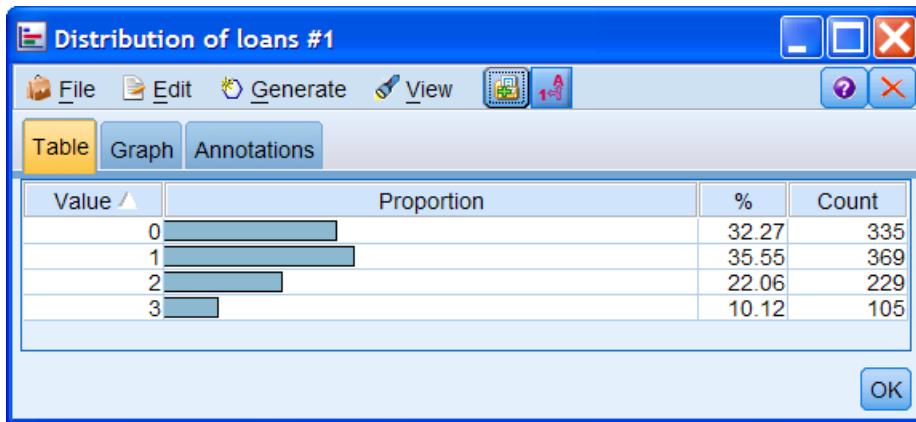
Figure 10.22 Distribution of loans Field

There are 2455 records in the file. Only 13.65% customers have no loans (335 records), and we need to use all these to develop our model. We can, then, randomly sample from the other categories so that we have a more equal proportion of customers with, and without, a loan.

The stream contains two Select nodes. The upper stream selects all customers with *loans* = 0. The bottom stream selects customers with *loans* > 0, and then uses a Sample node to randomly select 33% of this group. An Append node is used to put the two subsets into one file.

To see the result of these actions:

Run the Distribution node for **loans** attached to the Append node

Figure 10.23 Distribution of loans Field After Over-Sampling

We see that all customers with no loans have been retained, but there are fewer customers with 1 or more loans. Now those customers with no loans account for about one-third of the data, which will make it more likely we can construct a model to predict that category with greater accuracy (we would, of course, still need another dataset with the original proportions to validate any model). You can ask your instructor to review this stream in detail if you would like a deeper understanding of how the oversampling is accomplished.

10.7 Balancing Data

In some situations you may not have a large enough data file to over-sample, but the data may still have too few records in certain categories of the outcome field. As an alternative, the Balance node can be used to make the distribution of a categorical field more equal.

Balancing is carried out by duplicating and/or discarding records based on the conditions you specify. Records for which no condition holds are always passed through. The duplication of records is literally that. If a category (condition) has 100 records and it is duplicated by a factor of 3, there will then be 300 records in that category. Discarding records works in reverse by literally dropping some of the records which meet a specified condition.

There is no free lunch, though. You can increase the number of records in a category, such as customers with 0 loans, but you are not increasing the amount of information therein, simply adding copies of existing records. But when a dataset is very small, or the number of records in specific categories is too few, balancing may be the only method that will allow you to develop a reasonable model. As with over-sampling, you must validate any model developed with balanced data on a data file that matches the population distribution of the outcome field.

The Balance node is located in the Record Ops palette, but we won't need to create one from scratch. A Balance node can be generated directly from a Distribution table (or even a histogram, as continuous fields can also be balanced, although this is less common).

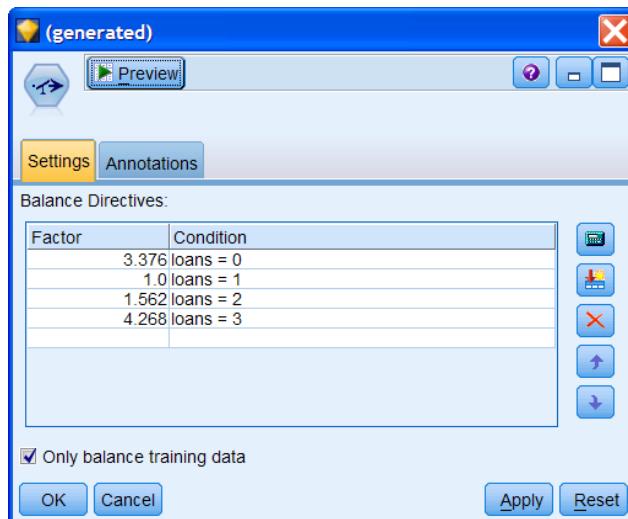
We'll balance the credit data on *loans*.

Run the Distribution node named **loans** attached to the Source node
 In the Distribution chart window, click **Generate...Balance Node (boost)** (not shown)

A node named (*generated*) has been added to the upper left of the Stream Canvas

Attach the (*generated*) node to the **Source** node
 Edit the (*generated*) node

Figure 10.24 Balance Node to Equalize the Categories of loans

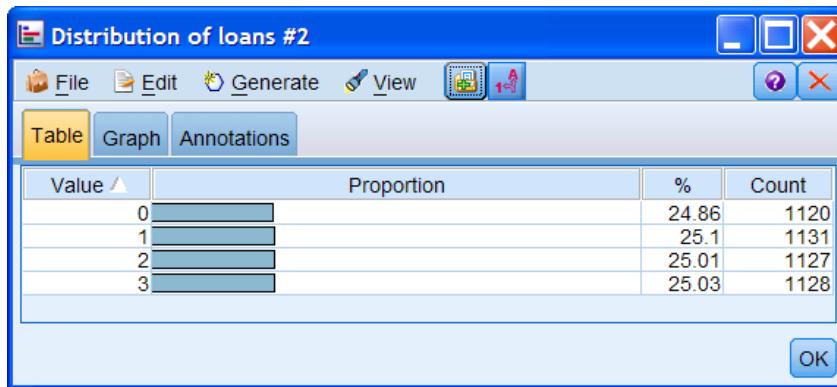


To use a Balance node, you need to specify a *Condition* (such as *loans* = 0) to identify a group of records, and then a *Factor* by which to increase, or decrease, records in this category. Factors greater than 1 increase these records, and factors less than 1 do the reverse. The category of *loans* = 1 had the greatest proportion of customers, so it is the baseline and will not be increased (its Factor is 1.0). All the other categories of *loans* will be increased to match it in size as closely as possible. There is a checkbox, clicked by default, to only balance the training data (this assumes the use of a Partition node; see next section). This option is ignored if there is no partition field.

We can see the result of this Balance node with a Distribution table.

Close the Balance node
Add a **Distribution** node to the stream and attach the **(generated)** node to it
Select **loans** as the Field
Click **Run**

Figure 10.25 Balanced Distribution of loans



The number of records in each category of *loans* is basically equal for all practical purposes. You can always edit the generated Balance node to refine the balancing operation.

Note

If you plan to use classic statistical techniques for modeling, such as linear or logistic regression, you should be aware that the standard errors, and thus the probability values, reported by these techniques will be incorrect on training data that is created with a Balance node using boosting. This is because the standard errors are based on true sample size, but balancing artificially increases the sample size.

10.8 The Partition Node

As was discussed in Lesson 1, in the Modeling stage of the CRISP-DM process, the data are split into training and testing (or validation) samples. Models are developed on the training data, and then tested on the testing data sample. The testing and training datasets should be created randomly from the larger total data file for a project, though usually not in equal sizes.

There are several methods available to create these two data files in PASW Modeler:

1. The Sample node can be used to select two random samples.
2. The Select node can be used with random functions to effectively select cases randomly.
3. The Partition node can be used to create a special partition field that can be used directly by modeling nodes.

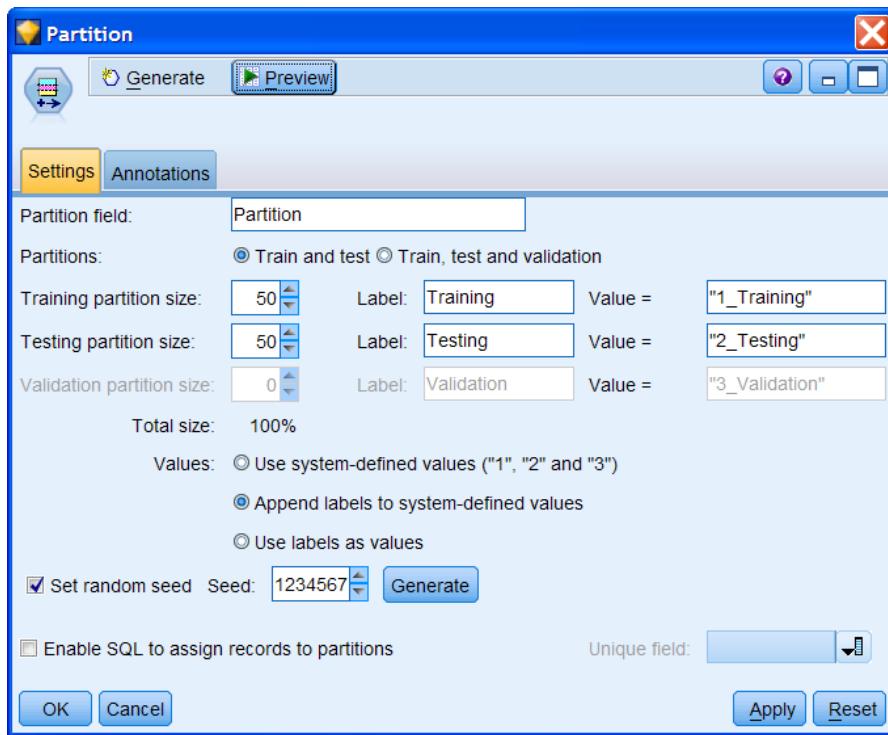
Because almost all data mining projects require training and testing data files, the generation of these samples has been automated in PASW Modeler. Rather than literally create two data files, PASW Modeler allows you to retain one data file that is “split” into two (or three) sets. One set is used for training, another for testing.

The Partition Node, found in the Field Ops palette, generates a partition field that divides the data into separate subsamples for the Training, Testing (and Validation) stages of model building. By default, the node randomly selects 50% of the cases for training purposes and reserves the other 50% for testing the model. These proportions can be altered (for example, 70% for training and 30% for testing, which is a common split).

If you prefer, you can subdivide the cases into three samples instead of just two, one for Training, one for Testing, and one for Validation. The model will still be built with the Training sample and tested with the Testing sample. However, the Testing sample can then be used to help further refine the model. For example, if you believe that the performance of the model on the Testing sample needs improvement, you can alter some of the parameters in the model node, recreate the model using the Training sample and then re-examine the performance with the Testing sample. You may have to do this several times before you are satisfied. Once you are satisfied that the model is the best you can get, the Validation sample, which unlike the Training and Testing samples played no role in developing the final model, is then used to see how well the model performs against yet unseen data.

We need to reopen the *Riskdef* stream we had been using before the last examples.

- Close the Distribution node output
- Click the **Streams** tab in the Manager
- Click on **Riskdef**
- Add a **Partition** node to the stream to the right of the **Type** node
- Connect the **Type** node to the **Partition** node
- Double-click the **Partition** node to edit it

Figure 10.26 Partition Node

The Partition node generates a categorical field with the role set to Partition. The default name of the field is, not surprisingly, *Partition*. To use the partition in an analysis, partitioning must be enabled in a PASW Modeler model node. When the model is created, it will only be developed on the Training set, but predictions will automatically be made for records in the Testing (and Validation) samples as well as the Training records.

The sizes (in percentages) of the three partitions can be set with the controls. The percentages should add to 100%.

The *Values* option provides three different ways to assign values to records in each partition. The default option is to *Append labels to system-defined value*, which assigns the value *1_Training* to Training cases, *2_Testing* to Testing cases, and *3_Validation* to Validation cases. The *Use system-defined values ("1", "2" and "3")* option assigns the same numeric values but without the label. The *Use labels as values* option identifies cases with text labels. You can change the default labels that PASW Modeler supplies.

It is of critical importance that the records in the training and testing samples remain the same from one model execution to another. If, instead, the records in the training sample could be in the testing sample when a model was later tested, that would defeat the whole purpose of having separate data files.

This possibility arises because PASW Modeler could use an internally generated random number every time the Partition node is run, and that would result in different assignments of cases to the training and testing datasets. To avoid this problem, PASW Modeler uses a default random seed (1234567) to generate random numbers.

The *Set random seed* option is used to ensure that the same records will be assigned to the Training and Testing (and Validation) samples each time the node is run. You can change the seed either by hand or by clicking the *Generate* button, which will generate a new seed. The crucial point is to continue to use the same seed so that the records are assigned to the same set (training, testing, or validation) each time the stream is run.

We will make only one change in this dialog box, which is to change the proportion of cases in the training and testing samples.

Change Training partition size to **70**

Change Testing partition size to **30**

Click **OK**

To demonstrate the effect of the Partition node, we'll request a Table to look at the records.

Add a **Table** node to the stream to the right of the **Partition** node

Connect the **Partition** node to the **Table** node

Run the **Table** node, then scroll to the last column

Figure 10.27 Table Output With Partition Field

The screenshot shows a Table window with the title "Table (13 fields, 4,117 records)". The window has a menu bar with File, Edit, Generate, and various icons. Below the menu is a toolbar with icons for file operations. The main area is a grid with 13 columns and approximately 20 rows of data. The columns are labeled: JMCARDS, HOWPAID, MORTGAGE, STORECAR, LOANS, RISK, and Partition. The Partition column contains values like "1_Training" and "2_Testing". At the bottom of the window is a scroll bar and an "OK" button.

The new *Partition* field has been added to the data and has values of *1_Training* and *2_Testing*. It appears that the first value is more common, which it should be given the split we specified above.

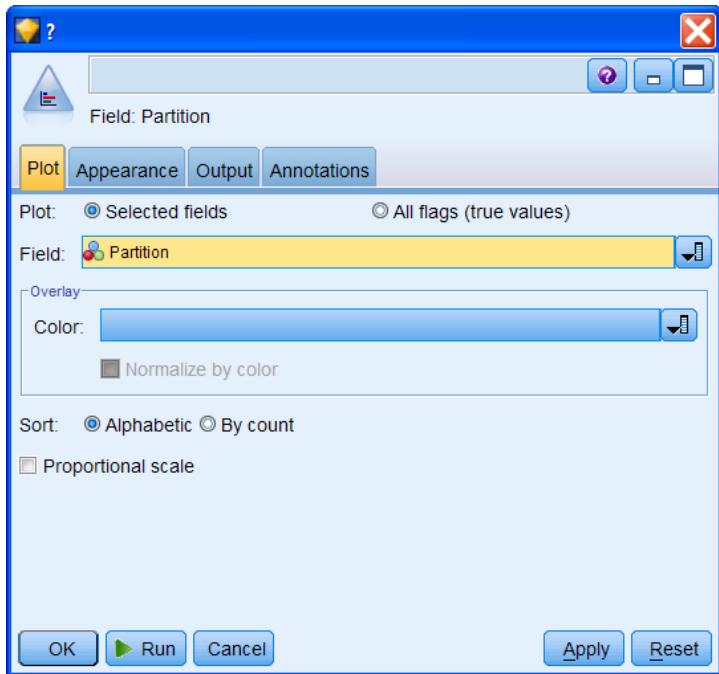
We can check the distribution of the *Partition* field with a Distribution node.

Close the Table window

Add a **Distribution** node from the Graphs palette to the stream near the Partition node

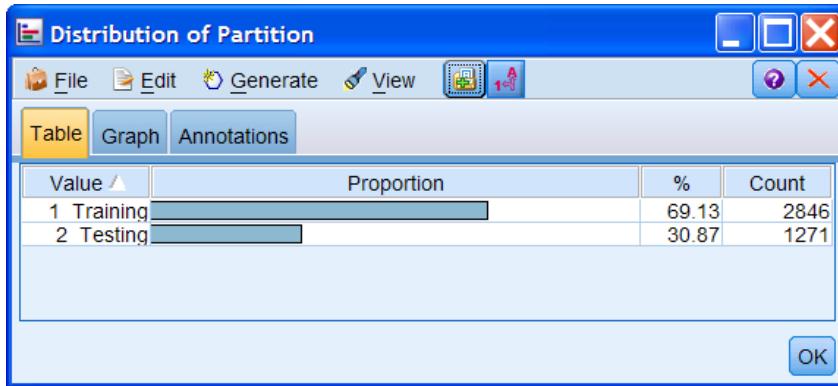
Connect the **Partition** node to the **Distribution** node

Select **Partition** as the field

Figure 10.28 Partition Field Selected in Distribution Node

Click **Run**

We requested a 70/30 split, and although the distribution isn't exactly this (because of random variation), it is quite close.

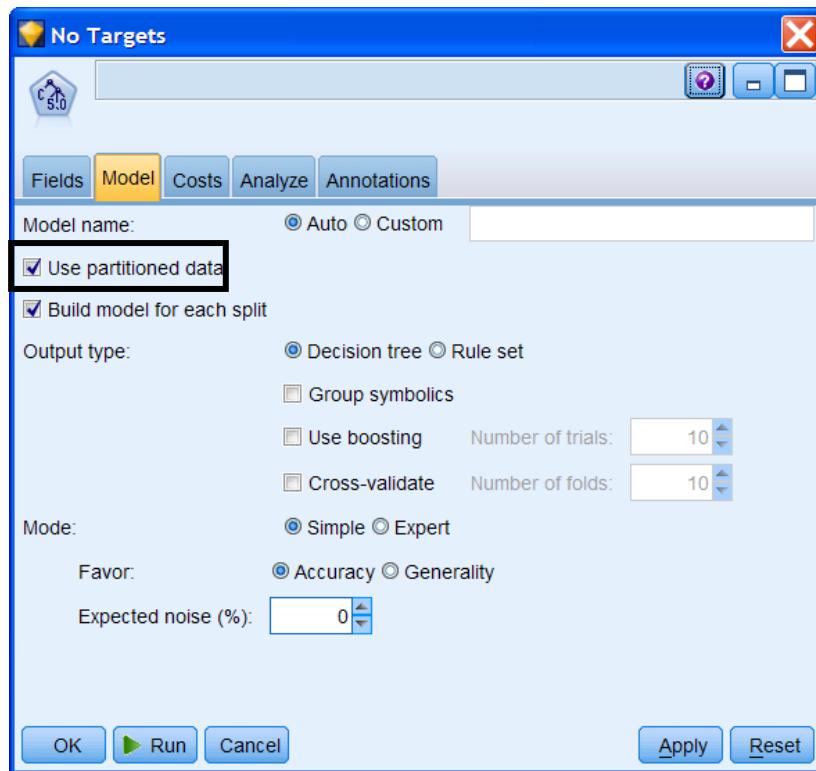
Figure 10.29 Distribution of Partition Field

As a final step, although we won't develop any models until Lesson 12, let's add a C5.0 model to the stream to see briefly how models recognize the *Partition* field.

Add a **C5.0** modeling node from the Modeling palette to the stream

Connect the **Partition** node to the **C5.0** node

Double-click the **C5.0** node to edit it

Figure 10.30 C5.0 Node with Default of Using Partitioned Data

There is a check box, selected by default, to use partitioned data. PASW Modeler then looks for a field with the role *Partition* and uses that field during modeling (if the field had a different role or name, it can be specified instead in the Fields tab dialog).

10.9 Data Caching

When a stream is run in PASW Modeler, the node at the point where the stream is run can be thought of as pulling the data from the previous node. This pulling of data continues upstream until the data are pulled from the source node. This process repeats every time the stream is run. Enabling a data cache provides a way of preventing the repetition of data preprocessing.

When a node carries a cache (specified by clicking *Cache....Enable* from the context menu on a node), the next time data are passed through the node they are stored in the cache. Nodes with caching enabled are displayed with a small document icon at the top right corner. When the data are cached at the node, the document icon is green.

Cached data are held in memory. Once the execution is completed, the cache is full and, from that point on, data are read from the cache and not from the source node. If your stream has many data manipulation nodes or is reading data from a text file, or aggregates data, caching after these nodes can substantially improve performance, albeit at the cost of increased memory usage.

Note that if changes are made to the stream upstream of a cached node, the cache will become flushed or emptied, as its contents will no longer be valid. However, the cache will refill when the stream containing the cached node is next run.

The main uses of data caches are:

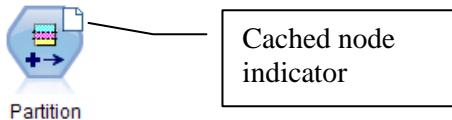
- Improve speed since they avoid repetitions of pre-processing
- Freeze samples, for example samples selected using a random function in the Derive or Partition nodes

To illustrate the use of data caches in freezing samples we will enable a cache on the Partition node

Close the Distribution window
Edit the **Partition** node
Click **Set random seed** check box to deselect this option
Click **OK**
Right-click the **Partition** node
Select **Cache...Enable** from the context menu

The cache is now enabled on the node, indicated by a small file icon on the node.

Figure 10.31 Partition Node with an Enabled Cache



To fill the cache the stream needs to be run from a downstream node.

Run the **Distribution** node connected to the Partition node
Close the Distribution output window

The small file icon on the Partition node should now be green in color, indicating that it is full of data. From this point on, if a stream is connected from this node and run, the data will be read from this Partition node cache and not from the source node. The stream can be run again and the resulting graph compared with the original to show that the identical Partition field is generated. This means that the data are cached as they are output from the node at which caching is done. If caching is done at a Partition node, the data are cached after the partition field is created.

To empty a previously filled cache, right-click the node and click *Cache...Flush*. A flushed cache will refill when data are next passed through the node. To disable the cache, select *Cache...Disable* from the node's Context menu.

The data in a data cache are saved in an Statistics data file format. A saved cache can be later restored, which again avoids any preprocessing steps performed on the data. Alternatively, the Statistics File node can read data saved from the cache. This permits data in a cache to be the data source of a different stream. The file extension for Statistics data files is *.sav. To save a data cache:

Ensure the **cache** is enabled and full
Right-click the **Partition** node
Select **Cache...Save Cache** from the context menu
Navigate to the **c:\Train\ModeleIntro** directory (if necessary)
Type **cachefile** in the **File Name:** text box (not shown)
Click **Save** to return to the Stream Canvas

A saved data cache can be read into PASW Modeler and used in a later stream through the Statistics File node in the Sources palette. (If you do so, note that the *Read labels as names* option button in the Statistics File node should be checked since PASW Modeler stores the field names as Statistics variable labels when saving a data cache.) In addition, data in a saved cache can be restored to the node at which caching was originally done. We demonstrate this latter variation.

Right-click the **Partition** node in the Stream Canvas

Click **Cache...Load Cache**

Navigate to the **c:\Train\Modeler\Intro** directory and click **cachefile.sav** in the file list (not shown)

Clicking the Open button will restore the data in the saved cache to the Partition node. Although slower, an alternative to saving a cache would be to use the File node on the Output palette to write a text copy of the file after preprocessing operations are complete and use the generated source node (Var. File) from this operation to read this data into a new stream. The generated Var. File node, in turn, could be cached to improve performance.

Click **Cancel** to close the Open cache dialog

Summary

In this lesson you learned about sorting, data selection, and the partitioning of data for modeling. You should be able to:

- Use the Distinct node to locate duplicate records
- Sort records
- Select records
- Sample records
- Create a partition field to create training and testing subsets
- Use caches to freeze data samples

Exercises

In these exercises, we will use the charity data file.

1. Import data from *charity.sav* (Statistics File; Read names and labels; Read labels as data).
2. First, we will sort the file on *Pre-campaign expenditure (ORISPEND)* in descending order. Connect a Sort node to the data source node; and then a Table node to the Sort node. After editing the Sort node, run the Table node. You should note the large number of responders in the first few cases (high expenditures). What other characteristics might you note for those with expenditures of over 500?
3. Connect a Select node to the Sort node. Select all Female Responders. (Hint: use the Expression Builder.) Test the node to verify that it works. Connect a Matrix node and run a matrix table to check your selection. How many records were selected?
4. Connect a Partition node (Field Ops palette) directly to the data source node. Request a 70% training sample and a 30% testing sample. Connect a Means node to the Partition node; use *Partition* as the group field and request statistics for *Pre-campaign Expenditure (ORISPEND)*. In this manner, you can verify that the two groups have been randomly selected and have similar distributions.
5. *For those with extra time:* Connect a second Select node (to the data source node) to select Females 50 years of age and older. Run a distribution graph on *Response* for this group and another for the entire file. How do they compare?

In a later lesson we will use the Partition node with a modeling procedure.

Lesson 11: Modeling Techniques in PASW Modeler

Objectives

- Provide a brief introduction to the modeling techniques available in PASW Modeler
- Discuss when and why to use a technique and its important features

Data

No data required.

11.1 Introduction

PASW Modeler includes a number of machine learning and statistical modeling techniques. Although there are different taxonomies, these techniques can be classified into three main approaches to modeling:

- Predictive
- Clustering
- Associations

In predictive modeling, sometimes referred to as *supervised learning*, inputs are used to predict the values for a target field. PASW Modeler has many predictive modeling nodes available, some of which are popular data mining methods while others come from classic statistics. They include: neural networks, five different rule induction methods, support vector machine, Bayes networks, the self-learning response model, a sequence detection method, regression and logistic regression analysis, discriminant analysis, Cox regression, and generalized linear models. In addition, time series analysis is available using either ARIMA or exponential smoothing.

The different clustering methods, sometimes referred to as unsupervised learning methods, have no concept of a target field. The aim of clustering techniques is to try to segment the data into groups of cases/records that have similar patterns of input fields. PASW Modeler has three clustering techniques available: Kohonen networks, k-means clustering, and two-step clustering.

Association techniques can be thought of as generalized predictive modeling. Here the fields within the data can act as both inputs and target fields. Association rules try to associate a particular conclusion with a set of conditions. There are three association techniques available within PASW Modeler: Apriori, Carma and Sequence. The Sequence node (mentioned above) will look for association rules over time (i.e., sequences). These could be stages in processing customer service problems, or web pages visited during a visit to a website that led to a product inquiry.

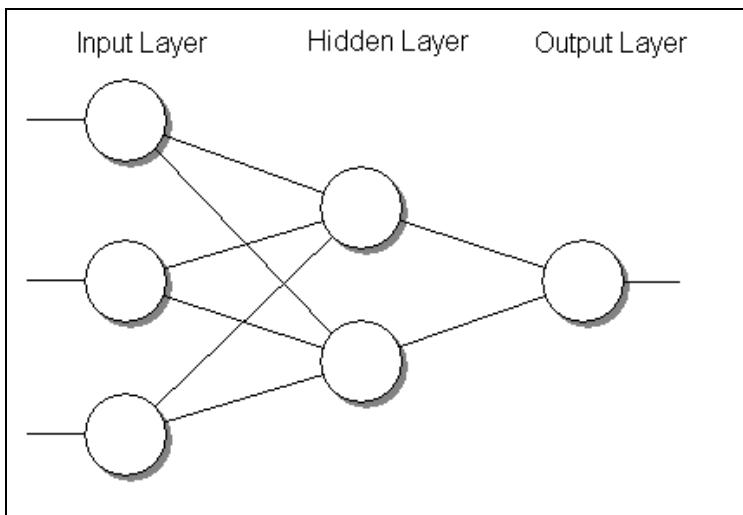
In the following sections we will briefly introduce some of these techniques. More detail will be given to machine learning techniques than to statistical techniques, since the latter methods are more likely to be familiar to you. It should be stressed at this stage that the power of PASW Modeler is that models can be built and results obtained without having to deeply understand the various techniques. We will, therefore, not be describing in great detail how each of the different methods works, just providing a brief overview of what they are capable of and when to use them.

11.2 Neural Networks

Historically, neural networks attempted to solve problems in a way modeled on how the brain operates. Today they are generally viewed as powerful modeling techniques.

A typical neural network consists of several neurons arranged in layers to create a network. Each neuron can be thought of as a processing element that is given a simple part of a task. The connections between the neurons provide the network with the ability to learn patterns and interrelationships in data. The figure below gives a simple representation of a neural network (a multi-layer perceptron).

Figure 11.1 Simple Representation of a Common Neural Network



When using neural networks to perform predictive modeling, the input layer contains all of the fields used to predict the outcome. The output layer contains the target of the prediction. The input fields and target field can be continuous or categorical (in PASW Modeler, categorical fields are transformed into a numeric form (dummy or binary set encoding) before processing by the network). The hidden layer contains a number of neurons at which outputs from the previous layer combine. A network can have any number of hidden layers, although these are usually kept to a minimum. All neurons in one layer of the network are connected to all neurons within the next layer.

While the neural network is learning the relationships between the data and results, it is said to be training. Once fully trained, the network can be given new, unseen data and can make a decision or prediction based upon its experience.

When trying to understand how a neural network learns, think of how a parent teaches a child how to read. Patterns of letters are presented to the child and the child makes an attempt at the word. If the child is correct she is rewarded and the next time she sees the same combination of letters she is likely to remember the correct response. However, if she is incorrect, then she is told the correct response and tries to adjust her response based on this feedback. Neural networks work in the same way.

PASW Modeler provides two different classes of supervised neural networks, the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF). In this course we will concentrate on the MLP type network and the reader is referred to the *PASW Modeler Node Reference* and the *Advanced Modeling with PASW Modeler* training course for more details on the RBF approach to neural networks.

Within a MLP, each hidden layer neuron receives an input based on a weighted combination of the outputs of the neurons in the previous layer. The neurons within the final hidden layer are, in turn, combined to produce an output. This predicted value is then compared to the correct output and the difference between the two values (the error) is fed back into the network, which in turn is updated. This feeding of the error back through the network is referred to as back-propagation.

To illustrate this process we will take the simple example of a child learning the difference between an apple and a pear. The child may decide in making a decision that the most useful factors are the shape, the color and the size of the fruit—these are the inputs. When shown the first example of a fruit she may look at the fruit and decide that it is round, red in color and of a particular size. Not knowing what an apple or a pear actually looks like, the child may decide to place equal importance on each of these factors—the importance is what a network refers to as weights. At this stage the child is most likely to randomly choose either an apple or a pear for her prediction.

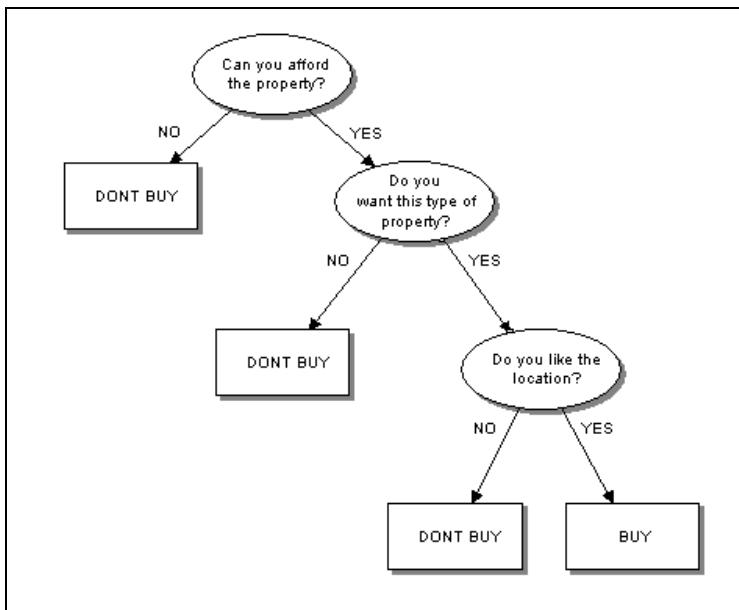
On being told the correct response, the child will increase or decrease the relative importance of each of the factors to improve their decision (reduce the error). In a similar fashion a MLP begins with random weights placed on each of the inputs. On being told the correct response, the network adjusts these internal weights. In time, the child and the network will hopefully make correct predictions.

11.3 Rule Induction

A common complaint with neural networks is that they are “black box” techniques; that is, it is very difficult to work out the reasoning behind their predictions. Rule Induction is a complementary technique in the sense that it does not suffer this problem.

PASW Modeler contains five different rule induction algorithms: C5.0, CHAID, QUEST, and C&R Tree (classification and regression tree) and Decision List. All of them derive a decision tree or a set of rules that try to describe distinct segments within the data in relation to a target field. The model’s output openly shows the reasoning for each rule and can therefore be used to understand the decision-making process that drives a particular outcome. Some differences between the algorithms will be given in Lesson 9.

To help understand rule induction, let us think about making a decision to buy a house. The most important factor may be cost—can you afford the property? The second may be what type of property are you looking for—a house or a condo? The next consideration may be the location of the property, etc.

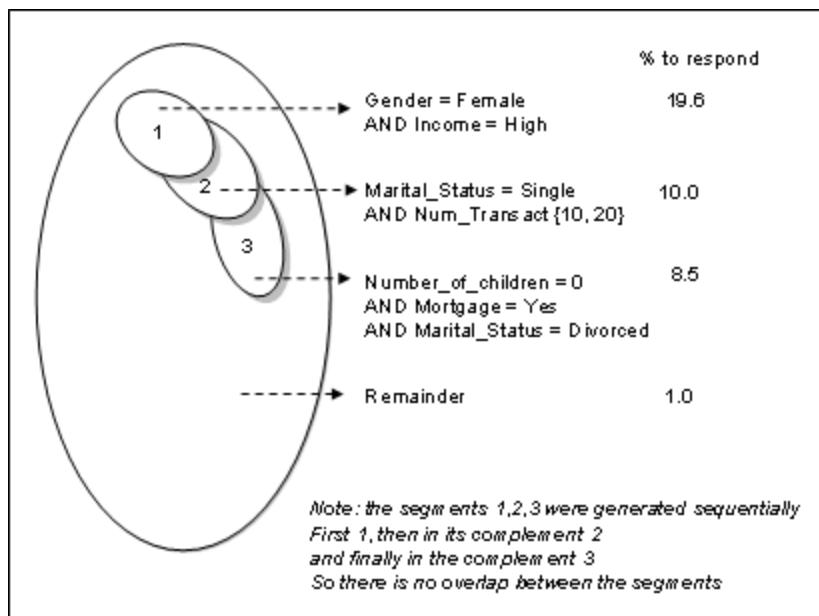
Figure 11.2 Graphical Representation of a Decision Tree

Another advantage of rule induction methods over neural networks is that the process automatically eliminates any fields that are not important in making decisions, while most neural networks include all inputs. This provides you with useful information and can even be used to reduce the number of fields entering a neural net.

The C5.0 rule induction method in PASW Modeler allows you to view the rules in two different formats: the decision tree presentation, which is useful if the user wants to visualize how the predictor fields split the data into subsets, and the rule set presentation which breaks the tree into collections of "IF – THEN" rules, organized by outcome. The latter is useful if we wish to see how particular groups of input values relate to one value of the outcome.

The Decision List algorithm in PASW Modeler differs from the other rule induction algorithms. It does not build a tree but only a list of decision rules. It is used to identify subgroups or segments that show a higher, or lower, likelihood of being in a specific category relative to the overall sample. Decision List predicts flag fields, but it can use a categorical field with more than two categories by grouping the others. The rules typically describe a part of the customer base, with the rest regarded as a Remainder group. The rules are presented as a list—hence *Decision List*.

Decision List could be used, for example, in a database marketing campaign where an offer will not be sent to all customers but only to the best responding, given the limitations of a particular campaign budget.

Figure 11.3 Graphical Representation of a Decision List

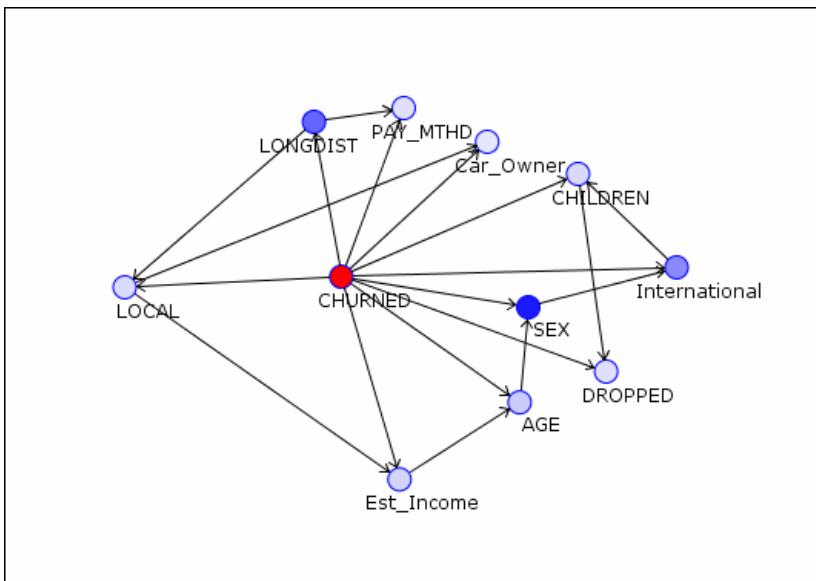
The rule induction algorithms are discussed, and C5.0 and CHAID are demonstrated, in Lesson 12.

11.4 Bayes Networks

A Bayesian network is a graphical model that displays fields (often referred to as nodes) in a dataset and the probabilistic, or conditional, independencies between them. Bayes Networks as implemented in Statistics are used to predict a categorical field.

Causal relationships between nodes may be represented by a Bayesian network; however, the links in the network (also known as arcs) do not necessarily represent direct cause and effect. For example, a Bayesian network can be used to calculate the probability of a patient having a specific disease, given the presence or absence of certain symptoms and other relevant data, if the probabilistic independencies between symptoms and disease as displayed on the graph hold true. Networks are very robust where information is missing and make the best possible prediction using whatever information is present.

The network is based on Bayesian probability theory, which uses prior distributions of each field and joint distributions to calculate a posterior distribution for fields of interest, especially the target.

Figure 11.4 Bayesian Network to Predict Customer Churn

Two different methods to develop a network are provided in PASW Modeler:

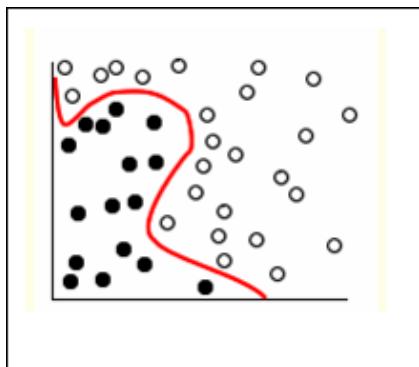
- Tree Augmented Naïve Bayes, which allows each predictor to depend on one other predictor, i.e., it can model interactions. It creates a relatively simple model that can be calculated quickly.
- Markov Blanket, which identifies all the fields in a network that are needed to predict the target. This can produce more complex networks and require more training time.

Field selection can be done to reduce the set of fields, and Bayes networks work best with a smaller set of predictors.

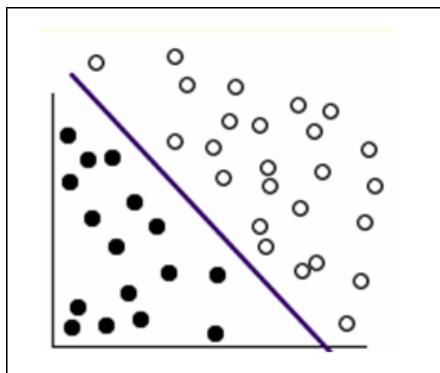
11.5 Support Vector Machines

A Support Vector Machine is a predictive model that attempts to classify outcomes by mapping data to a higher-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, and then the data are transformed in such a way that the separator can be drawn as a hyperplane. An SVM attempts to create the maximum separation between categories by appropriate placement of the hyperplane.

To illustrate, in Figure 11.5 we see a situation where the black and open circles—the target field—can be easily separated in two dimensions, but only with a complex curve, not a simple straight line or variant thereof.

Figure 11.5 Predicting a Binary Target

In Figure 11.6, the data have been mapped or transformed to another representation such that the target categories can be well separated here with a simpler function, i.e., a straight line (the higher dimension space can't be illustrated easily in two dimensions here).

Figure 11.6 SVM Mapping of Data to a Higher Dimension

The mathematical function used for the transformation is known as a kernel function. SVM in PASW Modeler supports linear, polynomial, radial basis, and sigmoid kernel functions.

Although the illustrations above show a categorical outcome, SVMs can also be used to predict a continuous target. SVM models are particularly suited for use with wide datasets—those with a large number of predictor fields.

11.6 Self-Learning Response Model

The Self-Learning Response Model (SLRM) node enables you to build a model that you can continually update, or reestimate, as a dataset grows without having to rebuild the model every time using the complete dataset. For example, this is useful when you have several products and you want to identify which product a customer is most likely to buy if you offer it to them. This model allows you to predict which offers are most appropriate for customers and the probability of the offers being accepted.

The SLRM model is based on creating a Bayesian network with a Naïve Bayes model. This type of network is based on a conditional independence model of each predictor given the target class, meaning that the predictors are not interrelated.

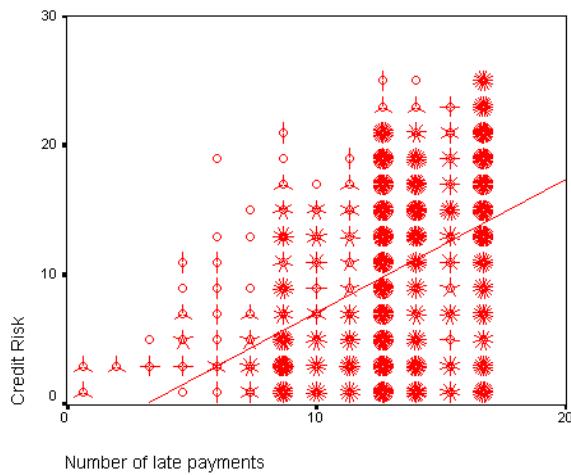
The model can initially be built using a small dataset with randomly made offers and the responses to those offers. As the dataset grows, the model can be updated and therefore becomes more able to predict the most suitable offers for customers and the probability of their acceptance based upon other input fields such as age, gender, job, and income. The offers available can be changed by adding or removing them from within the node dialog box, instead of having to change the target field of the dataset.

11.7 Linear Regression

Here we begin with a type of data mining model called statistical prediction models. Linear regression, logistic regression, discriminant analysis and generalized linear models—statistical modeling procedures available within PASW Modeler—make stronger data assumptions (linear model, normality of errors for regression, linear model in log-odds form, binomial or multinomial distribution of the target field, multivariate normality for discriminant, and so forth) than do machine learning techniques. Models can be expressed using simple equations, aiding interpretation, and statistical tests can guide field selection in the model. In PASW Modeler, the first three procedures have stepwise options that can automate input field selection when building models. They are not as capable, at least in standard form, as neural networks in capturing complex interactions among inputs and nonlinear relations.

Linear regression is a method familiar to just about everyone these days. It is the classic general linear model technique and is used to predict a continuous field with a set of predictors that are also continuous. However, categorical input fields can be included by using dummy-coded forms of these fields. Linear regression assumes that the data can be modeled with a linear relationship. The figure below presents a scatter plot depicting the relationship between the number of previous late payments for bills and the credit risk of defaulting on a new loan. Superimposed on the plot is the best-fit regression line.

Figure 11.7 Linear Regression Line Superimposed on Plot



Although there is a lot of variation around the regression line, it is clear that there is a trend in the data such that more late payments are associated with a greater credit risk.

The Regression node runs relatively quickly (single pass through the data). It is supported by statistical tests and goodness-of-fit measures. Since the final model is in the form of a single linear equation, model interpretation is straightforward.

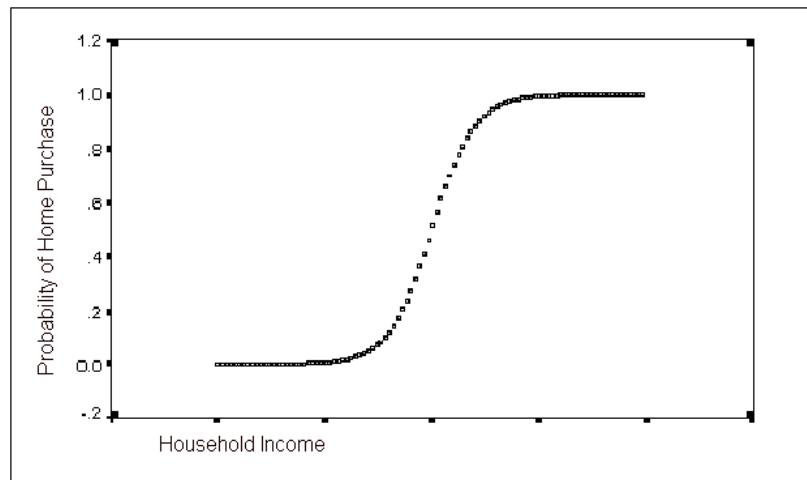
New in Modeler 14 is the Linear node, which allows categorical fields as inputs and displays the complete solution with all predictors in a convenient graphical form.

11.8 Logistic Regression

Logistic or multinomial regression attempts to predict a categorical target field. It is similar to linear regression in that it uses the general linear model as its theoretical underpinning, and so calculates regression coefficients and tries to fit cases to a line, although not a straight one. A common application would be predicting whether or not someone renews an insurance policy.

Logistic regression actually predicts a continuous function that represents the probability associated with being in a particular outcome category. This is shown in the figure below, which presents the two-category outcome case. It displays the predicted relationship between household income and the probability of purchase of a home. The S-shaped curve is the logistic curve, hence the name for this technique. The idea is that at low income, the probability of purchasing a home is small and rises only slightly with increasing income. But at a certain point, the chance of buying a home begins to increase in almost a linear fashion, until eventually most people with substantial incomes have bought homes, at which point the function levels off again. Thus the outcome ranges from 0 to 1 because it is measured in probability.

Figure 11.8 Logistic Function



After the procedure calculates the predicted outcome probability, it simply assigns a record to a predicted outcome category based on whether its probability is above .50 or not. An extension of this approach is used when the target field has three or more categories.

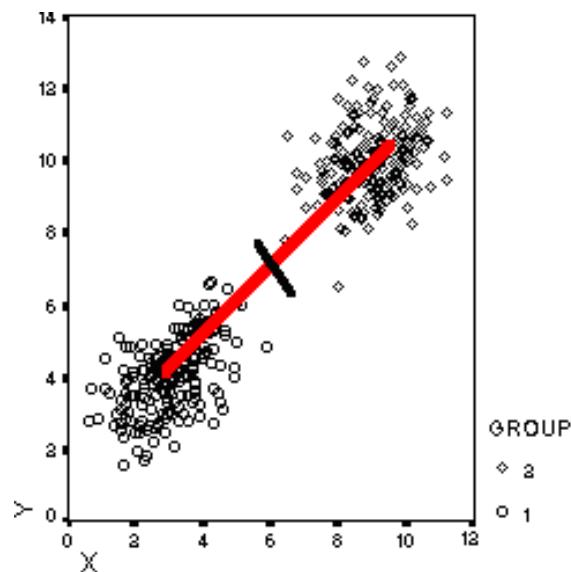
As with linear regression, logistic regression produces regression coefficients and associated statistics. The logistic regression coefficients can be related to the predicted odds of the target outcome category. This type of information is very powerful for decision-making. Although interaction effects (effects involving combinations of input fields) can be explicitly built into logistic regression models, they are not usually included, so logistic regression procedures, like linear regression procedures, are less likely to fit complex datasets than neural network or rule induction techniques.

11.9 Discriminant Analysis

This statistical technique attempts to predict a categorical target. In that sense, it is akin to logistic regression, but the underlying statistical model is very different. Discriminant Analysis formally makes stronger assumptions about the predictor fields, specifically that for each category of the target field they follow a multivariate normal distribution with identical population covariance matrices. Based on this you would expect discriminant to be rarely used since this assumption is seldom met in practice. However, Monte Carlo simulation studies indicate that multivariate normality is not critical for discriminant to be effective.

Discriminant Analysis is based on the assumption that the domain of interest is composed of separate *populations*. It attempts to find linear combinations of the predictors that best separate the populations. These linear combinations are a function, and one or more functions may be needed for a particular analysis. A discriminant analysis with two predictors and one function is represented in the figure below.

Figure 11.9 Two Distinct Samples with the Discriminant Axis



The two populations are best separated along an axis (a discriminant function) that is a linear combination of fields X and Y. The midpoint between the two populations is the cut-point. This function and cut-point would be used to classify cases, i.e., make predictions about category membership.

Discriminant Analysis produces coefficients that can be used to rank the predictors in importance on the discriminant function(s), and other statistical output, including a classification table to directly review the accuracy of the model.

As with regression models, interaction effects can be manually included but are not usually added by default to a model. Thus discriminant is less likely to fit complex datasets than the standard data mining procedures.

11.10 Generalized Linear Models

Linear regression models assume that the model error has a normal distribution, and that the variance of the residuals is homogeneous across the range of predicted values. These can be severe restrictions with many types of data, including data often encountered in data mining.

Generalized linear models were developed to relax these assumptions and still provide accurate predictions of a continuous target. Specifically, the target field is linearly related to the factors (categorical fields) and covariates (continuous fields) in the model via a specified link function. Generalized linear models can actually fit standard linear regression models, but also logistic models for binary data, loglinear models for count data, complementary log-log models for interval-censored survival data, plus many other statistical models through its very general model formulation.

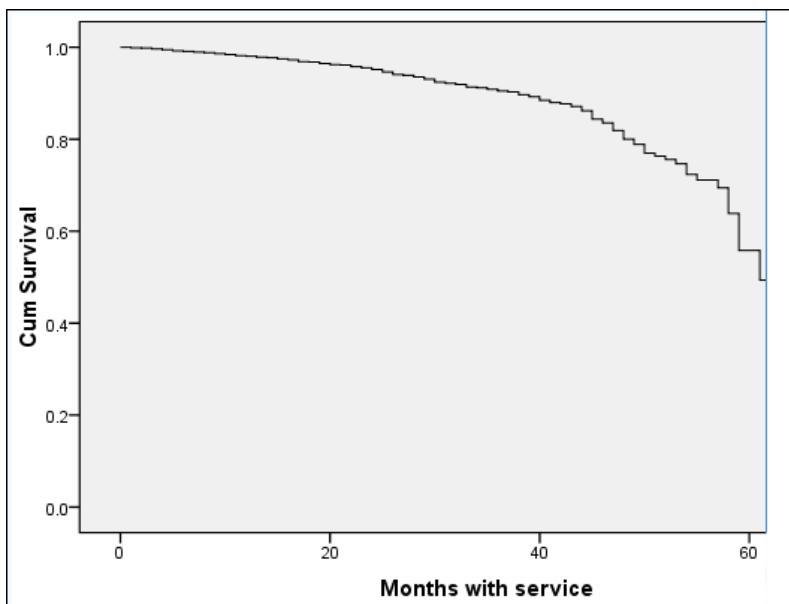
One of its most common uses in data mining could be for studies of the number of failures or defects in parts or machinery. This type of outcome usually follows a Poisson distribution and requires the special estimation techniques of generalized linear models.

A large amount of statistical output is available to help you determine the fit and suitability of a generated model. This technique is very powerful but does require knowledge of statistical theory and an in-depth knowledge of your data to be used successfully.

11.11 Cox Regression

Cox regression (or proportional hazards regression) develops a model for the analysis of the time until an event occurs. A set of predictors is used to predict the likelihood of the event of interest occurring at time t . Cox regression is based on survival analysis, which was developed to study such things as the time from diagnosis with a terminal illness until death. In the context of data-mining, Cox regression is often used to predict the time to churn, i.e., how long a customer is likely to remain a customer until they cancel their service, subscription, etc. with a company.

Figure 11.10 shows a general survival curve for all customers of a telecommunications firm. After 60 months, only about half of the original group of customers studied had not cancelled their service.

Figure 11.10 Customer Survival Time to Churn in Months

The target field should be a flag field, with string or integer storage, and with the event of interest coded as “true.” Some observations will likely be censored, which means that the event of interest has not yet occurred when data collection ceased. These observations will be used in the model up to the last time recorded for each.

The key statistical output of the model is the hazard rate (and ratio) at time t , which is the probability of the given event occurring in that time period, given survival until time t . Basically Cox Regression is a regression model with the “hazard” as the target field. A key assumption of the model is that the ratio of hazards between groups will remain constant over time (though the hazards themselves may increase or decrease).

It is also possible to perform Cox regression without any predictors; this is equivalent to performing a Kaplan-Meier analysis.

11.12 Automated Modeling

The automated modeling methods estimate and compare a number of different modeling methods, allowing you to try out a variety of approaches in a single modeling run. You can select the modeling algorithms to use, and the specific options for each, including combinations that would otherwise be mutually-exclusive. For example, rather than choose between the quick, dynamic, or prune methods for a Neural Net, you can try them all. The node explores every possible combination of options, ranks each candidate model based on the measure you specify, and saves the best for use in scoring or further analysis.

Auto Classifier

PASW Modeler offers several methods to predict a categorical target field (e.g., did a customer respond to an offer, or not). It is normally good practice to try several different types of models on the same problem, and then compare the results to find the best model, since there is no guarantee that a particular technique will be the top model on all types of data.

However, comparing these models can be a tedious process. Consequently, PASW Modeler provides the Auto Classifier node to automate the process of model comparison for categorical target fields.

Using a number of different methods (Neural Net, all the decision trees, Decision List, and Logistic Regression, etc.), you can try out a variety of approaches and compare the results. You can select the specific modeling algorithms that you want to use and the exact options for each. You can also specify multiple variants for each model, such as trying the quick, dynamic, and prune methods simultaneously for a Neural Net. The node generates a set of models based on the specified options and ranks the candidates based on the criteria you specify.

When an Auto Classifier node is run, the node estimates candidate models for every possible combination of options, ranks each candidate model based on the measure you specify, and saves the best models in a composite automated model nugget. This model nugget actually contains a set of one or more models generated by the node, which can be individually browsed or selected for use in scoring. The model type and build time are listed for each model, along with a number of other measures as appropriate for the type of model. You can sort the table on any of these columns to quickly identify the most interesting models.

Auto Numeric

Analogous to the Auto Classifier, PASW Modeler includes a node to automate the production of models for continuous targets fields. Model types included are Neural Net, C&R Tree, CHAID, Regression, Generalized Linear Models, Support Vector Machines (SVM) and Nearest Neighbor(KNN). Options can be selected for specific modeling algorithms to try multiple variants. Models are ranked on various criteria, including the correlation of the predicted and actual values of the target, or the relative error in predicting the target.

As with the Auto Classifier, the Auto Numeric node generates output comparing the models you selected. You can then automatically generate model nodes you wish to explore further.

Auto Cluster

The Auto Cluster node estimates and compares clustering models, which identify groups of records that have similar characteristics. The node works in the same manner as other automated modeling nodes, allowing you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields.

The Auto Cluster works in the same manner as the Auto Classifier and Auto Numeric nodes, but supports clustering models including TwoStep, K-Means, and Kohonen.

11.13 Clustering

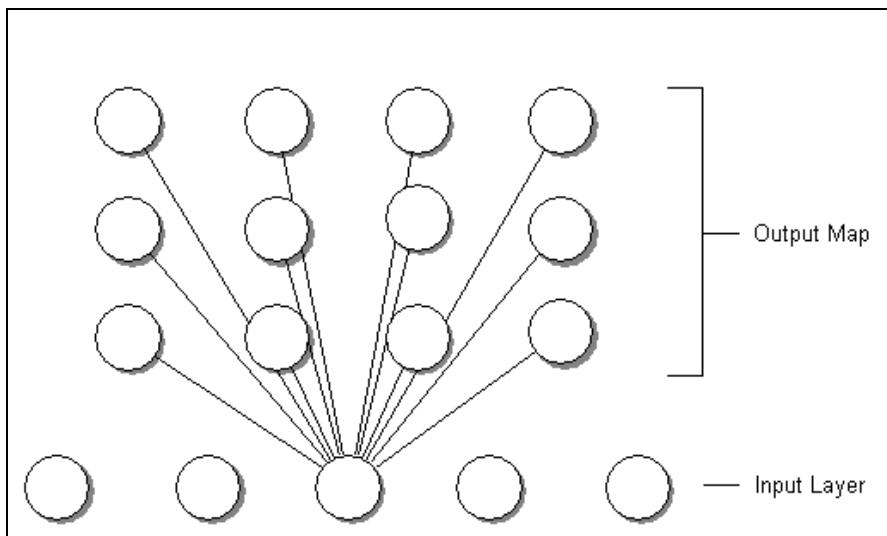
Clustering methods help discover groups of data records with similar values or patterns. These techniques are used in marketing (customer segmentation) and other business applications (records that fall into single-record clusters may contain errors or be instances of fraud). Clustering is sometimes performed prior to predictive modeling. In such instances, the customer groups might be modeled individually (taking the approach that each cluster is unique) or the cluster group might be an additional input to the model. PASW Modeler offers three clustering methods: K-means, Kohonen networks, and Two-step.

Kohonen Networks

A Kohonen network is a type of neural network that performs unsupervised learning; that is, it has no target field to predict. Such networks are used to cluster or segment data, based on the patterns of input fields. Kohonen networks make the basic assumption that clusters are formed from patterns that share similar features and will therefore group similar patterns together.

Kohonen networks are usually one- or two-dimensional grids or arrays of artificial neurons. Each neuron is connected to each of the inputs (input fields), and again weights are placed on each of these connections. The weights for a neuron represent a profile for that cluster on the fields used in the analysis. There is no actual output layer in Kohonen networks, although the Kohonen map containing the neurons can be thought of as an output. The figure below shows a simple representation of an output grid or Kohonen map.

Figure 11.11 Basic Representation of a Kohonen Network



Note that the connections from the input neuron layer are shown for only one neuron.

When a record is presented to the grid, its pattern of inputs is compared with those of the artificial neurons within the grid. The artificial neuron with the pattern most like that of the input “wins” the input. This causes the weights of the artificial neuron to change to make it appear even more like the input pattern. The Kohonen network also slightly adjusts the weights of those artificial neurons surrounding the one with the pattern that wins the input.

This has the effect of moving the most similar neuron and the surrounding nodes, to a lesser degree, to the position of the record in the input data space. The result, after the data have passed through the network a number of times, will be a map containing clusters of records corresponding to different types of patterns within the data. Kohonen networks will be examined in Lesson 11.

K-Means Clustering

K-means clustering is a relatively quick method for exploring clusters in data. The user sets the number of clusters (k) to be created, and the procedure selects k well-spaced data records as starting clusters. Each data record is then assigned to the nearest of the k clusters. The cluster centers (means on the fields used in the clustering) are updated to accommodate the new members. Additional data

passes are made as needed; as the cluster centers shift, a data record may need to be moved to its now nearest cluster.

Since the user must set the number of clusters, this procedure is typically run several times, assessing the results (mean profiles, number of records in each cluster, cluster separation) for different numbers of clusters (values of k).

Two-Step Clustering

Unlike the previous cluster methods discussed, two-step clustering will automatically select the number of clusters. The user specifies a range (Minimum (*Min*) and Maximum (*Max*) for the number of clusters. In the first step, all records are classified into pre-clusters. These pre-clusters are designed to be well separated. In the second step, a hierarchical agglomerative cluster method (meaning that once records are joined together to create clusters, they are never split apart) is used to successively combine the pre-clusters. This produces a set of cluster solutions containing from *Max* clusters down to *Min* clusters. A criterion (likelihood-based) is then used to decide which of these solutions is best.

The two-step clustering method thus has the advantage of automatically selecting the number of clusters (within the range specified) and does not require enormous machine resources (since only the *Max* clusters, not all records, are used in the second step).

11.14 Association Rules

Association rule methods search for things (events, purchases, attributes) that typically occur together in the data. Association rule algorithms automatically find the patterns in data that you could manually find using visualization techniques such as the web node, but can do so much quicker and can explore more complex patterns.

The rules found associate a particular outcome category (called a conclusion) with a set of conditions. The target fields may vary from rule to rule and as a result the user does not often focus on one particular target field. In fact, the advantage of these algorithms over rule induction is that associations can exist between any of the fields. One disadvantage to rule associations is that they attempt to find patterns in what is potentially a very large search space, and can be slow in running. A second disadvantage is that often many rules are found and the user must manually examine the rules for ones of interest.

The three algorithms provided by PASW Modeler to generate association rules are called Apriori, Carma and Sequence.

The algorithms begin by generating a set of extremely simple rules. These rules are then specialized by adding more refined conditions to them (making the rules more complex) and the most interesting rules are stored.

PASW Modeler allows certain restrictions on the algorithms to help speed up the process such as limiting the number of possible conditions within a rule. The result is a set of rules that can be viewed but cannot be used directly for predicting.

11.15 Sequence Detection

Sequence detection methods search for sequential patterns in time-structured data. Their focus on time-ordered sequences, rather than general association, is what distinguishes them from the association rule methods discussed earlier. In such analyses there may be interest in identifying

common sequence patterns or in finding sequential patterns that often lead to a particular conclusion (for example, a purchase on a web-site, or a failure of a piece of equipment).

The Sequence node in PASW Modeler does sequence analysis and uses the CARMA algorithm, which makes only two passes through the data. It can also generate nodes that make predictions based on specific sequences.

Application areas of sequence detection include retail shopping, web log analysis, and process improvement (for example, finding common sequences in the steps taken to resolve problems with electronic devices).

Sequence detection is applied to categorical fields and if continuous fields are input, their values will be treated as categories. That is, a field that takes the values from 1 to 100 would be treated as having one hundred categories.

11.16 *Principal Components*

Principal components analysis and factor analysis are data reduction techniques that can be used prior to predictive modeling and, less so, to clustering. Principal components can replace sets of highly correlated continuous fields with a smaller number of uncorrelated fields that are linear combinations of the original fields. It is more likely to accompany statistical than machine learning methods, and is often used in analyses involving survey data with many rating scale fields. For more information, see the PASW Modeler user guides, Help file, or the modeling training courses.

11.17 *Time Series Analysis*

A time series is a field whose values represent equally spaced observations of a phenomenon over time. Examples of time series include quarterly interest rates, monthly unemployment rates, weekly beer sales, annual sales of cigarettes, and so on.

Time series analysis is usually based on aggregated data. There is normally no business need requiring sales forecasts on a minute-by-minute basis, while there is often great interest in predicting sales on a weekly, monthly, or quarterly basis. For this reason, individual transactions and events are typically aggregated at equally spaced time points (days, weeks, months, etc.), and forecasting is based on these summaries..

Classic time series involves forecasting future values of a time series based on patterns and trends found in the history of that series (exponential smoothing and simple ARIMA) or on predictor fields measured over time (multivariate ARIMA, or transfer functions). PASW Modeler provides all these techniques.

The Time Series node is different from other PASW Modeler nodes in that you cannot simply insert it into a stream and run the stream. The Time Series node must always be preceded by a Time Intervals node that specifies such information as the time interval to use (years, quarters, months etc.), the data to use for estimation, and how far into the future to extend a forecast, if used.

The Time Series node provides an Expert PASW Modeler that will find and fit an appropriate model to a set of time series data. Or, if you have the necessary knowledge, you can specify model parameters yourself to develop a model and make forecasts.

11.18 Which Technique, When?

Apart from the basics, this is a very difficult question to answer. Obviously if you have a field in the data you want to predict, then any of the supervised learning techniques or one of the statistical modeling methods (depending on the target field's type) will perform the task, with varying degrees of success. If you want to find groups of individuals that behave similarly on a number of fields in the data, then any of the clustering methods is appropriate. Association rules are not going to directly give you the ability to predict, but are extremely useful as a tool for understanding the various patterns within the data. If there is interest in sequential patterns in data, then sequence detection methods are the techniques of choice and some of them can be used to generate predictions.

But if you want to go further and decide which particular prediction technique will work better, then unfortunately the answer is that it depends on the particular data you are working on. In fact, more accurately, it depends on the particular fields you want to predict and how they are related to the various inputs. There are suggested guidelines as to when one technique may work better than another, and we will mention these in the following lessons, but these are only suggestions and not rules. They will be broken on many occasions!

The advantage of PASW Modeler is the simplicity of building the models. Neural networks, rule induction (decision trees) and regression models can be built with great ease and speed, and their results compared. You must remember that data mining is an iterative process: models will be built, broken down, and often even combined before the user is satisfied with the results.

One final yet important point to keep in mind when building models is that PASW Modeler will only find rules or patterns in data if they exist. You cannot extract a model with high predictive accuracy if no associations between the input fields and target field exist.

Summary

In this lesson you have been introduced to a number of the machine learning and statistical modeling capabilities of PASW Modeler. You should now have an understanding of the different types of analyses you can perform and the different algorithms that can help you achieve your desired outcome. In the next lesson we will describe how to build a neural network within PASW Modeler.

Lesson 12: Rule Induction

Objectives

- Introduce the rule induction nodes, C5.0, CHAID, QUEST, C&R Tree, and Decision List
- Build a C5.0 rule model
- Browse and interpret the results
- Build a Rule Set to view the results in a different way
- Build a CHAID model for comparison

Data

Throughout this lesson we will continue using the credit risk data (*Risk.txt*) introduced in the previous lessons with the aim of building a model that predicts the credit risk field. Following recommended practice, we will use a Partition Node to divide the cases into two partitions (subsamples), one to build the model and the other to test the model.

12.1 Introduction

Rule induction or decision tree methods are capable of culling through a set of predictors by successively splitting a dataset into subgroups on the basis of the relationships between predictors and the target field. In this lesson we introduce the algorithms in PASW Modeler that build rule induction models. We will explain the differences between the algorithms and work through examples using C5.0 and CHAID. We will make few changes to the default settings, and the reader is referred to the *PASW Modeler Node Reference* manual for more details on alternative settings and expert options. These topics are also covered in the *Predictive Modeling with PASW Modeler* training course.

12.2 Rule Induction in PASW Modeler

PASW Modeler contains a number of different algorithms for performing rule induction: C5.0, CHAID, QUEST, and C&R Tree (classification and regression trees) and Decision List. They are similar in that they can all construct a decision tree by recursively splitting data into subgroups defined by the predictor fields as they relate to the target field. They differ in several ways that are important to users.

Type of Output: C5.0, QUEST, and Decision List only use categorical target fields (measurement level nominal, ordinal, though the order will be ignored), and Decision List only predicts a flag field (although it uses fields with more than two values by grouping the values into two categories). C&R Tree and CHAID support both categorical and continuous targets. Thus, all could be used to build a credit risk model in which the outcome is either *good risk* or *bad risk* (binary), but only CHAID and C&R Tree could be used to build a model to predict next year spending (in dollars) for recently acquired customers.

Type of Split: When the dataset is recursively split into subgroups (on a predictor), C&R Tree and QUEST support only binary (two group) splits, while CHAID, C5.0, and Decision List support splits with more than two subgroups.

Criterion for Selection of Predictor: The algorithms differ in the criterion used to drive the splitting. For C5.0 an information theory measure is used: the information gain ratio. When C&R Tree predicts a categorical field, a dispersion measure (the Gini coefficient by default) is used. CHAID uses a chi-square test; QUEST uses a chi-square test for categorical predictors and analysis of

variance for continuous inputs. Finally, Decision List uses the statistical confidence that a segment (split) has a higher response probability than that for the overall sample.

Handling of Missing Predictor Values: All algorithms allow missing values for the predictor fields, although they use different methods. C5.0 uses a fractioning method, which passes a fractional part of a record down each branch of the tree from a node whose split is based on a field for which the record is missing. C&R Tree and QUEST use substitute prediction fields, where needed, to advance a record with missing values through the tree during training. CHAID and Decision List make the missing values a separate category and allow them to be used in tree building.

Building Trees Interactively: Three of the algorithms—CHAID, QUEST, and C&R Tree—support the ability to build the tree interactively, level-by-level, including selecting specific fields for a split. Decision List provides the ability to create one's own rules.

Growing Large Trees and Pruning: As decision trees grow large and bushy, the percentage of cases that pass through any given path in the tree decreases. Such bushy trees: 1) may not generalize as well to data, and 2) may have rules that apply to tiny groups of data. Three of the algorithms—QUEST, C5.0, and C&R Tree—grow large trees and then prune them back, a method found to be effective. However, they differ in their pruning criteria. In addition, C5.0 contains options that favor accuracy (maximum accuracy on training sample) or generality (results that should better generalize to other data). Also, all the algorithms allow you to control the minimum subgroup size (their definitions differ slightly), which helps avoid branches with few data records.

Rulesets: All the algorithms can represent a model as a ruleset for a categorical target. Rulesets can be easier to interpret than complex decision trees. However, a decision tree produces a unique classification for each data record, while more than one rule from a ruleset may apply to a record, which adds complexity (but a prediction can still be made, by voting among the rules). The Decision List algorithm presents the rules as a list. In case a data record satisfies multiple rules, it will be assigned to the first rule on the list it satisfies.

For all these reasons, you should not expect the algorithms to produce identical results for the same data. You should expect that important predictors would be included in trees/rulesets built by any algorithm. For these reasons, you may often try more than one algorithm for a data-mining project, and we will do so in this lesson.

Those interested in more detail concerning the algorithms can find additional discussion in the *Predictive Modeling with PASW Modeler* training course. Also, you might consider *C4.5: Programs for Machine Learning* (Morgan Kauffman, 1993) by Ross Quinlan, which details the predecessor to C5.0 (see also Ross Quinlan's website at www.rulequest.com); *Classification and Regression Trees* (Wadsworth, 1984) by Breiman, Friedman, Olshen and Stone, who developed CART (Classification and Regression Tree) analysis; the article by Loh and Shih (1997, "Split Selection Methods for classification trees," *Statistica Sinica*, 7: 815-840) that details the QUEST method; and for a description of CHAID, "The CHAID Approach to Segmentation Modeling: CHI-squared Automatic Interaction Detection," lesson 4 in Richard Bagozzi, *Advanced Methods of Marketing Research* (Blackwell, 1994).

12.3 Rule Induction Using C5.0

We first use the C5.0 node to create a rule induction model. Once trained, a Generated C5.0 node labeled with the name of the predicted field will appear in the Models tab of the Outputs manager. This node represents the trained C5.0 decision tree and contains the rule induction model in either

decision tree or rule set format. Its properties can be browsed and new data can be passed through this node to generate predictions.

Before a data stream can be used by the C5.0 node—or any node in the Modeling palette—the measurement level of all fields used in the model must be defined (either in the source node or a Type node). Additionally, the fields must be fully instantiated, which means that the data must be passed through the Type node before modeling.

This is because all modeling nodes use the Type information to set up the models (you can indicate which fields are the predictors and which field is the target within a modeling node, but it is more efficient to do so in a Type node). As a reminder, the table below shows the available roles for a field.

Table 12.1 Role Settings

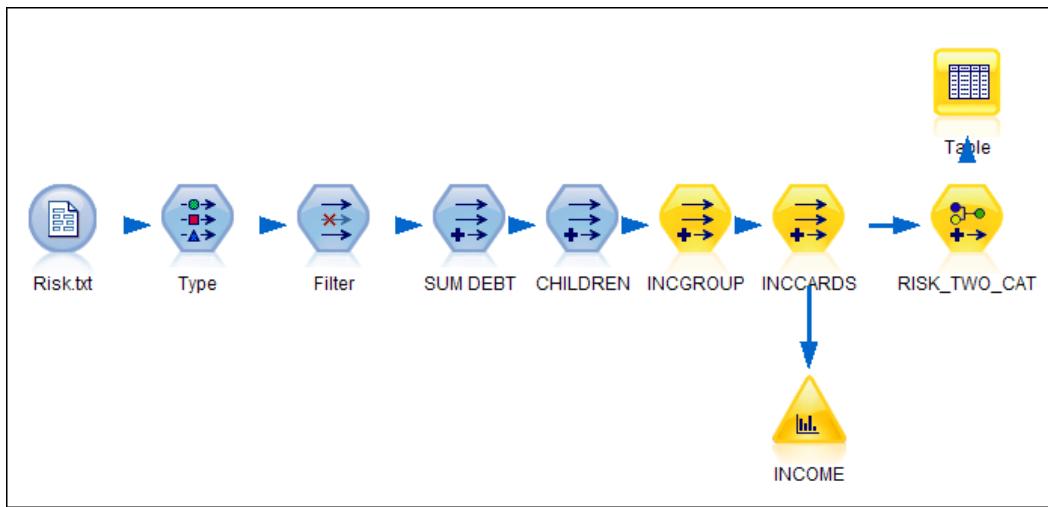
INPUT	The field acts as an input (predictor) within the modeling.
TARGET	The field is the target for the modeling.
BOTH	Allows the field to be act as both an input and a target in modeling. Role suitable for the association rule and sequence detection algorithms only, all other modeling techniques will ignore the field.
NONE	The field will not be used in machine learning or statistical modeling. Default if the field is defined as Typeless.
PARTITION	Indicates a field used to partition the data into separate samples for training, testing, and (optional) validation purposes.
SPLIT	Only available for categorical (flag, nominal, ordinal) fields. Specifies that a model is to be built for each possible value of the split field.
FREQUENCY	Only available for numeric fields. Setting this role enables the field value to be used as a frequency weighting factor for the record.
RECORD ID	Only relevant for the Linear node, where the specified field will be used as the unique record identifier.

Role can be set by clicking in the Role column for a field within the Type node or the Type tab of a source node and selecting the role from the drop-down menu. Alternatively, this can be done from the Fields tab of a modeling node.

Within the Type node or Types tab, the field to be predicted (or explained) must have role TARGET (or it must be specified in the Fields tab of the modeling node). All fields to be used as predictors must have their role set to INPUT. Any field not to be used in the modeling must have its role set to NONE. Any field with role BOTH will be ignored by C5.0 (or any decision tree modeling node). A field with role of PARTITION will be used to split the data, as we discussed in Lesson 7.

We want to use some of the fields that we created in Lesson 5 for this modeling exercise. So we'll open the stream saved there, then make some modifications and add appropriate nodes.

Click **File...Open Stream**, navigate to the **c:\Train\ModelerIntro** directory
 Double-click **Backup_Data Preparation.str**
 Delete all nodes from **INCGROUP** to the right (as shown in Figure 12.1); select all the nodes that are to delete (in golden) and then click **Delete** on the right-click menu

Figure 12.1 Data Preparation Stream with Nodes to Delete Outlined

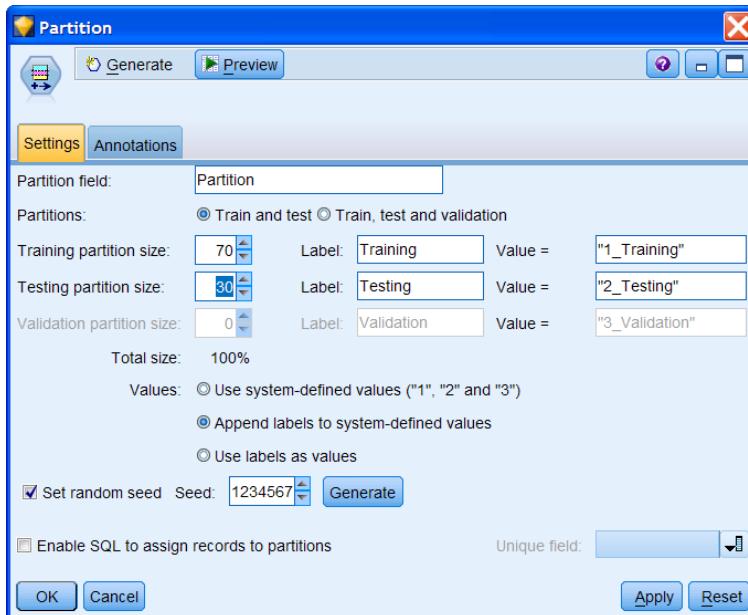
We will use the new fields *SUM DEBT* and *CHILDREN* for modeling, along with the original fields.

Place a **Partition** node from the Field Ops palette to the right of the **CHILDREN** node
 Connect the **CHILDREN** node to the **Partition** node

Place a **Type** node from the Field Ops palette to the right of the **Partition** node
 Connect the **Partition** Node to the **Type** node
 Double-click the **Partition** node to edit it.

As we did in Lesson 10, we will split the data 70/30 between the training and testing subsets. We will retain the default random number seed.

Change Training partition size to **70**
 Change Testing partition size to **30**

Figure 12.2 Partition Node with Partition Size Settings

Click **OK**

Next we will add a Table node to the stream. This not only will force PASW Modeler to autotype the data but also will act as a check to ensure that the data file is being correctly read.

Place a **Table** node from the Output palette above the **Type** node in the Stream Canvas

Connect the **Type** node to the **Table** node

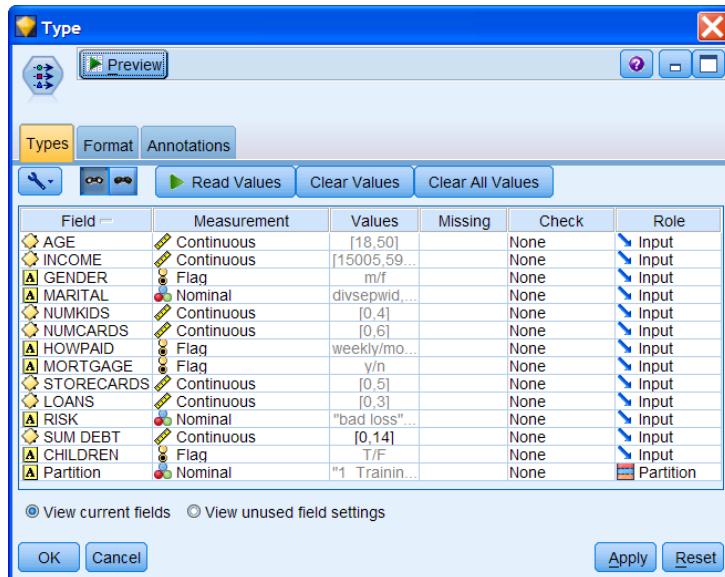
Run the **Table** node

The values in the data table look reasonable (not shown). Note that the field *Partition* has again been created.

Click **File...Close** to close the Table window

Double-click the **Type** node

Figure 12.3 Type Node with Fully Instantiated Data

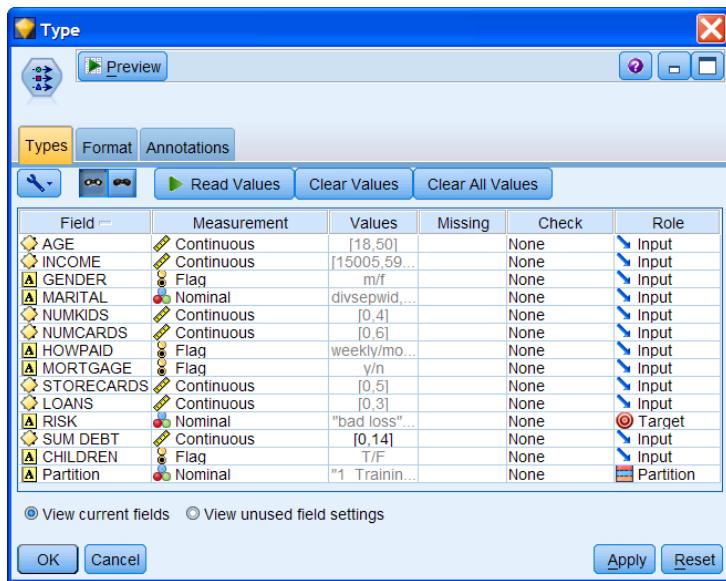


This is the second Type node in the stream. We added this one because we created new fields to use in modeling, and all data must be fully instantiated and typed before we can begin modeling.

Recall that there is a Filter node upstream of this Type node where we remove *ID*. Since you might wish to identify records by *ID* after predictions have been made, include that field we could have removed the filtering of *ID* before passing the data through the second Type node.

Click in the cell located in the **Role** column for **Risk** (current value is **Input**) and select **Target** from the list

The *Partition* field automatically has a role of PARTITION, so we are now ready for modeling. All other fields have role INPUT and so will be used as predictors.

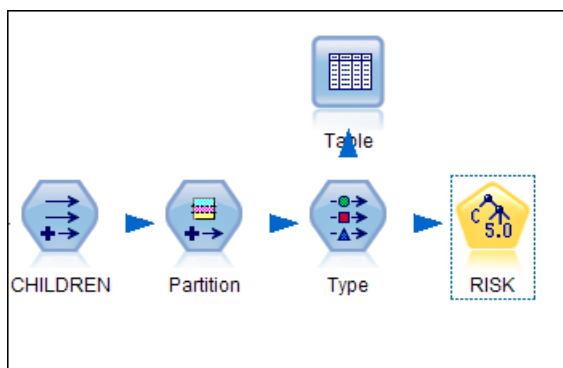
Figure 12.4 Type Node with Roles Set

Click **OK**

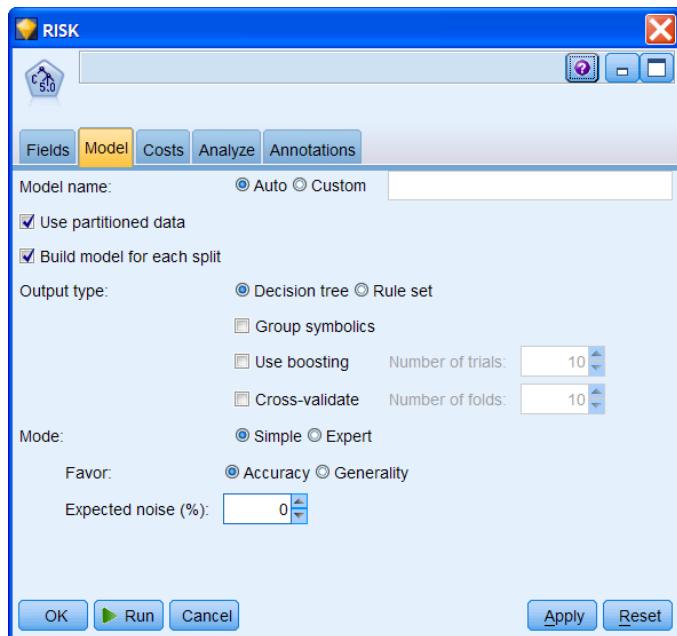
Place the **C5.0** node from the Modeling palette to the upper right of the **Type** node in the Stream Canvas

Connect the **Type** node to the **C5.0** node

The name of the C5.0 node should immediately change to RISK.

Figure 12.5 C5.0 Modeling Node Added to Stream

Double-click the **C5.0** node

Figure 12.6 C5.0 Model Dialog

The *Model name* option allows you to set the name for both the C5.0 and resulting C5.0 rule nodes. The form (decision tree or rule set, both will be discussed) of the resulting model is selected using the *Output type:* option.

The *Use partitioned data* option is checked by default so that the C5.0 node will make use of the Partition field created by the Partition node earlier in the stream. Whenever this option is checked, only the cases the Partition node assigned to the Training sample will be used to build the model; the rest of the cases will be held out for Testing and/or Validation purposes. If unchecked, the field will be ignored and the model will be trained on all the data.

By default, a model is built using all the training data partition. The *Cross-validate* option provides a way of validating the accuracy of C5.0 models when there are too few records in the data to permit a separate testing/holdout sample. It does this by splitting the data into N equal-sized subgroups and fitting N models. Each model uses N-1 of the subgroups for training, then applies the resulting model to the remaining subgroup and records the accuracy. Accuracy figures are pooled over the N holdout subgroups and this summary statistic estimates model accuracy applied to new data. Since N models are fit, N-fold validation is more resource intensive. It does not present the N decision trees or rule sets, just estimated accuracy. By default N, the number of folds, is set to 10.

For a predictor field that has been defined as categorical, C5.0 will normally form one branch per category. However, by checking the *Group symbolics* check box, the algorithm can be set so that it finds sensible groupings of the values within the field, thus reducing the number of rules. This is often desirable. For example, instead of having one rule per region of the country, group symbolic values may produce a rule such as:

Region [South, Midwest] ...
Region [Northeast, West] ...

Once trained, C5.0 builds one decision tree or rule set that can be used for predictions. However, it can also be instructed to build a number of alternative models for the same data by selecting the *Boosting* option. Under this option, when it makes a prediction it consults each of the alternative

models before making a decision. This can often provide more accurate prediction, but takes longer to train. Also the resulting model is a set of decision tree predictions and the outcome is determined by voting, which is not simple to interpret. But if accuracy is more important than model transparency, you may find boosting useful.

The C5.0 algorithm can be set to favor either *Accuracy* on the training data (the default) or *Generality* to other data. In our example, we favor a model that is expected to better generalize to other data and so we select *Generality*. This is more common since most data-mining projects are interested in making predictions for new data.

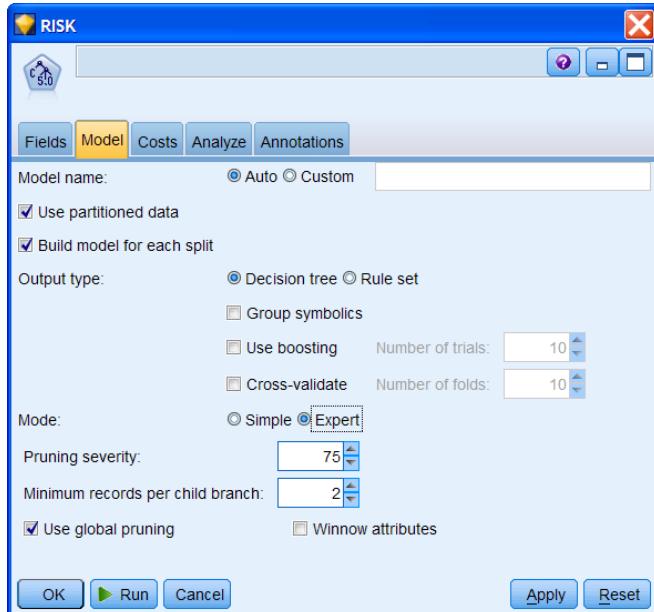
Click **Generality** option button

In the data mining, and, more broadly, the machine learning community, errors in data are often referred to as *noise*. We are concerned here with noise as errors in prediction. C5.0 will automatically handle noise within the data and, if known, you can inform PASW Modeler of the expected proportion of noisy or erroneous data. This option is rarely used.

As with all of the modeling nodes, after selecting the Expert option or tab, more advanced settings are available. In this course, we discuss the Expert options briefly. The reader is referred to the *PASW Modeler User's Guide* or the *Predictive Modeling with PASW Modeler* training course for more information on these settings.

Check the **Expert** option button

Figure 12.7 C5.0 Expert Options



By default, C5.0 will produce splits if at least two of the resulting branches have at least two data records each (*Minimum records per child branch*). For large datasets you may want to increase this value to reduce the likelihood of rules that apply to very few records. To do so, increase the value in the *Minimum records per child branch* box.

There are two types of *pruning* that C5.0 applies to a tree, both of which occur after a tree is fully grown. As with a tree in your garden, models are often best if pruned back to a state with fewer nodes. We recommend leaving the pruning settings at their default values, at least initially.

Click the **Simple** Mode option button, and then click **Run**

A C5.0 Rule node, labeled with the predicted field (*RISK*), is generated and will appear in the Models palette of the manager. Generated models are placed in the Models palette (located on the Models tab in the managers window in the upper right corner of the PASW Modeler main window), where they are represented by diamond-shaped icons (also called “nuggets”). From there, they can be selected and browsed to view details of the model.

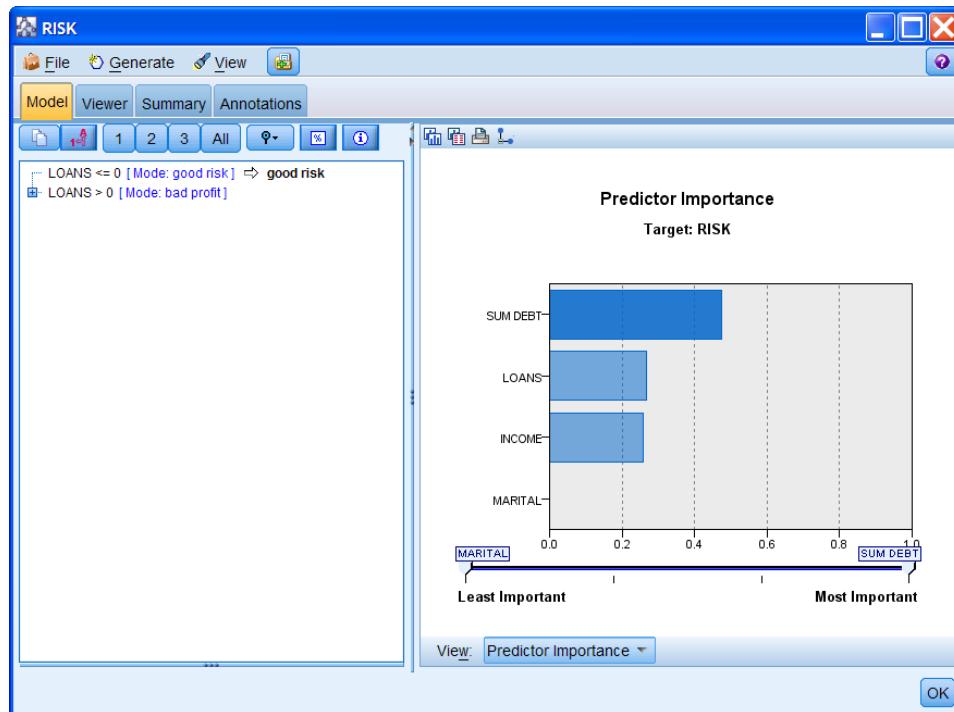
The completely gold icon (the nugget) means that it is a fully refined model, which can be placed into the stream to generate predictions or to allow further analysis of their properties. Notice, the nugget is not only added to the Models palette, but as well added to the stream canvas and connected to the C5.0 model node. If we would run the C5.0 model node with other settings, the nugget would reflect the updated generated model automatically.

12.4 Browsing the Model

Once the C5.0 Rule node is in the Models palette, the model can be browsed.

Right-click the **C5.0** node named **RISK** in the Models palette, then click **Browse**

Figure 12.8 Browsing the C5.0 Rule Node



Two panes are visible in the C5.0 model window. The pane on the left contains the model in the form of a decision tree. Initially, only the first branch of the tree is visible. The other pane contains a bar chart depicting variable importance, which is a measure that indicates the relative importance of each

field in estimating the model. Since the values are relative, the sum of the values for all fields on the display is 1.0.

According to what we see of the tree so far, *LOANS* is the first split in the tree. If *LOANS* ≤ 0 , the *Mode* value for *RISK* is *good risk* and if *LOANS* > 0 , the *Mode* value is *bad profit*. The *Mode* lists the modal (most frequent) category for the branch, and the mode will be the predicted value, unless there are other fields that need to be taken into account within that branch to make a prediction.

In this instance, the branch for *LOANS* ≤ 0 cannot be further refined since there is no \oplus symbol to the left of *LOANS* in this line. Also, the arrow indicates a terminal branch where a prediction is made. Thus the mode of *good risk* becomes the prediction for those customers without any outstanding loans. This makes perfect common sense, since individuals without debt from loans are, on the average, better credit risks.

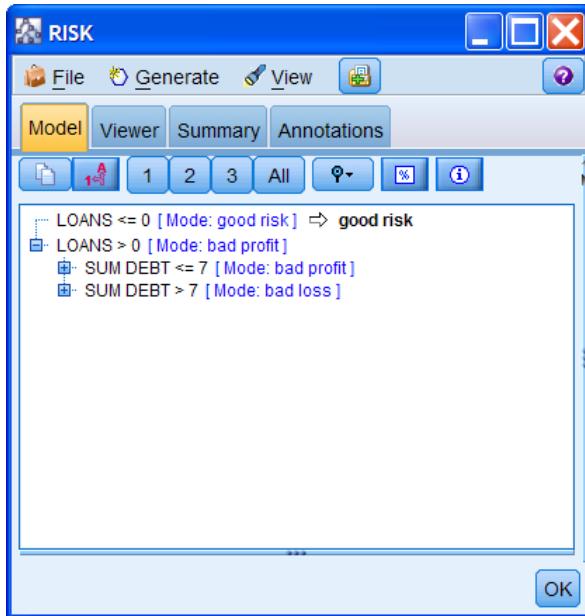
For those people with one or more loans, no prediction can yet be made because there are other fields that need to be taken into account.

To view the predictions we need to further unfold the tree. First close the right pane of the model window.

To unfold the branch *LOANS* > 0 , just click the expand button.

Click \oplus to **unfold** the branch **LOANS > 0**

Figure 12.9 Unfolding a Branch



SUM DEBT, one of our derived fields, is the next split, into two separate branches. Those with total summed debt ≤ 7 have a mode value of *bad profit*, while those with total summed debt greater than 7 have a mode value of *bad loss*.

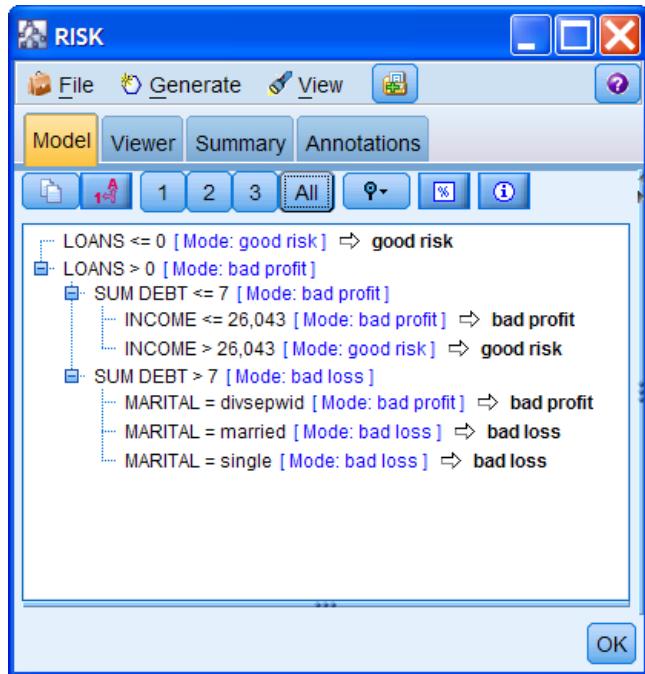
Again, this agrees with our practical experience, and it is always necessary to compare the predictions of a model with both previous models and knowledge and experience about the specific problem.

Although finding interesting, unexpected relationships is one of the delights and strengths of data mining, most patterns should make some intuitive sense.

However, we can't make any more predictions yet because these two branches can be unfolded further. We could unfold each separate branch to see the rest of the tree, but we will take a shortcut:

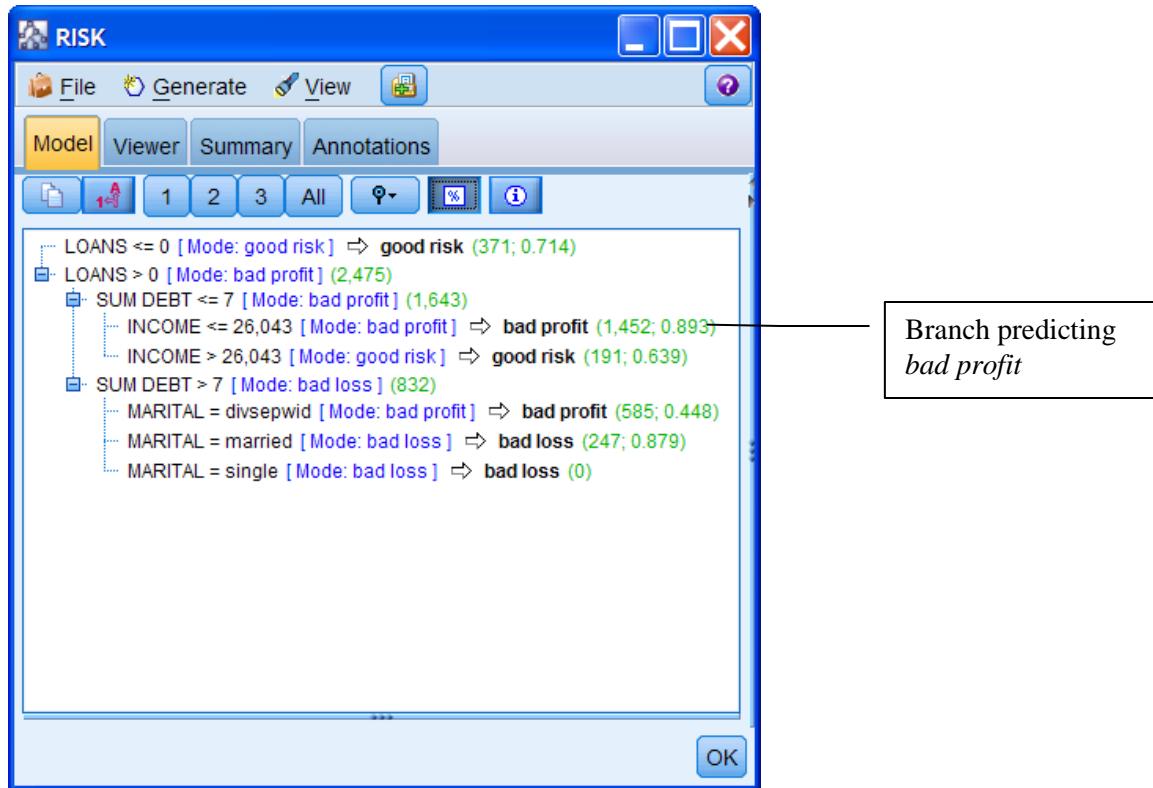
Click the **All** button in the Toolbar

Figure 12.10 Fully Unfolded Tree



Now the full tree is visible. We can see that there are 6 terminal nodes where a branch of the tree ends (and a prediction is made). If we are interested in the *bad profit* risk group, for example, the node with that prediction corresponds to the branch where *LOANS* > 0, *SUM DEBT* <=7, and *INCOME* <= 26,043. To get an idea about the number of records and the percentage of *bad profit* records within such branches we ask for more details.

Click **Show or hide instance and confidence figures** button in the toolbar

Figure 12.11 Instance and Confidence Figures Displayed

Record count and percentage accuracy figures are added to each terminal node in green. So for the branch described in the fourth line, we see that 1,452 individuals had 1 or more loans, had summed debts less than or equal to 7, and had incomes less than or equal to 26,043. That is a large proportion of the training file. The confidence or accuracy figure for this set of individuals is 0.893, which represents the proportion of records within this set correctly classified (predicted to be *bad profit* and actually being *bad profit*).

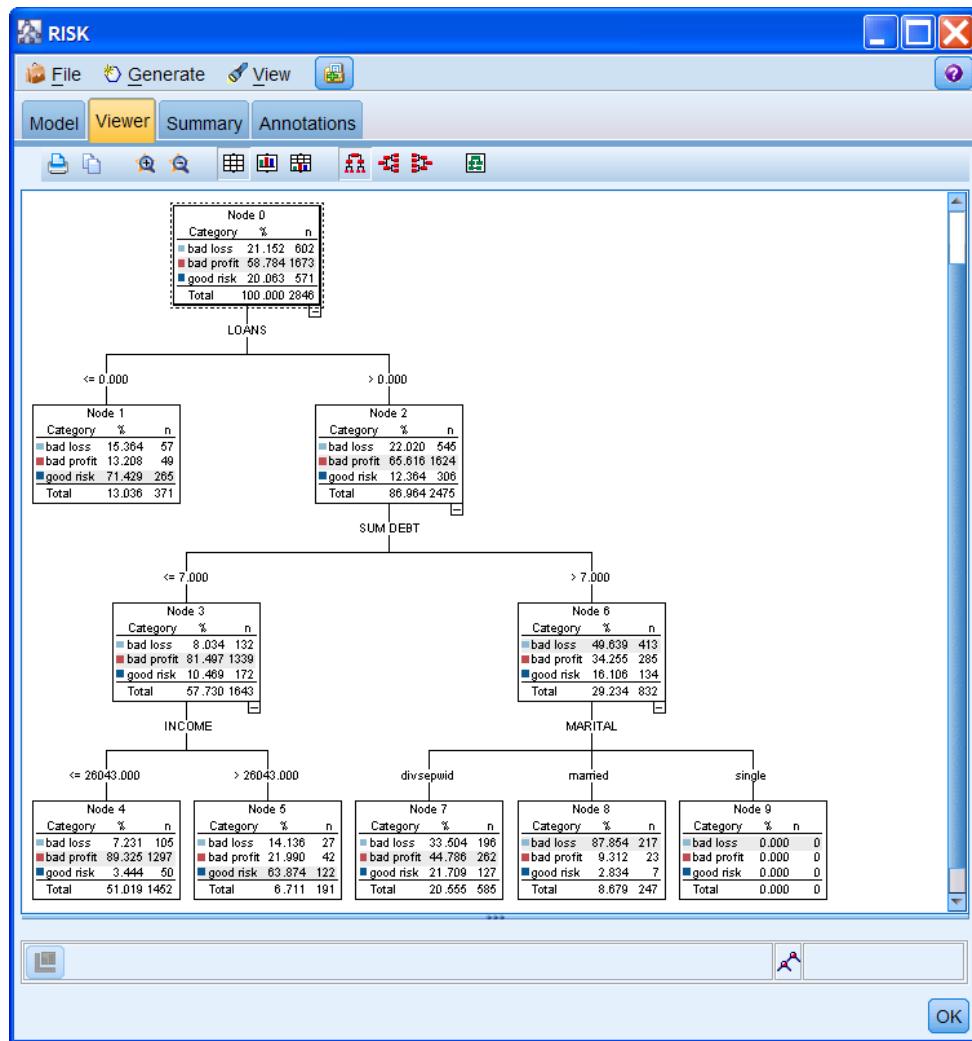
On the last line you will find a branch with 0 records. Apparently, no singles were present in the group with *SUM DEBT > 7* and *LOANS > 0*. On the other hand, PASW Modeler has the information from the Type tab that MARITAL has three groups. If we were to score another dataset with this model, how should we classify singles with at least one loan and with summed debts greater than 7? PASW Modeler assigns the group the modal category of the branch. So, as the mode in the *LOANS > 0* and *SUM DEBT > 7* group is *bad loss*, the singles inherit this as their prediction.

If you would like to present the results to others, an alternative format is available in the Viewer tab that helps visualize the decision tree.

Click the **Viewer** tab

Expand the Window, and click the **Decrease Zoom**  tool (to view more of the tree)

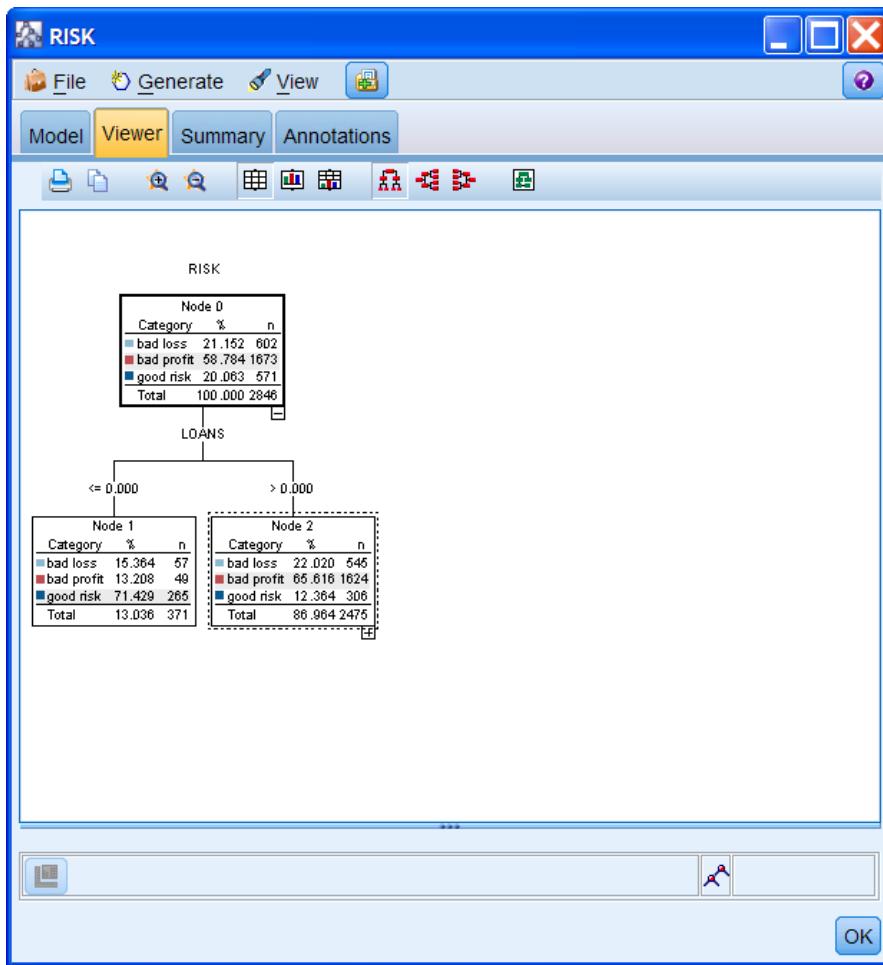
Figure 12.12 Decision Tree in the Viewer Tab



The root of the tree shows the overall percentages and counts for the three categories of risk. The modal category is shaded in each node.

The first split is on *LOANS*, as we have seen in the text display of the tree. Similar to the text display, we can decide to expand or collapse branches. In the right corner of some nodes a – or + is displayed, referring to an expanded or collapsed branch, respectively. For example, to collapse the tree at node 2:

Click in the lower right corner of **node 2**

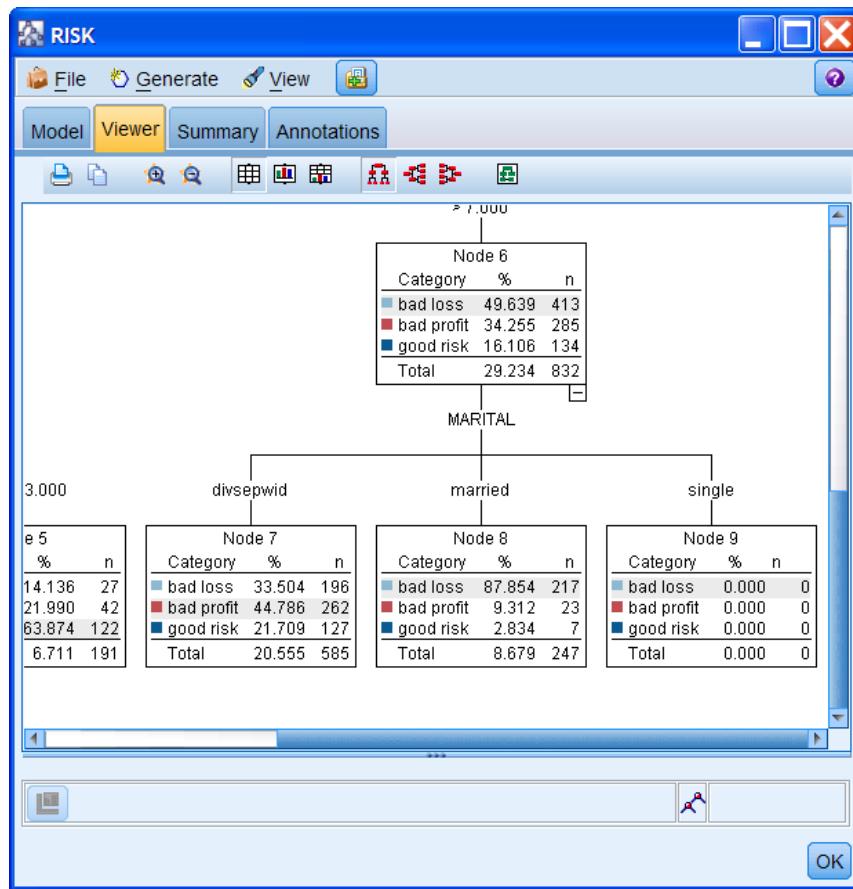
Figure 12.13 Collapsing a Branch

Now only the two nodes below the root are visible, as most of the tree branches out from Node 2.

It can be interesting to see the distribution of responses in a node, especially those where two values are about equally frequent.

Click in the lower right corner of **Node 2** to expand it
Zoom in on Node 7

We see that the modal category of *bad profit* for *RISK* occurred in 44.786% of the records (which is why the accuracy of prediction in this branch was listed as .448). But one-third of the customers have a value of *bad loss* for *RISK*. So if a misclassification occurs for this group (people with loans, with summed debts >7, and who are divorced, separated, or widowed), the true value may well be *bad loss*.

Figure 12.14 Detail for Node 7

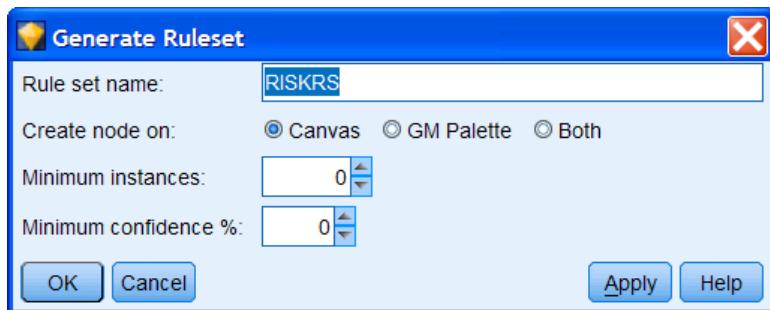
In the Viewer tab, toolbar buttons are available for showing frequency information as graphs and/or as tables, changing the orientation of the tree, and displaying an overall map of the tree in a smaller window (tree map window) that aids navigation in the Viewer tab.

12.5 Generating and Browsing a Rule Set

When building a C5.0 model, the C5.0 node can be instructed to generate the model as a rule set, as opposed to a decision tree. A rule set is a number of IF ... THEN rules which are collected together by outcome.

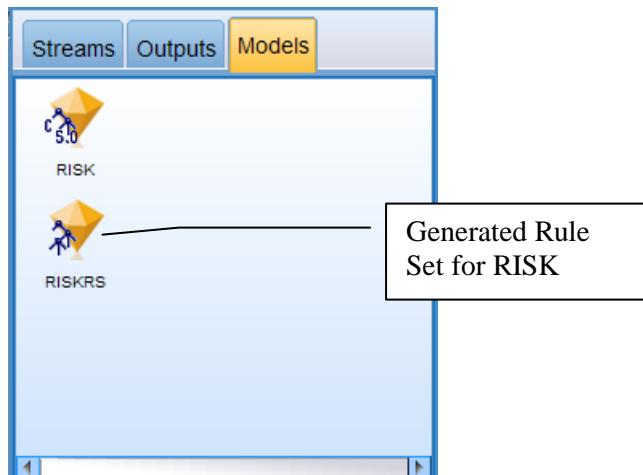
A rule set can also be produced from the Generate menu when browsing a C5.0 decision tree model.

In the **C5.0** Rule browser window, click **Generate...Rule Set**

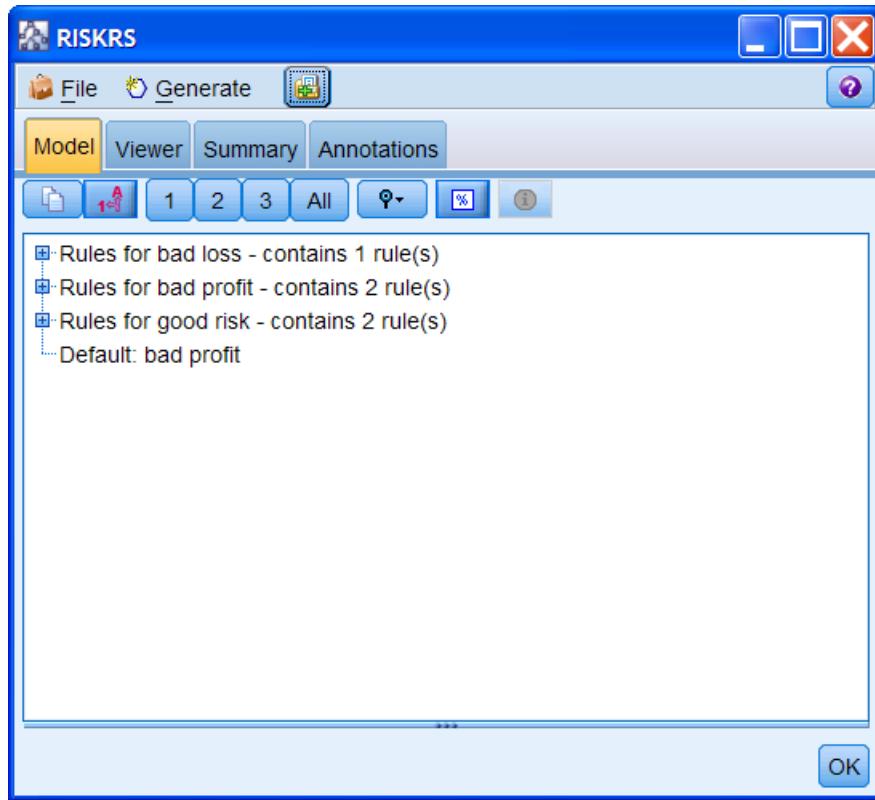
Figure 12.15 Generate Ruleset Dialog

Note that the default *Rule set name* appends the letters “RS” to the target field name. You may specify whether you want the C5.0 Ruleset node to appear in the Stream Canvas (*Canvas*), the generated Models palette (*GM palette*), or both. You may also change the name of the rule set and lower limits on support (percentage of records having the particular values on the input fields) and confidence (accuracy) of the produced rules (percentage of records having the particular value for the target field, given values for the input fields).

Click **GM Palette**
Click **OK**

Figure 12.16 Generated C5.0 Rule Set Node

Click **File...Close** to close the C5.0 Rule browser window
Right-click the C5.0 Rule Set node named **RISKRS** in the Models palette in the Manager, then click **Browse**

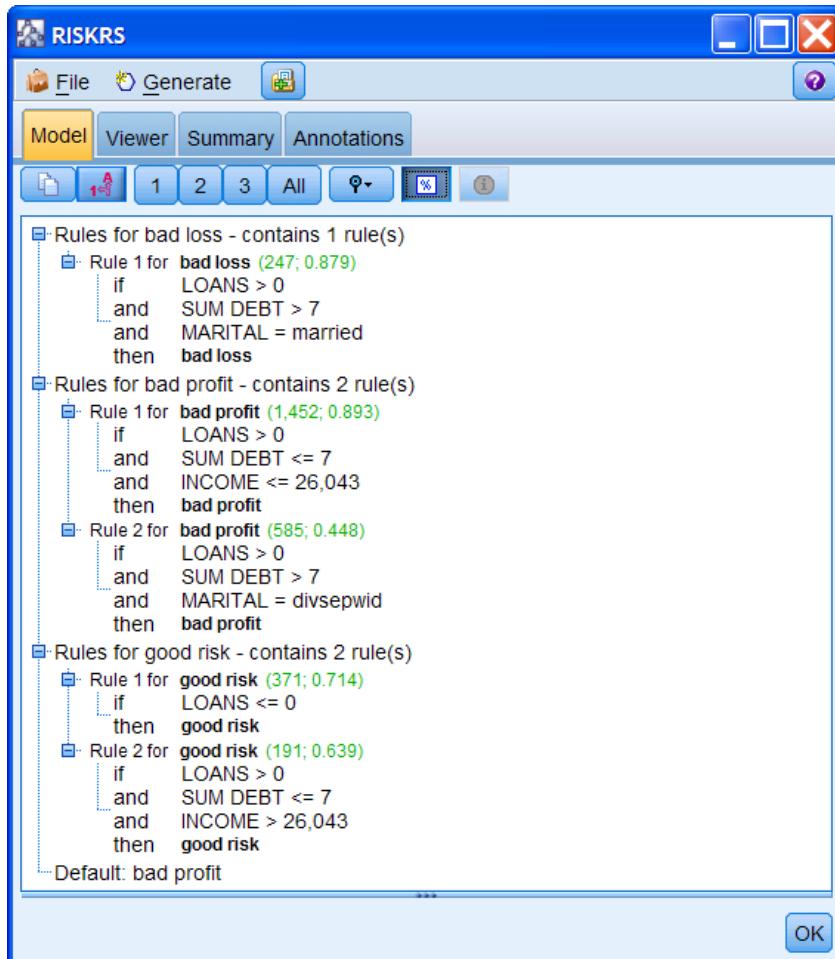
Figure 12.17 Browsing the C5.0 Generated Rule Set

Apart from some details, this window contains the same menus as the browser window for the C5.0 Rule node.

Click **All** button to unfold

Click **Show or hide instance and confidence figures** button in the toolbar

The numbered rules now expand as shown below.

Figure 12.18 Fully Expanded C5.0 Generated Rule Set

For example Rule 1 (*bad loss*) has this logic: If the person has one or more loans, is married, and has summed debts greater than 7, then predict *bad loss* (this is the only prediction for *bad loss* with data in the current file). This form of the rules allows you to focus on a particular conclusion rather than having to view the entire tree.

If the Rule Set is added to the stream, a Settings tab will become available that allows you to export the rule set in SQL format, which permits the rules to be directly applied to a database.

Click **File...Close** to close the Rule set browser window

12.6 Determining Model Accuracy

The predictive accuracy of the rule induction model is not given directly within the C5.0 node. In Lesson 15 we will introduce the Analysis node that provides information on model performance, but here we calculate accuracy ourselves with the Matrix node.

Creating a Data Table Containing Predicted Values

First, we will use the Table node to examine the fields created by the C5.0 model. When generating the C5.0 model nugget in the Models palette, that nugget will be connected to the current stream as well.

Place a **Table** node from the Output palette below the generated C5.0 nugget node named **Risk**

Connect the generated C5.0 nugget node named **RISK** to the **Table** node
Right-click the **Table** node, then click **Run** and scroll to the **right** in the table

Figure 12.19 Two New Fields Generated by the C5.0 nugget Node

ID	PANS	RISK	SUM DEBT	CHILDREN	Partition	\$C-RISK	\$CC-RISK
1		good risk	4	T	1 Training	good risk	0.711
2		bad loss	2	T	1 Training	good risk	0.711
3		good risk	4	T	1 Training	good risk	0.634
4		bad loss	4	F	2 Testing	good risk	0.634
5		good risk	3	F	1 Training	good risk	0.711
6		good risk	4	T	1 Training	good risk	0.634
7		good risk	3	F	1 Training	good risk	0.711
8		good risk	4	T	1 Training	good risk	0.634
9		bad loss	4	T	1 Training	good risk	0.634
10		good risk	4	T	1 Training	good risk	0.634
11		bad profit	3	F	1 Training	good risk	0.634
12		bad profit	2	T	1 Training	good risk	0.711
13		good risk	3	F	2 Testing	good risk	0.711
14		bad loss	2	F	1 Training	good risk	0.711
15		good risk	3	F	2 Testing	good risk	0.711
16		good risk	2	T	2 Testing	good risk	0.711
17		good risk	3	F	1 Training	good risk	0.711
18		bad profit	4	T	1 Training	good risk	0.634
19		good risk	4	T	1 Training	good risk	0.634
20		bad loss	4	T	1 Training	good risk	0.634

Two new columns appear in the data table, $\$C\text{-RISK}$ and $\$CC\text{-RISK}$. The first represents the predicted value for each record and the second the confidence value for the prediction.

Click **File...Close** to close the Table output window

Comparing Predicted to Actual Values

We will view a data matrix to see where the predictions were correct, and then we evaluate the model graphically with a gains chart.

Place a **Select** node from the **Record Ops** palette to the right of the generated C5.0 Nugget node named **RISK**

Connect the generated C5.0 nugget named **RISK** to the **Select** node

We need to use a Select node because we want to check the accuracy only on the training data. When developing a model, we don't view the performance of the model on the validation data until we are reasonably satisfied with model accuracy.

Double-click the **Select** node

Click the **Expression Builder**  button

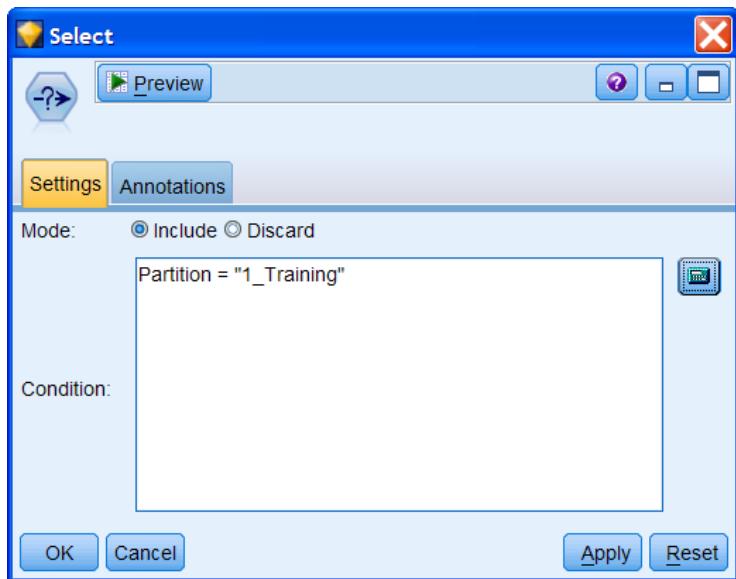
Move **Partition** from the **Fields** list box to the Expression Builder text box

Click the  (equal sign button)

Click the Select from existing field values button  and insert the value **1_Training**

Click **OK**, and then click **OK** again to close the dialog

Figure 12.20 Completed Selection for the Training Partition



Now attach a **Matrix** node to the **Select** node.

Place a **Matrix** node from the Output palette below the **Select** node

Connect the **Matrix** node to the **Select** node

Double-click the **Matrix** node to edit it

Put **RISK** in the **Rows**:

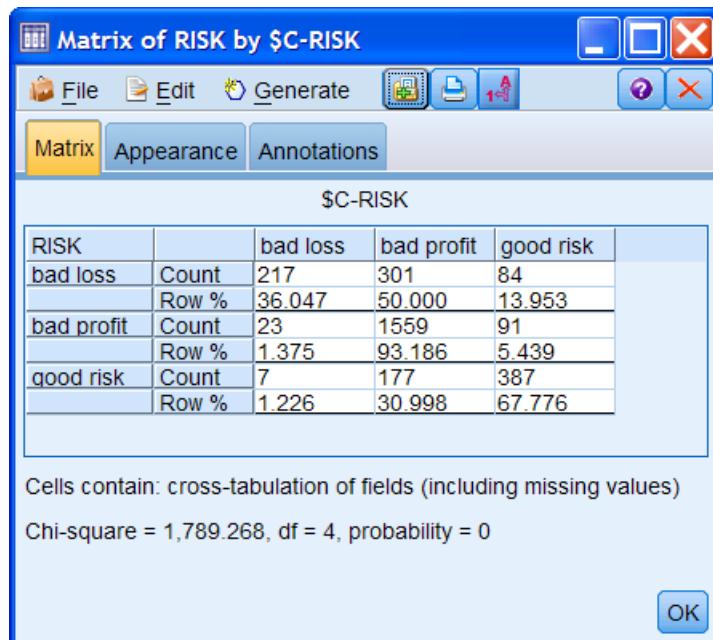
Put **\$C-RISK** in the **Columns**:

Click the **Appearance** tab

Click the **Percentage of row** option (not shown)

Click **Run**

For each actual risk category, the *Percentage of row* choice will display the percentage of records predicted into each of the outcome categories.

Figure 12.21 Matrix Output for the Training Sample

Looking at the Training sample results, the model predicts about 67.8% of the good risk category correctly, 93.2% of the bad but profitable risks, and 36.0% of the bad loss risks correctly. Is this an acceptable result? There is no absolute answer to this question. Overall, the model is fairly accurate since most customers are in the bad profit category, and the almost all these records have been predicted correctly. However, we may be more interested in avoiding a bad loss than predicting the other two categories, and model accuracy for the bad loss group is quite low.

As a consequence, this model might be considered not acceptable, and it would be back to the drawing boards, which in practice means re-running C5.0 with different settings, or trying different rule induction methods.

To emphasize the point about model evaluation, once you have modeling results from PASW Modeler, any decision about whether the model is satisfactory becomes not just statistical but also practical and related to the goals and expectations of the particular data mining project. What is acceptable, for example, in a direct mail campaign may not be acceptable for a bank offering loans to customers.

Click **File...Close** to close the Matrix windows

Evaluation Chart Node

The Evaluation Chart node offers an easy way to evaluate and compare predictive models in order to choose the best model for your application. Evaluation charts show how models perform in predicting particular outcomes. They work by sorting records based on the predicted value and confidence of the prediction, splitting the records into groups of equal size (quantiles), and then plotting the value of a criterion for each quantile, from highest to lowest.

In addition, the *Split by partition* option in the node provides an easy and convenient way to validate the model by displaying not only the results of the model using the training data, but in a separate

chart, showing how well it performed with the testing or holdout data. Of course, this assumes that you made use of the Partition Node to develop the model. Otherwise, this option will be ignored.

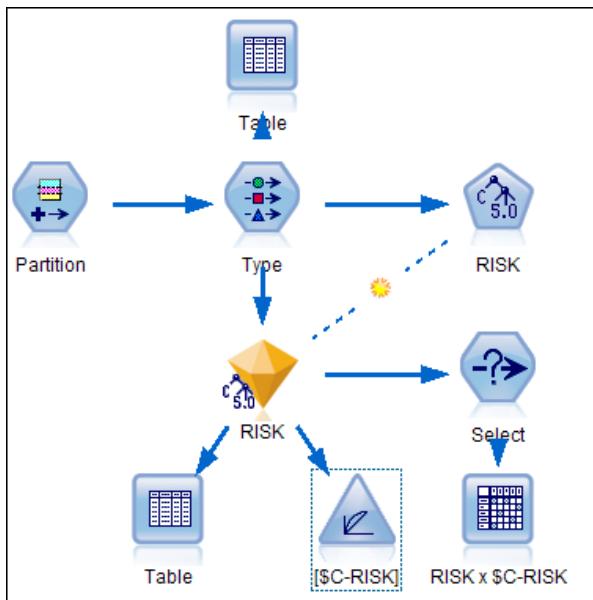
Outcomes are handled by defining a specific value or range of values as a hit. Hits usually indicate success of some sort (such as a sale to a customer) or an event of interest (such as someone given credit being a good credit risk). Flag target fields are straightforward; by default, hits correspond to *true* values. For nominal target fields, by default the first value in the set defines a hit. For the credit risk data, the first value for the *RISK* field is bad loss. To specify a different value as the hit value, use the Options tab of the Evaluation node to specify the target value in the *User defined hit* group. There are five types of evaluation charts, each of which emphasizes a different evaluation criterion. Here we discuss Gains and Lift charts. For information about the others, which include Profit and ROI charts, see the *PASW Modeler User's Guide*.

Gains are defined as the proportion of total hits that occurs in each quantile. We will examine the gains when the data are ordered from those most likely to those least likely to be in the bad loss category (based on the confidence of the model prediction).

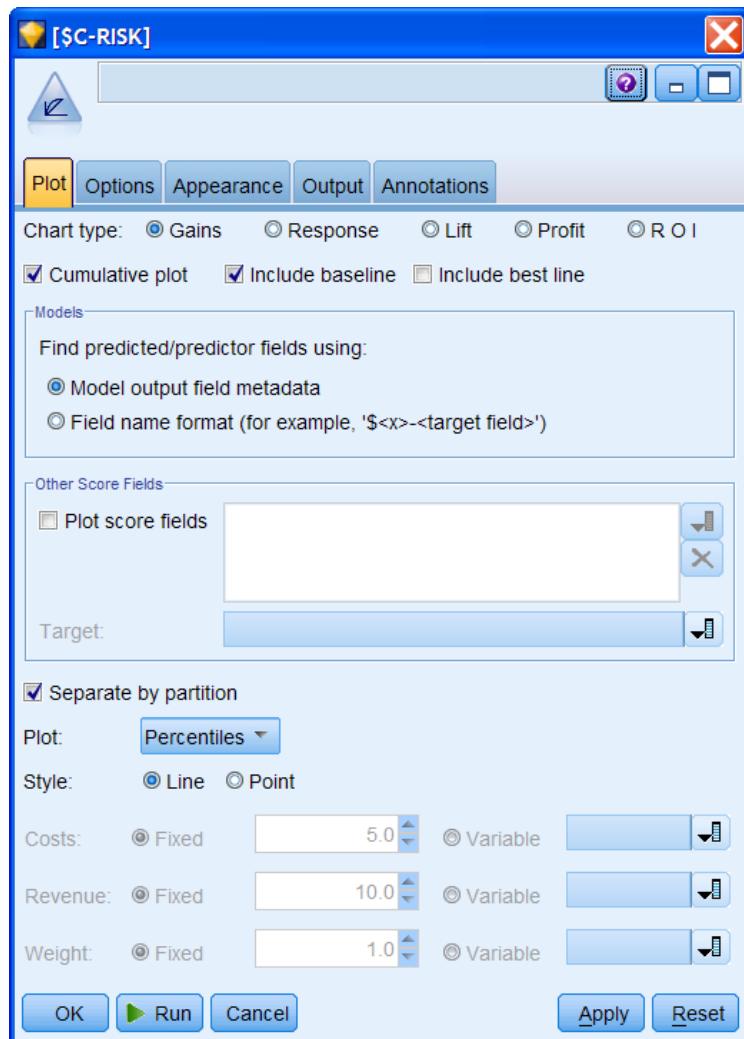
Place an **Evaluation** node from the Graphs palette near the generated C5.0 nugget named **RISK**

Connect the generated C5.0 nugget named **RISK** to the **Evaluation** node

Figure 12.22 Stream with Evaluation Node Connected to Generated Model Node



Double-click the **Evaluation** node to edit it

Figure 12.23 Evaluation Node Dialog Box

The *Chart type* option supports five chart types with *Gains* chart being the default. If *Profit* or *ROI* chart type is selected, then the appropriate options (cost, revenue and record weight values) become active so information can be entered. The charts are cumulative by default (see *Cumulative plot* check box), which is helpful in evaluating such business questions as “how will we do if we make the offer to the top X% of the prospects?” The granularity of the chart (number of points plotted) is controlled by the *Plot* drop-down list and the *Percentiles* choice will calculate 100 values (one for each percentile from 1 to 100). For small data files or business situations in which you can only contact customers in large blocks (say some number of groups, each representing 5% of customers, will be contacted through direct mail), the plot granularity might be decreased (to deciles (10 equal-sized groups) or vingtiles (20 equal-sized groups)).

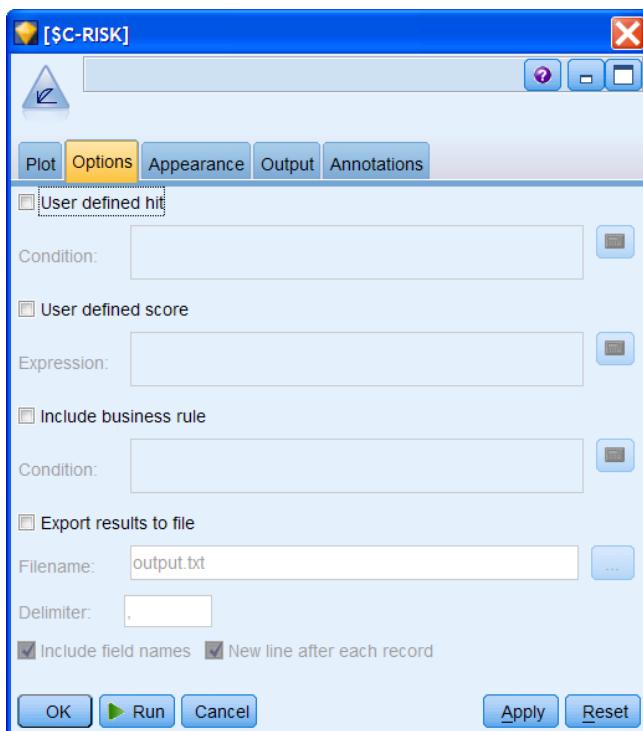
A baseline is quite useful since it indicates what the business outcome value (here gains) would be if the model predicted at the chance level. The *Include best line* option will add a line corresponding to a perfect prediction model, representing the theoretically best possible result applied to the data where hits = 100% of the cases.

The *Separate by partition* option provides an opportunity to test the model against the unseen data that was held out by the Partition node. If checked, an evaluation chart will be displayed for both the Training and Testing samples. We will turn it off because we don't want to do this now.

- Click the **Include best line** check box
- Click **Separate by partition** to deselect it
- Click the **Options** tab

To change the definition of a hit, check the *User defined hit* check box and then enter the condition that defines a hit in the Condition box. For example, if we want the evaluation chart to be based on the good risks category, the condition would be @TARGET = "good risk", where @TARGET represents the target fields from any models in the stream. The Expression Builder can be used to build the expression defining a hit. This tab also allows users to define how scores are calculated, which determines how the records are ordered in Evaluation charts. Typically scores are based on functions involving the predicted value and confidence.

Figure 12.24 Evaluation Node Options Tab

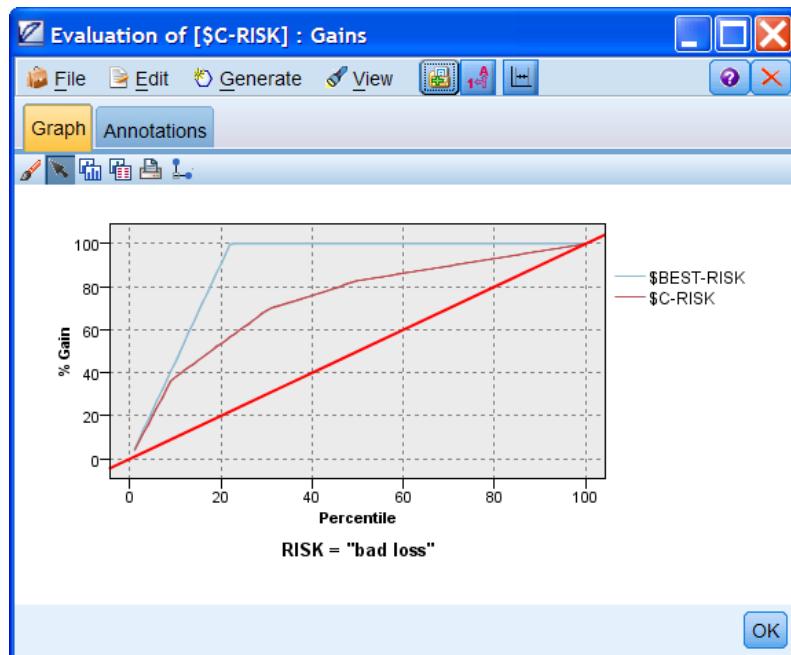


The *Include business rule* option allows the Evaluation chart to be based only on records that conform to the business rule condition. So if you wanted to see how a model(s) performs for males in the southern part of the country, the business rule could be *REGION* = "South" and *SEX* = "M".

The model evaluation results used to produce the evaluation chart can also be exported to a file (*Export results to file* option).

Click **Run**

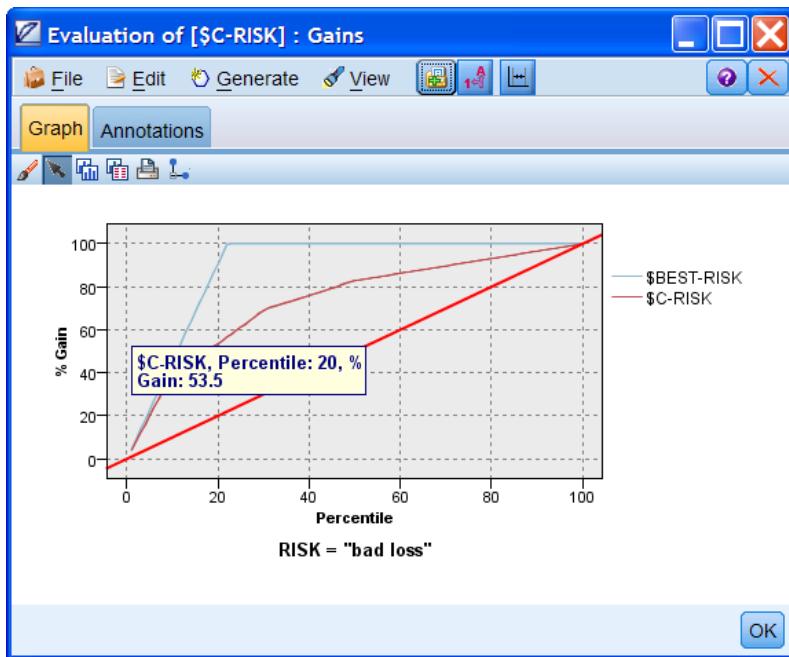
Figure 12.25 Gains Chart (Cumulative) with Bad Loss Credit Group as Target



The vertical axis of the gains chart is the cumulative percentage of the hits, while the horizontal axis represents the ordered (by model prediction and confidence) percentile groups. The diagonal line presents the base rate, that is, what we expect if the model is predicting the outcome at the chance level. The upper line (labeled *\$BEST-RISK*) represents results if a perfect model were applied to the data, and the middle line (labeled *\$C-RISK*) displays the model results. The three lines connect at the extreme [(0, 0) and (100, 100)] points. This is because if either no records or all records are considered, the percentage of hits for the base rate, best model, and actual model are identical. The advantage of the model is reflected in the degree to which the model-based line exceeds the base-rate line for intermediate values in the plot and the area for model improvement is the discrepancy between the model line and the perfect model line. If the model line is steep for early percentiles, relative to the base rate, then the hits tend to concentrate in those percentile groups of data. At the practical level, this would mean for our data that many of the bad loss customers could be found within a small portion of the ordered sample. (You can create bands on an Evaluation chart, as we did earlier on a histogram, and generate a Select or Derive node for a band of business interest.)

Here we can see that the C5.0 model line is quite steep in the early percentiles, almost matching the best possible model for the first 10% or so of the customers. After that, the line representing a perfect model continues with a steep increase between the 10th and 20th percentiles, while the results from the actual model flatten considerably (as we saw from the Matrix node, the overall accuracy of predicting bad loss is low).

If you hold your cursor over the model line at any particular point, the percentage of hits is displayed. At the 20th percentile value on the horizontal axis, we see that under the base rate we expect to find 20% of the hits (bad losses) in the first 20 percentiles (20%) of the sample, but the model produces 53.5% of the hits in the first 20 percentiles of the model-ordered sample. This is a considerable improvement over chance, but it is still only about half of the bad loss customers.

Figure 12.26 Hit Rate for Bad Losses at the 20th Percentile

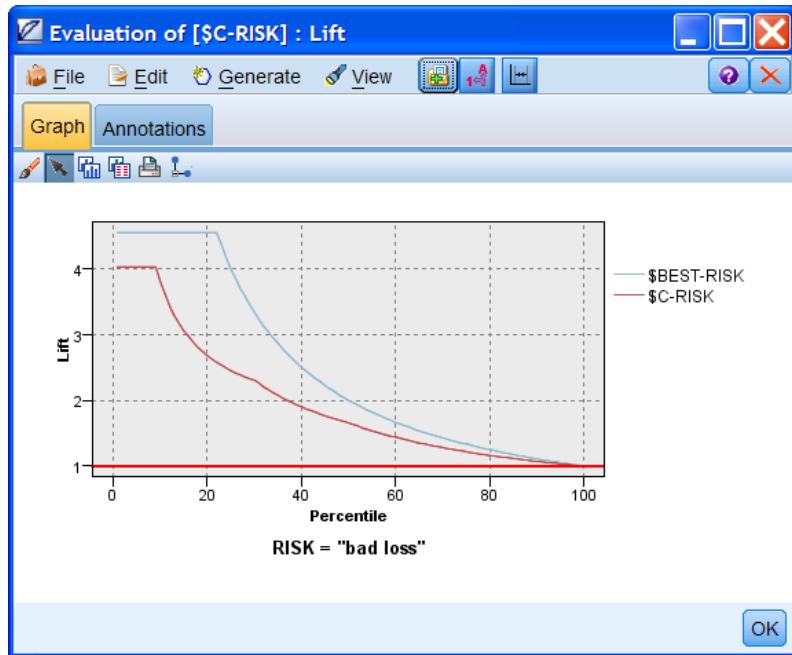
For the remainder of the percentiles (20 through 100), the distance between the model and base rate narrows, indicating that these last model-based percentile groups contain a relatively small (lower than the base rate) proportion of bad loss individuals. The Gains chart provides a way of visually evaluating how the model will do in predicting a specified outcome.

The lift chart is another way of representing this information graphically. It plots a ratio of the percentage of records in each quantile that are hits divided by the overall percentage of hits in the training data. Thus the relative advantage of the model is expressed as a ratio to the base rate.

- Close the Evaluation chart window
- Double-click the **Evaluation node** named **\$C-RISK**
- Click the **Plot** tab
- Click the **Lift Chart Type** option (not shown)
- Click **Run**

The first 10 percentiles show lift values around 4.0 (recall this is cumulative lift), providing another measure of the relative advantage of the model over the base rate. This is excellent, but the lift drops rapidly after the 10th percentile.

Figure 12.27 Lift Chart (Cumulative) with Bad Loss Credit Group as Target



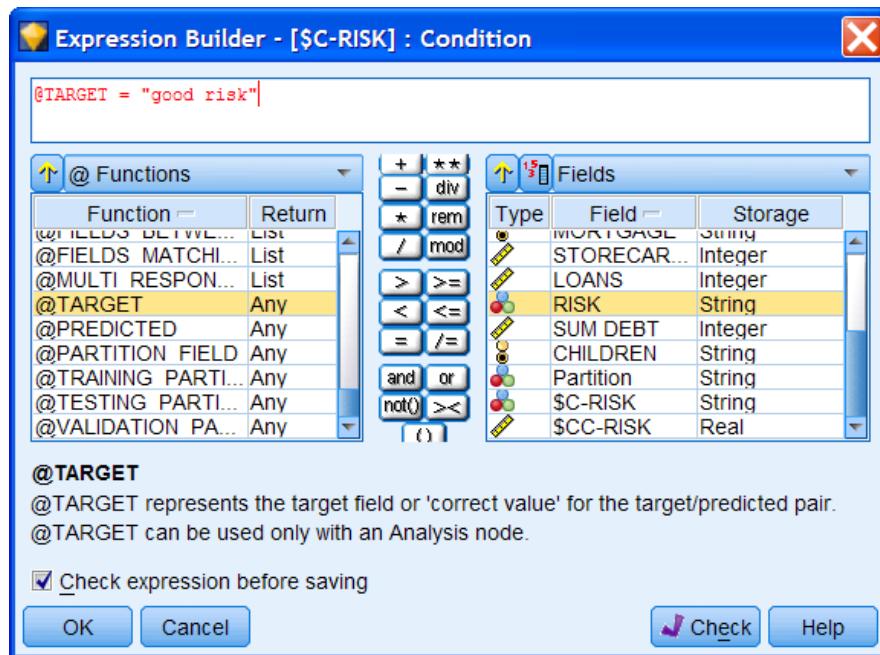
Gains charts and lift charts are very helpful in marketing and direct mail applications, since they provide evaluations of how well the campaign would do if it were directed to the top X% of prospects, as scored by the model.

Close the Lift chart window

Changing Target Category for Evaluation Charts

By default, an Evaluation chart will use the first target category to define a hit. To change the target category on which the chart is based, we must specify the condition for a *User defined hit* in the Options tab of the Evaluation node. To create a gains chart in which a hit is based on the good risk category:

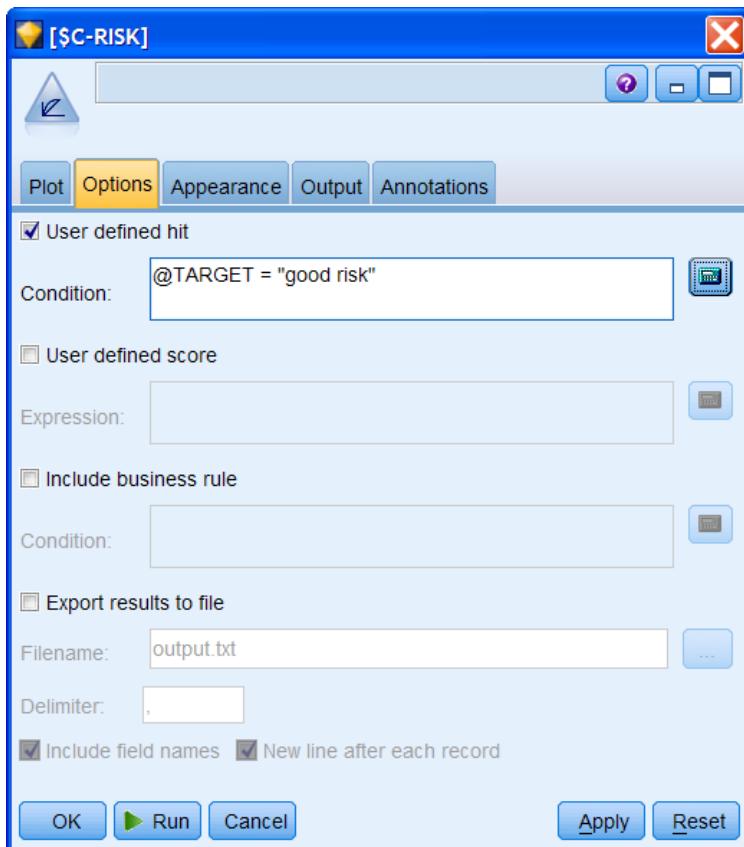
- Double-click the **Evaluation** node
- Click the **Options** tab
- Click the **User defined hit** checkbox
- Click the **Expression Builder** button  in the **User defined hit** group
- Click **@Functions** on the functions category drop-down list
- Select **@TARGET** on the functions list, and click the Insert button 
- Click the **=** button 
- Right-click **RISK** in the Fields list box, then select **Field Values**
- Select **good risk**, and then click the **Insert** button

Figure 12.28 Specifying the Hit Condition within the Expression Builder

The condition (good risk is the target value) defining a hit was created using the Expression Builder. Note the expression will be checked when OK is clicked.

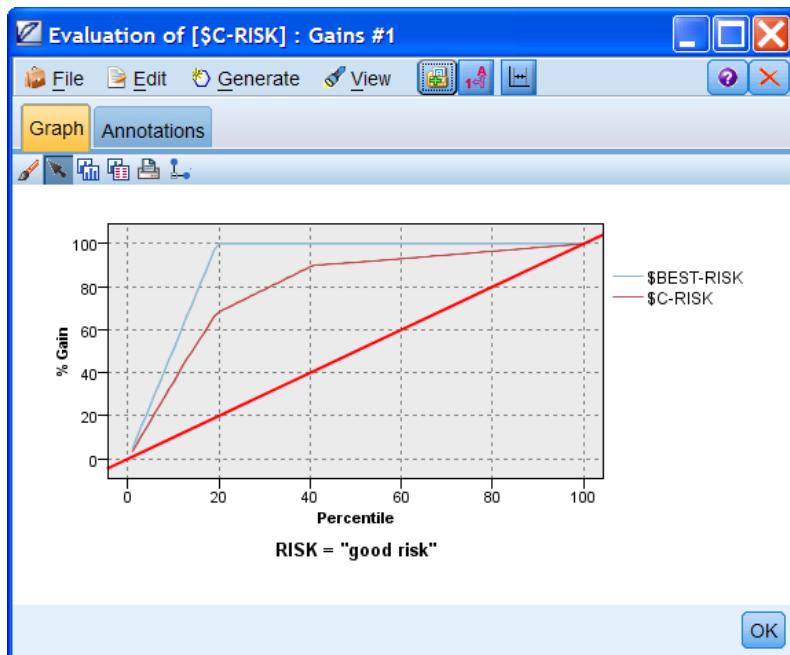
Click OK

Figure 12.29 Defining the Hit Condition for RISK



In the evaluation chart, a hit will now be based on the good risk target category. We want to return to displaying a Gains chart.

- Click **Plot** tab
- Click **Gains** option button
- Click **Run**

Figure 12.30 Gains Chart for the Good Risk Category

The gains chart for the *good risk* category is better (steeper in the early percentiles) than that for the *bad loss* category. For example, the top 20 model-ordered percentiles in the Training data chart contain over 68% of the *good risks* as opposed to the same chart when we looked at *bad losses* (that value was 53.5%)

Close the Evaluation chart window

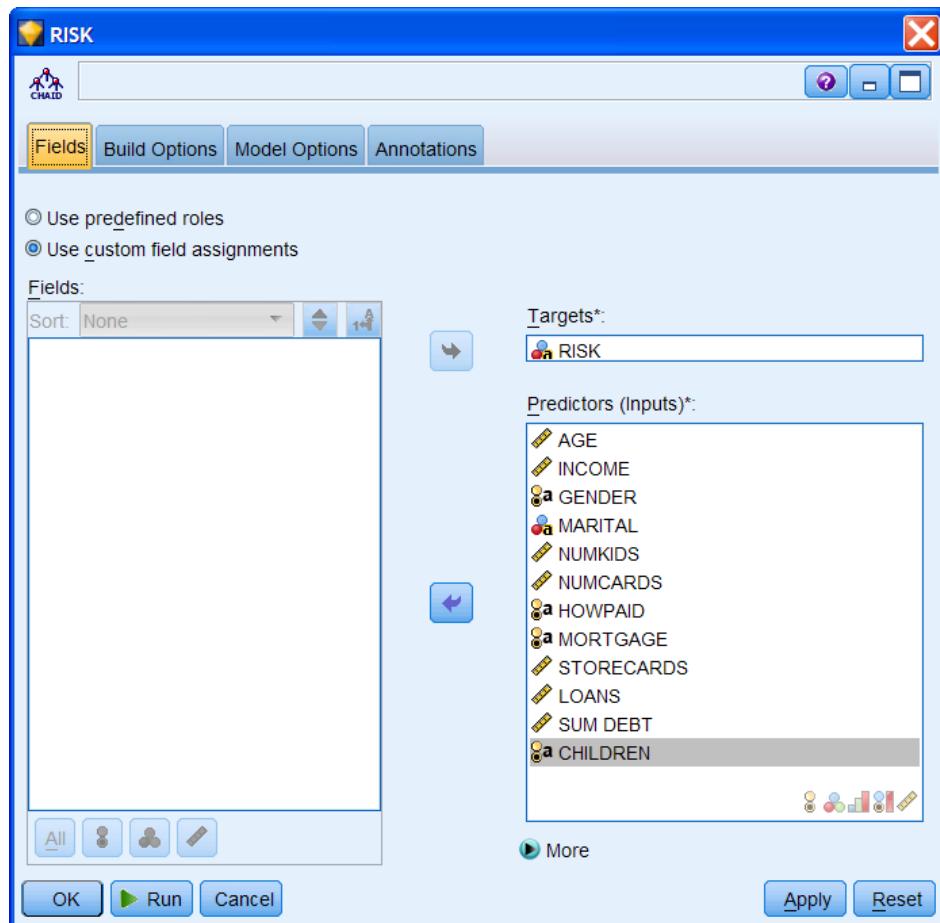
12.7 Rule Induction Using CHAID

When a model needs improvement, one approach is to use a different algorithm, and we illustrate that in this section by constructing a CHAID model to predict *RISK*. The CHAID algorithm and some of its characteristics were briefly described earlier in the lesson, so we simply apply the modeling node here.

Add a **CHAID** modeling node from the Modeling palette above the **Type** node

Connect the **Type** node to the **CHAID** node

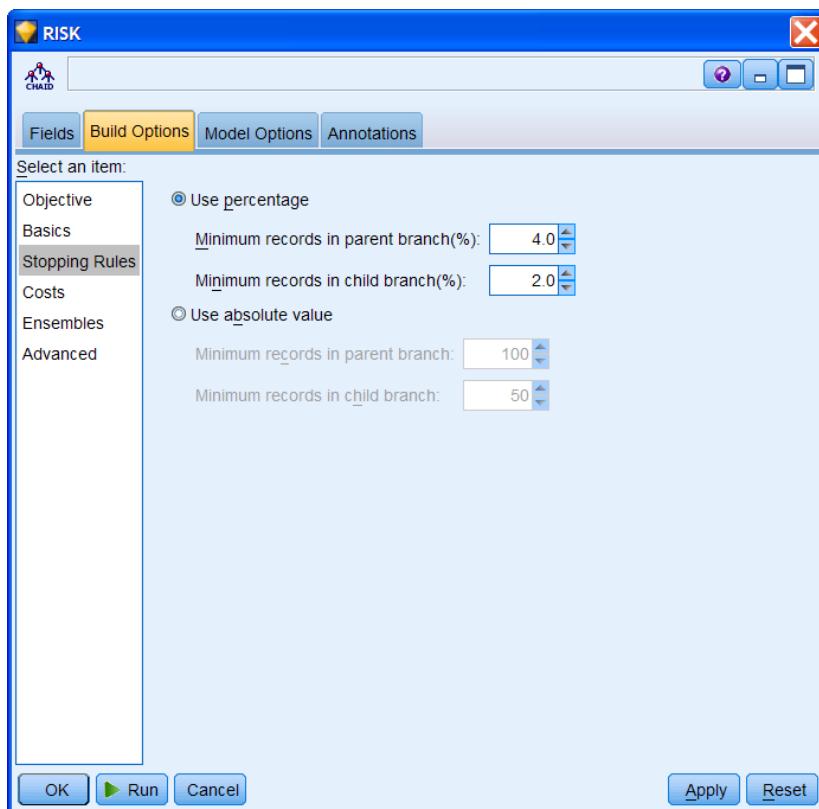
Edit the **CHAID** node

Figure 12.31 CHAID Model Dialog

There are different types of settings for CHAID compared to C5.0. One immediately apparent is the *Maximum tree depth*: which by default limits the depth of the tree to 5 levels below the root node. Under the Stopping Rules setting, we can specify stopping criteria—the minimum number of records in both a parent and child branch, similar to C5.0. There is no pruning of CHAID trees, unlike C5.0, but CHAID offers the ability to grow a tree interactively, step by step, and then manually prune the tree (*Launch interactive session* option button).

To try to match the C5.0 model as closely as possible, we will increase the default minimum size of the parent and child nodes to make the CHAID tree more likely to generalize to new data.

Click **Build Options** tab and then click **Stopping Rules** option
 Change **Minimum records in parent branch (%)** to 4
 Change **Minimum records in child branch (%)** to 2

Figure 12.32 Changing Stopping Criteria for CHAID

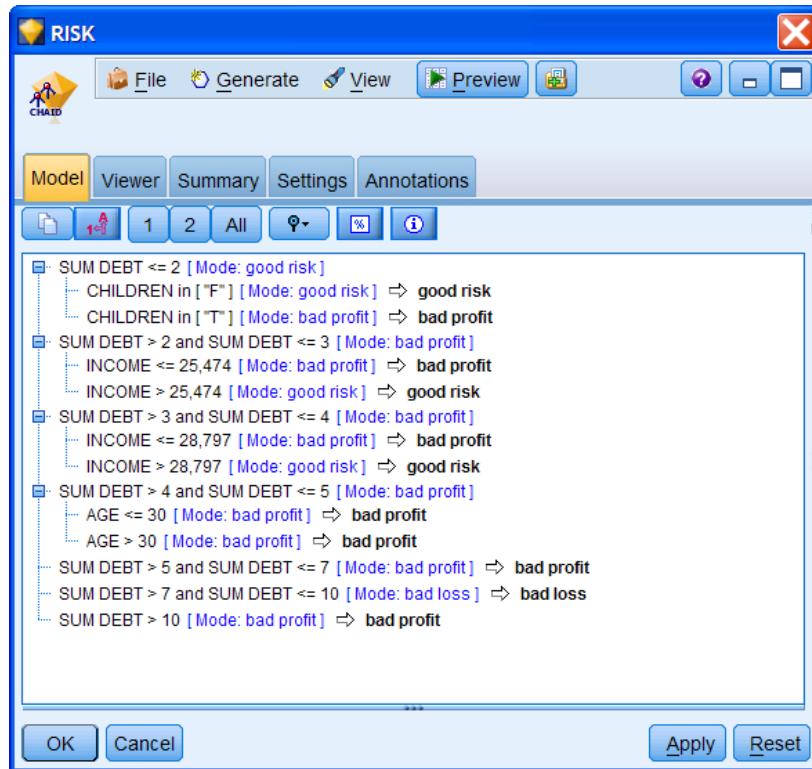
Click **Run**

Right-click the generated **CHAID** model named **RISK** on the canvas and click **Edit**

Close the right pane of the Model Brower

Click **All** button in the Model Brower

Figure 12.33 CHAID Model Fully Unfolded

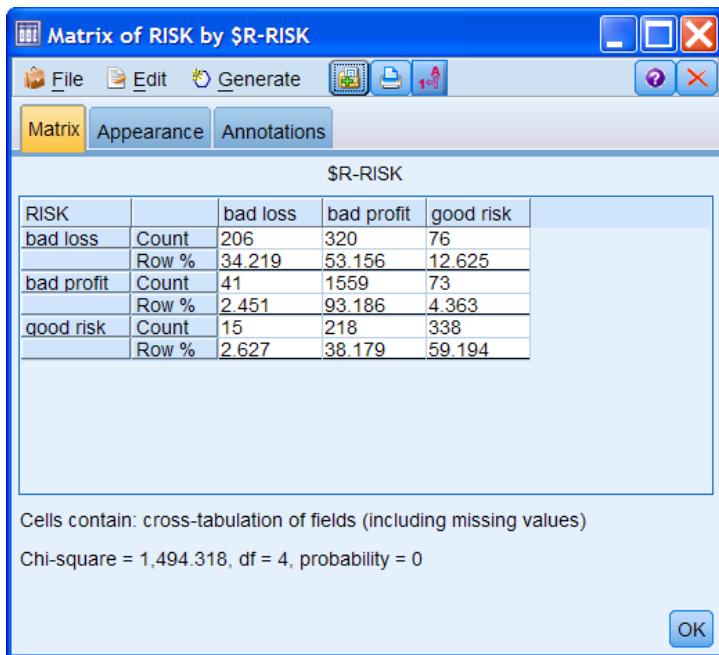


Select **Viewer** tab in the Model Browser
Widen the Tree pane to view the full tree

The tree produced by CHAID is different from the tree produced by the C5.0 model. The first split occurs not on *LOANS* but on *SUM DEBT*. The model does not use *LOANS* because this field is clearly not as important in the CHAID model as it was in the C5.0 model.

We will check the accuracy of the model in the same way we did for the C5.0 model.

Close the CHAID Model Brower window
Right-click on the **Select** node and select **Copy Node**
Right-click near the **CHAID** model and select **Paste**
Connect the **CHAID** model to the **Select** node
Add a **Matrix** node to the stream after the second **Select** node
Edit the **Matrix** node
Put **RISK** in the **Rows**:
Put **\$R-RISK** in the **Columns**:
Click the **Appearance** tab
Click the **Percentage of row** option (not shown)
Run the **Matrix** node

Figure 12.34 Matrix Output for the Training Sample for CHAID Model

The accuracy values in all the three categories are a little different with C5.0 (see Figure 12.21).

You may also want to request a Gains or Lift chart for the CHAID model.

We will save this stream for use in later lessons.

Close the Matrix window
Click **File...Save Stream As**
Navigate to the **c:\Train\ModelerIntro** directory (if necessary)
Type **Rule Induction** in the File name: text box
Click **Save**

Summary

In this lesson you have been introduced to the five rule induction algorithms within PASW Modeler.

You should now be able to:

- Build a rule induction model using C5.0
- Browse the model in the form of a decision tree
- Generate a rule set from the model
- Look at the instances and confidence of a decision tree branch and interpret the model
- Change the hit condition in an evaluation chart
- Generate a CHAID model

Exercises

In this session we will attempt to predict the field *Response to campaign* using a C5.0 model and then a CHAID model. For the purposes of modeling, we will use a version of the Charity data with more records in this lesson and the remaining exercises.

1. Begin with a clear Stream canvas. Place a Statistics file node on the Stream canvas and connect it to the *charitybig.sav* (Read labels as names, Read labels as data).
2. Attach a Type and Table node in a stream to the source node. Run the stream and allow PASW Modeler to automatically define the measurement level of the fields. Edit the Type node as needed to set the appropriate measurement level for each field. Specify *Response to campaign* as Flag.
3. Connect a Partition node and specify a 70% training and 30% test partition. Connect a Type node to the right of the Partition node.
4. We will attempt to predict *Response to Campaign* using the fields listed below. Set the role of all five of these fields to Input and the *Response to campaign* field to Target and measurement level Flag. Set all other fields to role None.
Pre-campaign expenditure
Pre-campaign visits category
Gender
Age
Mosaic Bands (which should be changed to measurement level Nominal)
5. Connect a C5.0 node to the Type node, click on Expert mode and change the minimum number of records per child branch to 150. Then run the model.
6. Once the C5.0 rule induction model has been generated, browse the C5.0 Rule node in the Generated Models palette. Look at variable importance. Use the Tree Viewer to browse the model. Generate a rule set for this model.
7. Connect a Select node to the C5.0 model and select only the Training Partition records. Connect a Matrix node to the Select node and create a matrix of actual response against predicted response. How well is the model doing in predicting Responders?
8. Connect an Evaluation node and request a gains chart with the target category of "Responder" and include the "Best Fit" line. (Hint: Options tab and specify User-defined hit to define the target category.)
9. Connect a CHAID node to the Type node. On the Build Options tab, select Stopping Rules and change the Stopping criteria to 6% for parent branch and 4% for child branch. Run the node and perform the same evaluations as you did for the C5.0 node. Are the models in agreement?
10. Save the stream with the name *ExerLesson12.str*.

Lesson 13: Automating Models for Categorical Targets

Objectives

- Demonstrate how to perform automated modeling for a categorical target field

Data

Throughout this lesson we will continue using the credit risk data (*Risk.txt*) introduced in the previous lessons.

13.1 Introduction

As illustrated in the last few lessons, developing more than one model is commonplace in data-mining projects. To further streamline this process, PASW Modeler offers the Auto Classifier node, which can build several different types of models to predict flag or nominal fields. Supported model types include Neural Net, C&R Tree, QUEST, CHAID, C5.0, Logistic Regression, Decision List, Bayes Net, Discriminant, Nearest Neighbor, and SVM.

The node generates a set of models based on specified options and ranks the best candidate models according to a criterion you select.

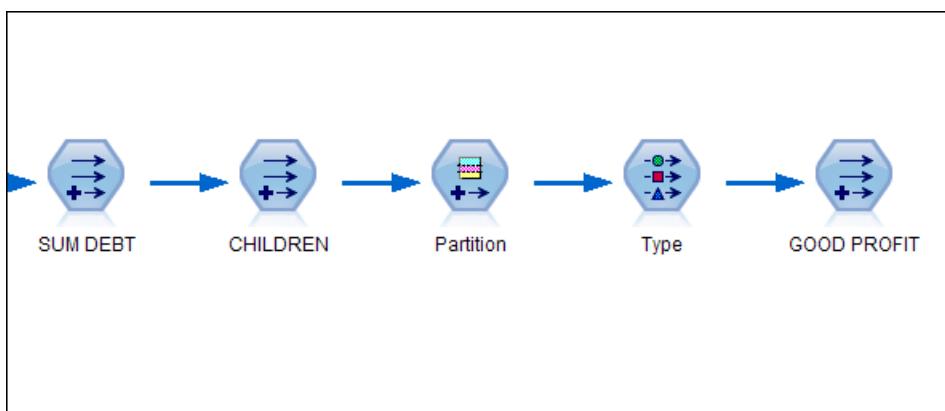
The use of the Auto Classifier node can reduce the time to build models for a flag target fields or a nominal target field substantially. Once you find one or more models that look promising, the node can create a model node in the model manager that you can then use to explore and investigate the models in more detail.

We will use the Auto Classifier node in this lesson to predict those customers who are a *good risk* for the financial institution. To do so, we will collapse the other two categories into one. This will allow the models to focus on only the most important distinction.

13.2 Creating a Flag Field

We will work with an existing stream file that contains just the essential nodes for this example.

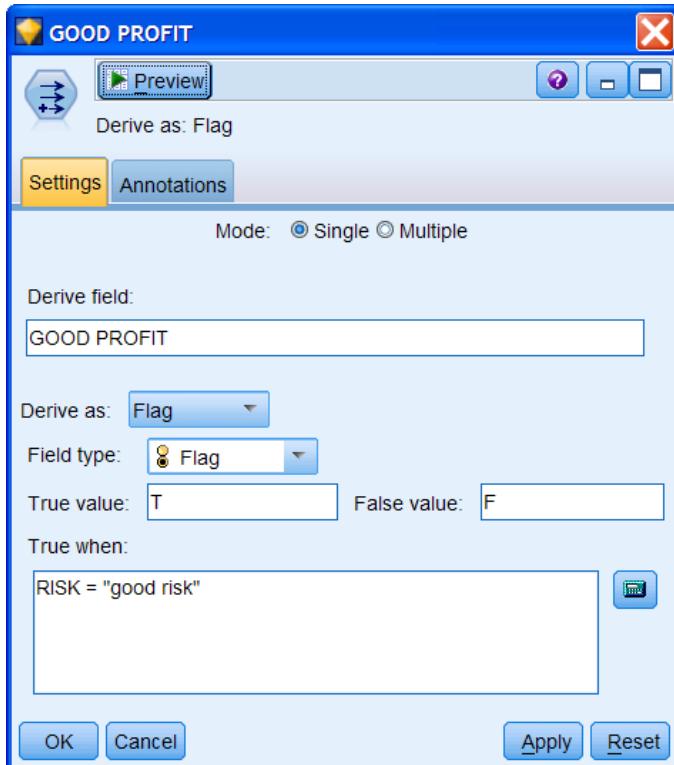
Click **File...Open Stream**, and then navigate to the **c:\Train\Modeler\Intro** directory
Double-click **Auto Classifier.str**

Figure 13.1 Auto Classifier Stream

There is a Partition node with the same settings as in Lesson 12. The Derive node labeled *GOOD PROFIT* creates the flag field we will use for modeling. We'll edit it to see how the field is constructed.

Edit the node labeled *GOOD PROFIT*

The Derive type is *Flag*. The value of *GOOD PROFIT* will be *T* when the value of *RISK* is *good risk*, else it will be false. This simple operation will create the new field with only two values (there is no missing data to be concerned with).

Figure 13.2 Creating the Flag Field *GOOD PROFIT*

Whenever you create a new field, it is good practice to check your work. Even if you have the logic correct, you can make a typing error that isn't a CLEM language error. One way to check the Derive node is to use a Matrix node to see the relationship between the original field *RISK* and the new field *GOOD PROFIT*.

Close the Derive node

Add a **Matrix** node from the Output palette to the stream above the Derive node

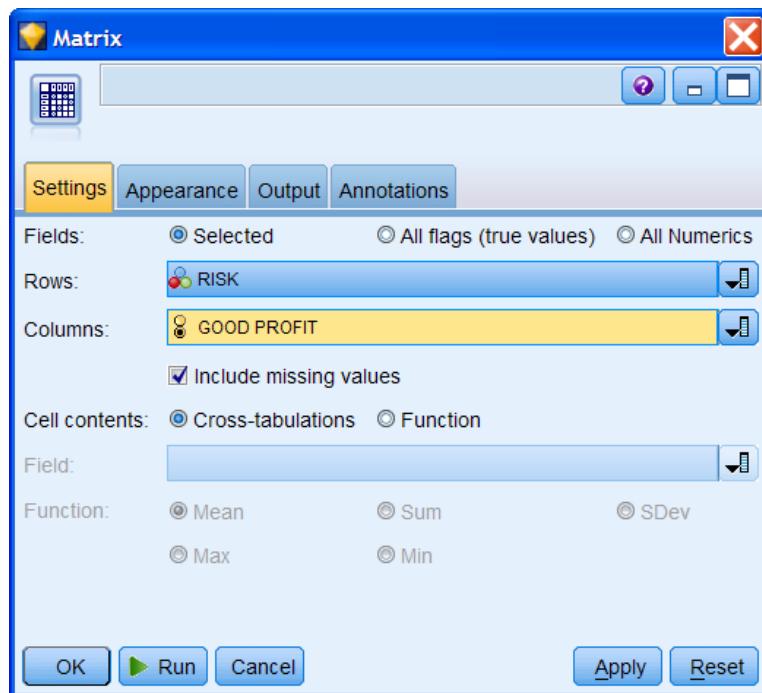
Connect the **GOOD PROFIT** Derive node to the **Matrix** node

Edit the **Matrix** node

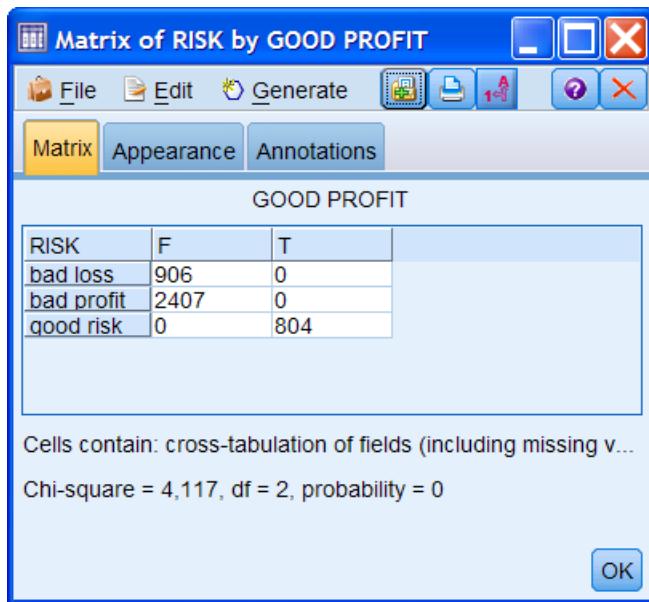
Select **RISK** as the Rows field

Select **GOOD PROFIT** as the Column field

Figure 13.3 Matrix Node Specification for RISK and GOOD PROFIT



Click **Run**

Figure 13.4 Matrix of RISK and GOOD PROFIT

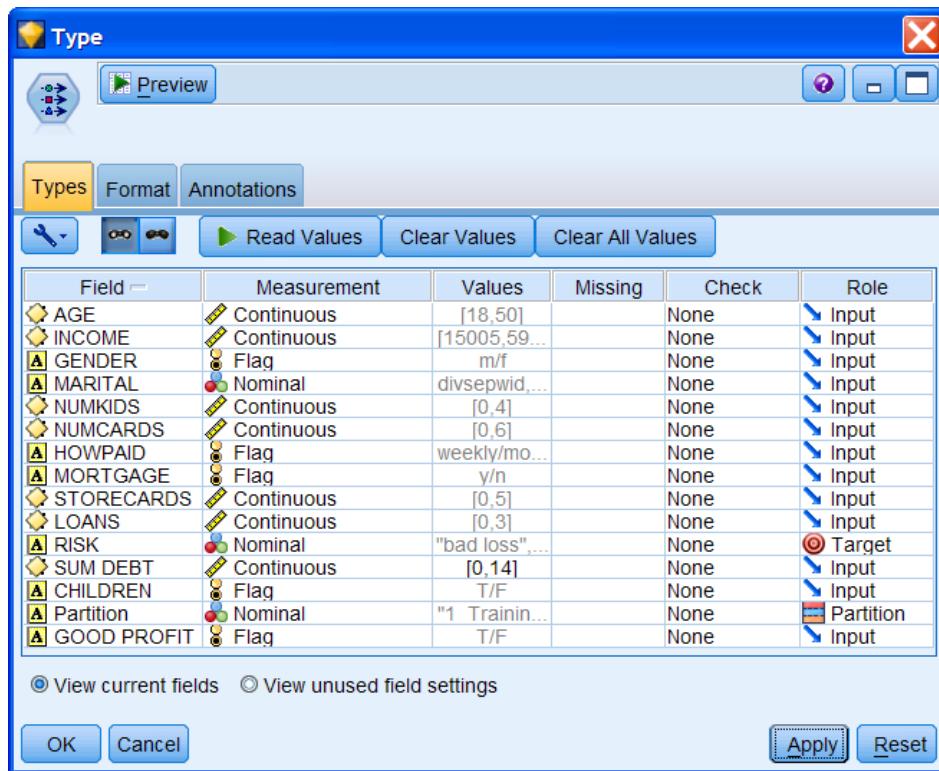
We observe the expected relationship. Both *bad loss* and *bad profit* are coded *F* on *GOOD PROFIT*, and *good risk* is coded *T*.

Before adding an Auto Classifier node, we should retype the data because we have created a new field which will be the target for the models. But we can reuse the existing Type node to save time.

- Close the Matrix node
- Right-click the **Type** node attached to the data source node and select **Copy Node** from the context menu
- Right-click near the **GOOD PROFIT** node and select **Paste**
- Attach the **GOOD PROFIT** node to the pasted **Type** node
- Add a **Table** node from the Output palette to the stream below the new **Type** node
- Connect the new **Type** node to the **Table** node
- Run the **Table** node

After the Table has run:

- Close the Table window
- Edit the new **Type** node

Figure 13.5 Type Node Settings

The new field *GOOD PROFIT* has measurement level Flag. Also, we need to change the Role for the new field and for *RISK*.

Click in the Role column for **RISK** and change the Role to **None**

Click in the Role column for **GOOD PROFIT** and change the Role to **Target** (not shown)

Click **OK**

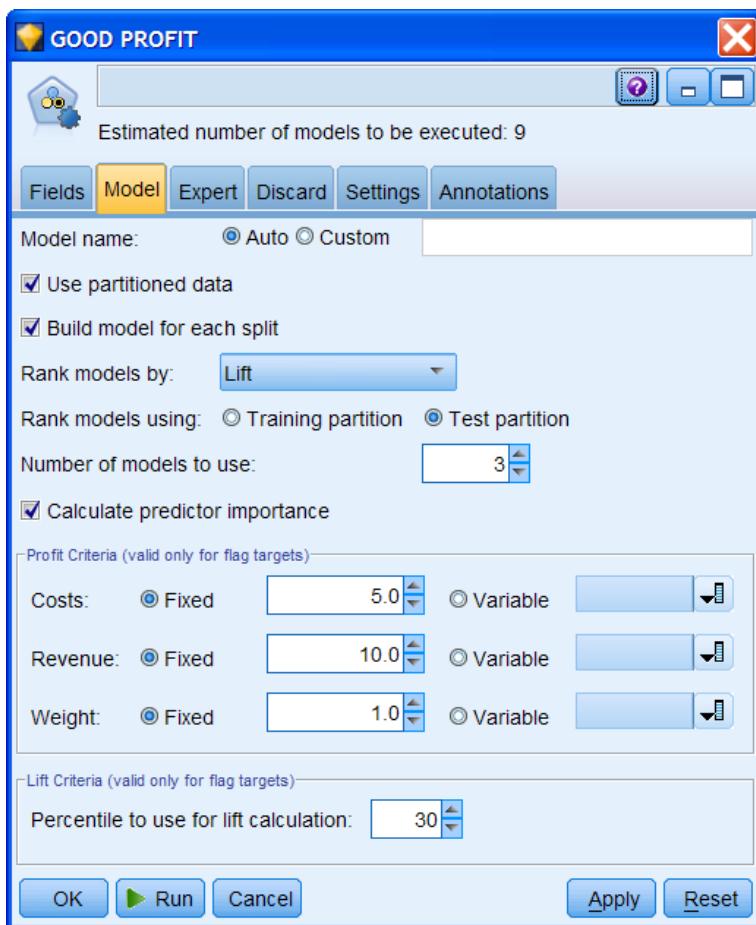
13.3 Using the Auto Classifier

Now we are ready for automated modeling.

Add an **Auto Classifier** node from the Modeling palette to the stream to the right of the **Type** node

Attach the **Type** node to the **Auto Classifier** node

Edit the **Auto Classifier** node

Figure 13.6 Auto Classifier Node

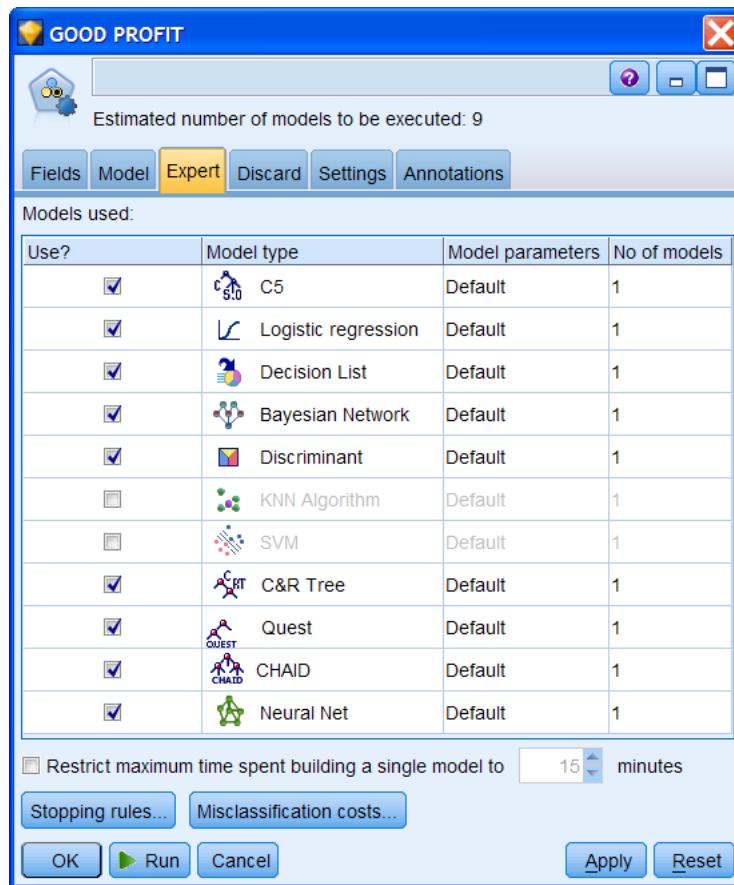
The Auto Classifier node has several settings to allow you to compare the models on the most relevant criteria. By default, models are ranked on profit, but this is only applicable if costs and revenue are defined (note that there are default values that should not be accepted unless they specifically apply to these data). Other choices for ranking include overall accuracy, lift, number of fields, and area under the curve.

The Auto Classifier recognizes the partitioning of the data, and it will build models on the Training data and test them on the Test data. The models by default will be ranked on the Test data, but this is not the optimal choice. As we explained in previous lessons, models should not be tested until you judge them ready for testing and have tried various model settings and then explored the results.

Click the Rank models by: dropdown and select **Overall accuracy**

Click **Training partition** for Rank models using: option

Click **Expert** tab

Figure 13.7 Auto Classifier Expert Tab

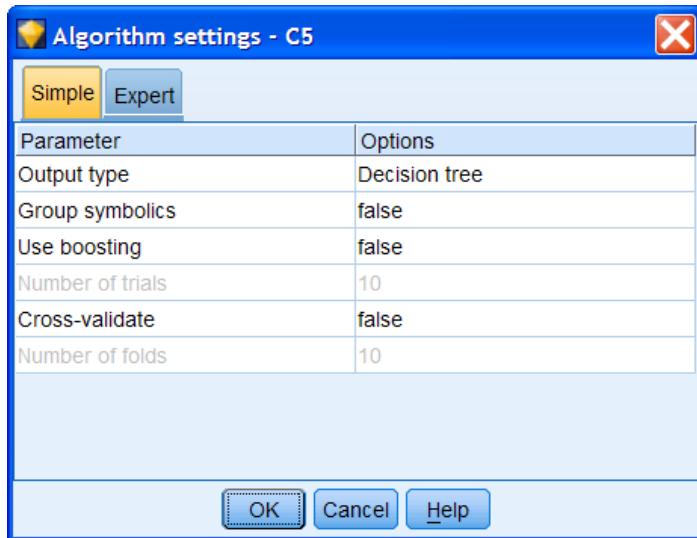
Notice that there is an option to restrict the amount of time spent building a single model (there is another option under *Stopping rules* to restrict the total time for all model building).

Model parameters, such as the depth of a tree or the number of records in parent and child branches, can be set for each model in the *Model parameters* column by selecting *Specify* from a dropdown list. We will deselect the models that are not decision trees.

Click the check boxes for **Neural Net**, **Logistic regression**, **Bayesian Network**, and **Discriminant** to deselect them

Let's look at the ability to change model parameters. Since we are familiar with C5.0 models, we'll review those settings.

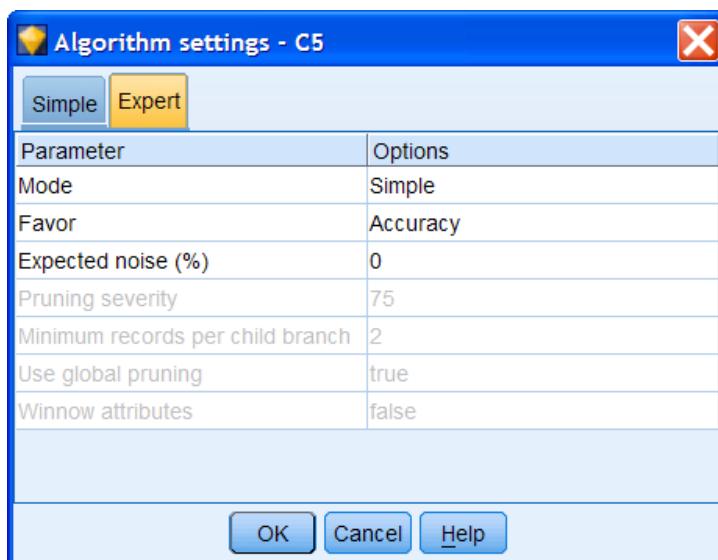
Click in the Model Parameters cell for **C5.0** and select **Specify** from the dropdown list

Figure 13.8 Algorithm Settings for C5.0, Simple Tab

The specific options are similar to those available in the separate modeling nodes (here C5.0), with the difference that rather than choosing one setting or another, you can choose as many as you want to apply in most cases. For example, if comparing Neural Net models, you can choose several different training methods, and try each method with and without a random seed. All possible combinations of the selected options will be used, making it very easy to generate many different models in a single pass. Use care however, as choosing multiple settings can cause the number of models to multiply very quickly.

By default the model won't group categorical fields, so let's change that to request a model for each.

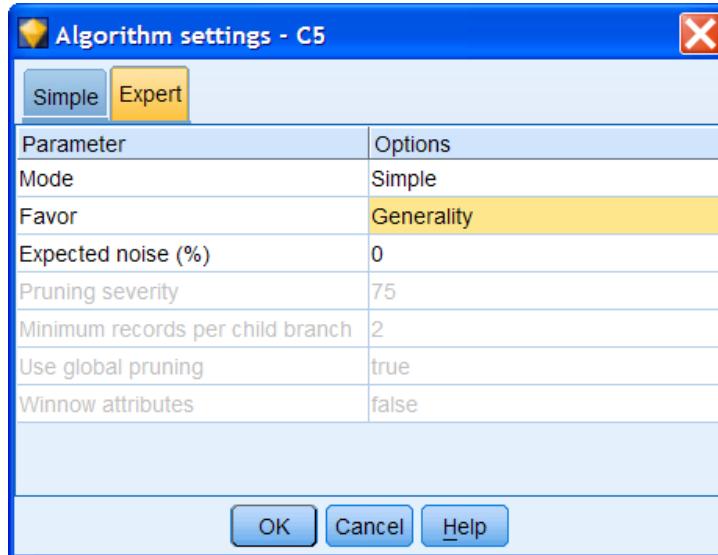
Click in the Options cell for **Group symbolics** and select **Both** (not shown)
Click the **Expert** tab

Figure 13.9 Algorithm Settings for C5.0, Expert Tab

Some choices here are grayed out and thus inactive. You can activate them by changing the *Mode* setting to *Expert*. Let's have models created that favor generality. Both models to be created will use this setting.

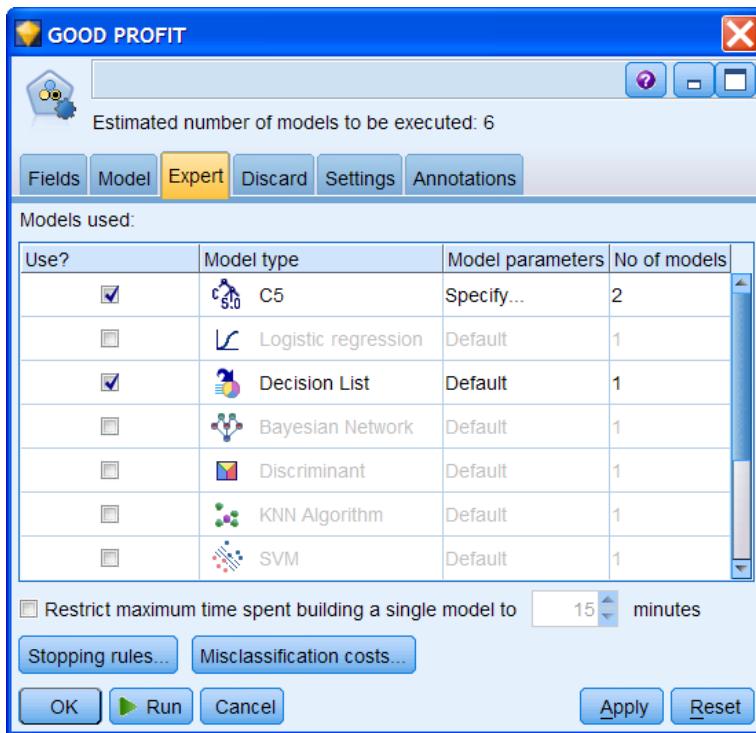
Click in the Options cell for **Favor** and select **Generality**

Figure 13.10 Setting for Favor Changed to Run Models with Both Settings



Click **OK**

In the Expert tab window, we now see that 2 models will be created with C5.0.

Figure 13.11 Two Models Requested Using C5.0

Six models will be constructed with the current selections.

Misclassification Costs

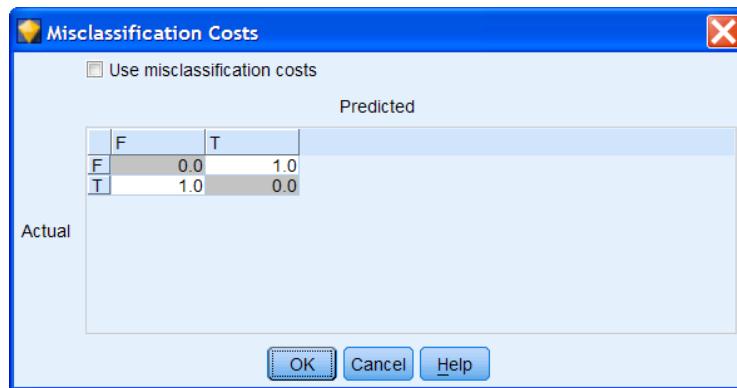
In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

Misclassification costs are not taken into account when ranking or comparing models using the Auto Classifier node. A model that includes costs may produce more errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of less expensive errors.

Click the **Misclassification costs** button

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select *Use misclassification costs* and enter your custom values into the cost matrix.

Figure 13.12 Misclassification Costs Dialog

We won't add to the complexity of this example by specifying costs, and we don't know what the relative costs of the two errors should be, in any case.

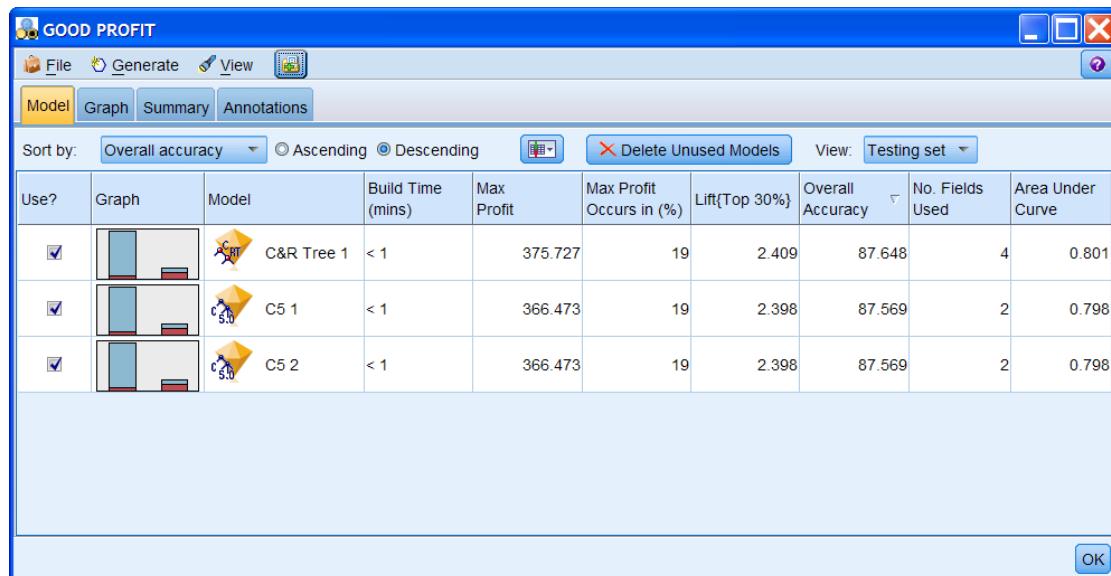
Click OK

The **Discard** button opens a dialog that allows you to tell PASW Modeler to drop models that don't meet certain standards of accuracy, lift, profit (if defined), or that use too many fields, among other options. We won't discard any models here because only 6 will be constructed.

Click Run

After the execution, Right-click the generated model nugget then

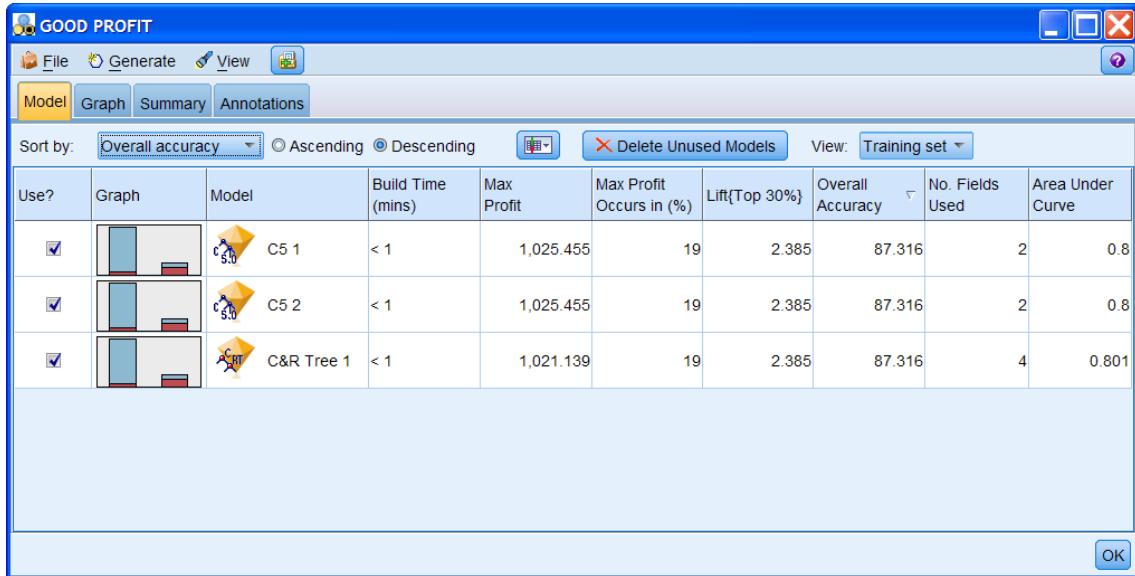
Click **Browse** from the Context menu

Figure 13.13 Auto Classifier Model Evaluation Report (Default)

Although we requested that the models be ranked by overall accuracy on the Training data, by default the Testing set results are displayed and the models are ranked by Max profit. We shouldn't look at them yet.

Select the **Training set** on the **View** drop-down list
 Select the **Overall accuracy** on the **Sort by** drop-down list (if necessary)

Figure 13.14 Auto Classifier Model Evaluation Report, Training Set

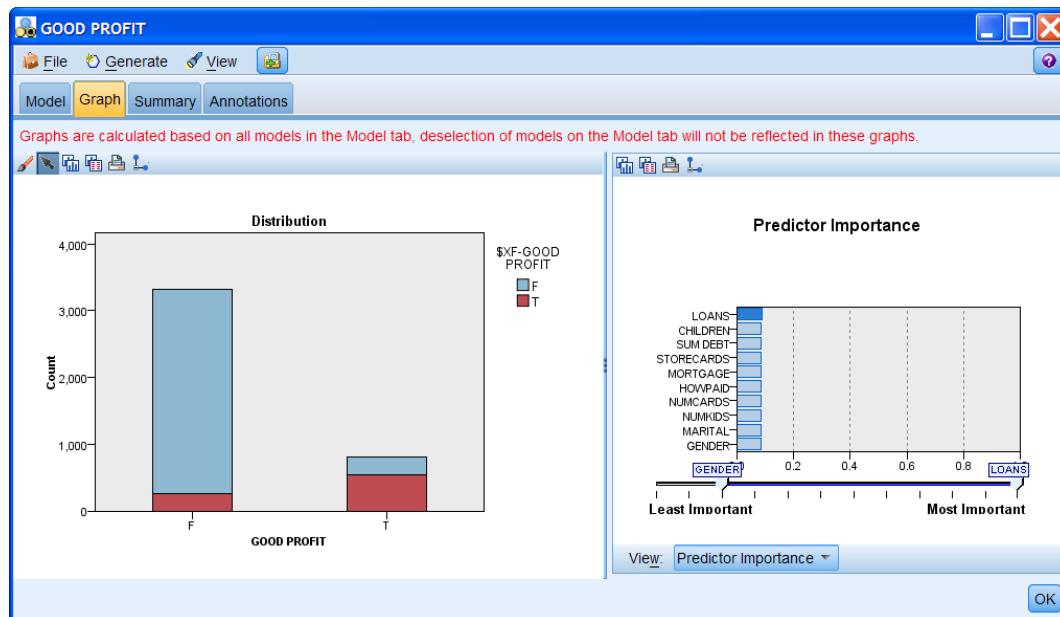


The best three models on the Training data are C&R Tree, and the two C5.0. The number of fields used by each model is listed, with both C5.0 models using only two fields. Information on each model is listed in the Summary tab, including the build settings and fields used. The number after each model is a reference value to differentiate multiple versions of the same type of model. Models can be sorted by several criteria, such as Lift, profit, number of fields, and area under the curve.

Also displayed is a thumbnail of a bar chart showing the distribution of actual values, overlaid with the predicted values, to give a quick visual indication of how many records were correctly predicted in each category. You can double-click on a thumbnail to generate a full-sized graph. The full-sized plot includes up to 1000 records and will be based on a sample if the dataset contains more records.

Let's look in more detail at the graph for the model that is the combination of the three models at hand).

Click **Graph** tab

Figure 13.15 Overlay Bar Chart of Actual and Predicted Value of GOOD PROFIT

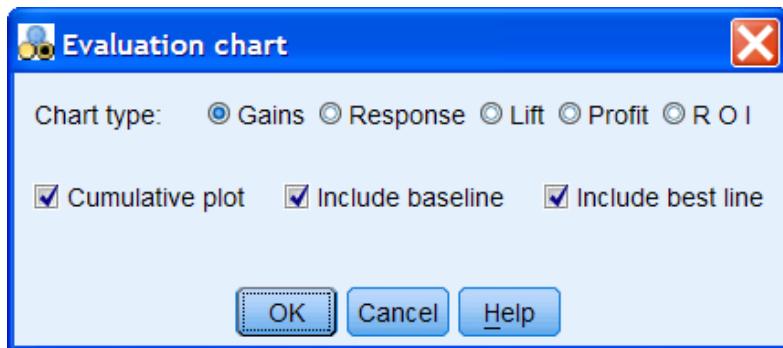
The predicted value ($\$XF\text{-}GOOD\ PROFIT$) is overlaid on the actual value of the target field. Ideally, if the model was 100% accurate, each bar would be of only one color because the overlap would be perfect. Here, we observe that prediction from the model is more accurate for the *False* category of *GOOD PROFIT*, but less accurate for those customers who were a good profit (*True*).

Note

The distribution of *GOOD PROFIT* is somewhat skewed, with about four times as many *False* records as *True*. This can make it somewhat difficult for a model to predict the category with fewer records, as is true for the C&R Tree model. Ideally, the training data will have an equal number of records in each category of the target. This requires you to over sample records from the smaller category. Note that the model must be tested eventually on data with the population distribution of the target field.

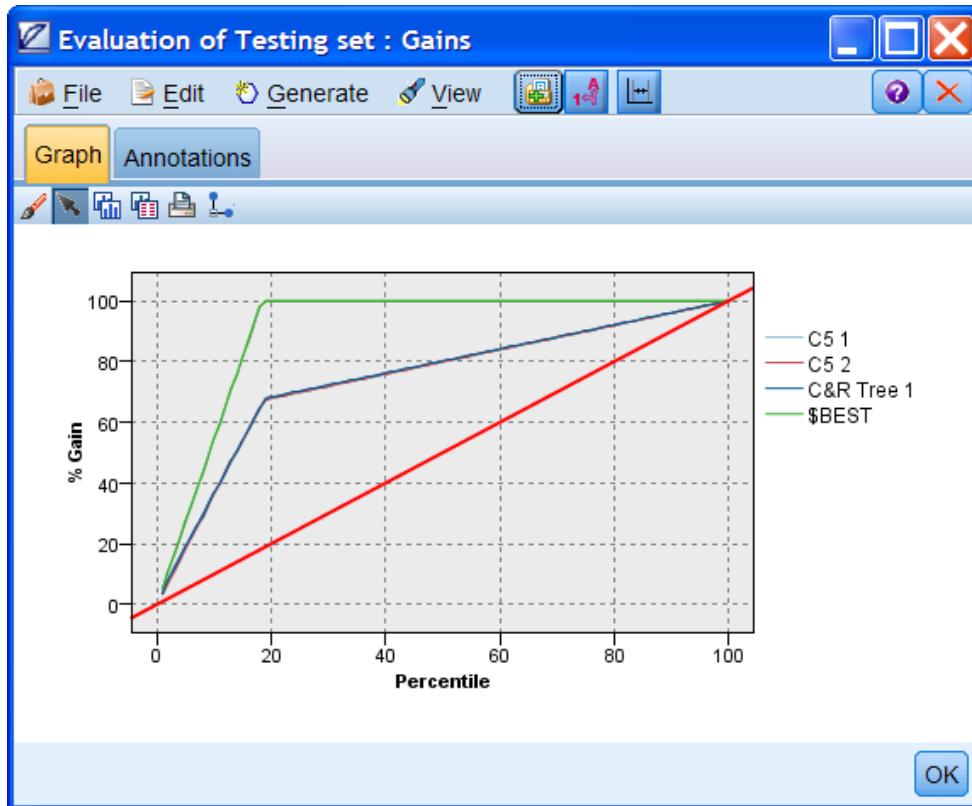
We don't have time in this lesson to actually explore each model, so let's simply pick the two or three best performing models, based on their overall accuracy on the Training data. Looking closely at the model results, it appears that the results for the two C5.0 models are identical, so we need to use only one of them (changing how categories are grouped made no difference), plus C&R Tree and possibly Decision List. First, though, we'll see how each performs on the Testing data partition.

- Click the **Model** tab
- Select the **Testing set** on the **View** drop-down list
- Click **Generate...Evaluation Chart(s)**

Figure 13.16 Evaluation Chart Generate Dialog Box

You can select which type of Evaluation chart to produce and what should be included.

Click **OK**

Figure 13.17 Evaluation Chart for Testing Data for C5.0 and C&R Tree Models

The model performance on the Gains chart is so similar that the three lines are almost completely superimposed over each other. Only by increasing the size of the chart can the lines be discriminated. Although this output doesn't help us decide between the three models for predicting the *good risk* group, the ease of creating Evaluation charts from the Auto Classifier Model Results window makes model comparison simple.

Although we won't demonstrate this option in this lesson, generated models can also be saved to the Model manager for use in the PASW Modeler stream for additional assessment, or to make predictions on new data. We will save the stream for use in later lessons.

Close the Evaluation Chart window
Close the Auto Classifier Results window
Click **File...Save Stream As**
Save the stream file as **Auto Classifier Models**

Summary

In this lesson you have been introduced to the Auto Classifier. You should now be able to:

- Use the default settings of the Auto Classifier to predict a flag field
- Make changes to the model specifications to request additional models
- Use the Auto Classifier Model Results to evaluate model performance

Exercises

In this exercise, we will use the Auto Classifier to predict Response to campaign in the file *CharityBig.sav* and generate evaluation charts to evaluate the models.

1. Open the *ExerLesson12.str* stream (or use the backup file *Backup_ExerChapter12.str*). Delete all nodes downstream from the Type node (we then only have a data source node, a Partition Node and a Type node).
2. Review the measurement level for *Response to Campaign* field and specify it as a Flag if it is not already. Leave the Role specifications the same.
3. Attach an Auto Classifier node to the Type node. Edit the Auto Classifier node to rank the models by “Overall Accuracy” using the “Training partition.” Run the node.
4. Review the results for the Training set first. Notice the wide range of accuracy of the models. And, notice that the C5.0 model is clearly the best. Now review the results for the Testing set. Are the top three models (ranked by overall accuracy) different than for the Training set?
5. Generate evaluation charts for the top three models.
6. Save the stream as *ExerLesson13.str*.

Lesson 14: Automating Models for Continuous Targets

Objectives

- Demonstrate how to perform automated modeling for continuous fields

Data

We use the Statistics data file *customer_offers.sav*, which contains customer data from a telecommunications company. It includes demographic data, information on the use of telecommunications services, lifestyle information, and financial data.

14.1 Introduction

In the previous lesson we learned how to automate the creation of models to predict a flag target. PASW Modeler provides a similar capability to construct models for continuous fields with the Auto Numeric node. The node works in the same manner as the Auto Classifier node, allowing you to choose the algorithms to use and to experiment with multiple combinations of options in a single modeling pass. Models can be compared based on correlation, relative error, or the number of fields used. There are six different types of models available.

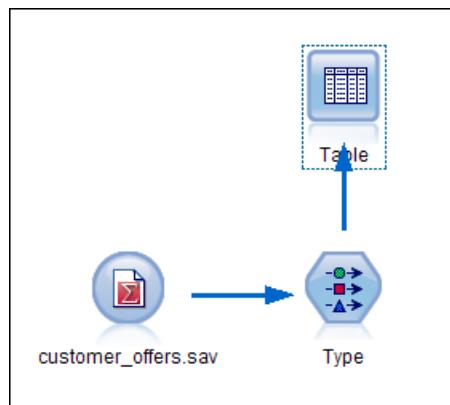
In this lesson we will use the Auto Numeric node to predict the amount of long distance service used by subscribers to a telecommunications firm, using demographic and lifestyle fields. This is the same data file used when we looked for anomalous data in Lesson 5.

14.2 Auto Numeric Stream

We will work with an existing stream file that contains the key nodes for this example.

Click **File...Open Stream**, to the **c:\Train\ModelerIntro** directory
 Double-click **Auto Numeric.str**

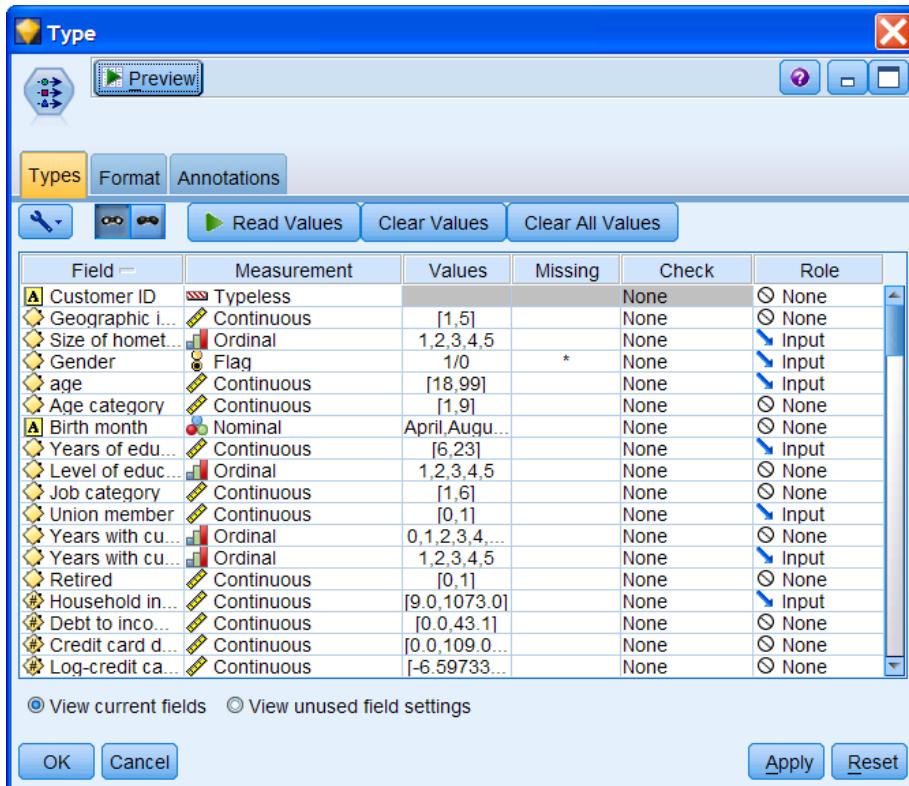
Figure 14.1 Auto Numeric Stream



There is a Statistics File source node to read the data file to begin the stream. It has the setting to read variable labels as PASW Modeler field names. The Type node already has the model settings to use with the Auto Numeric node.

Edit the **Type** node

Figure 14.2 Type Node Settings



There are a variety of fields in the file, 120 in total. Most have the Role set to *None* so they won't be used for modeling, but nine have Role *Input* and will be used as a predictor.

The field to be predicted is *Long distance over tenure*, which is the total number of minutes of long distance service used by this customer while he/she has had a current account (their tenure).

Scroll down until you can see the **Long distance over tenure** field (not shown)

The telecommunications firm would like to predict this quantity to determine how many long distance minutes a customer is likely to use. The target field has values which range from 0.9 to over 13,000 minutes. Predicting long distance minutes will help in both marketing and planning activities.

Partitioning the File

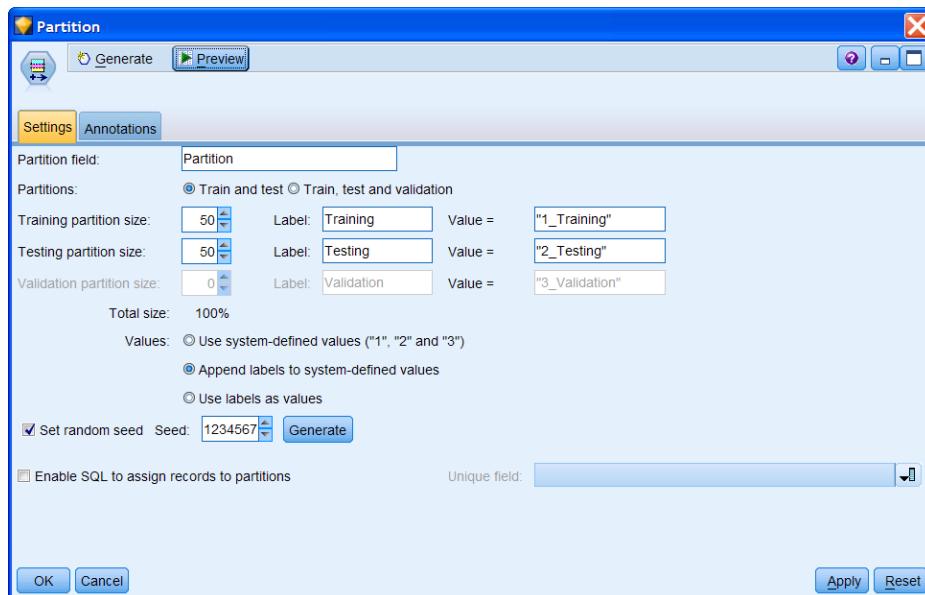
There are 5000 records in this file, and we will add a Partition node to the stream to create training and testing samples.

Close the Type node

Add a **Partition** node from the Field Ops palette to the right of the **Type** node

Connect the **Type** node to the **Partition** node

Edit the **Partition** node

Figure 14.3 Partition Node

We'll use the typical sizes of 70% for the training partition and 30% for the testing partition.

Change the value of the Training partition size to **70**

Change the value of the Testing partition size to **30** (not shown)

Click **OK**

14.3 Using the Auto Numeric

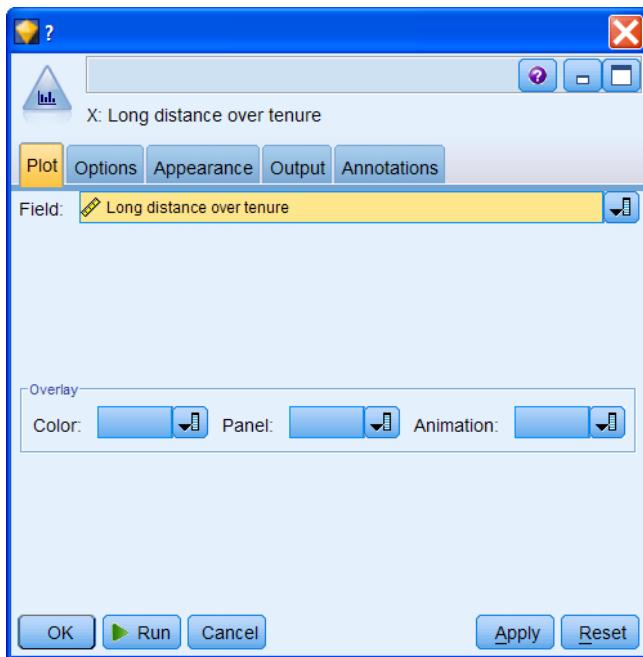
We could immediately model the total number of long distance minutes, but before we do, let's look at the distribution of this field, using a Histogram node.

Add a **Histogram** node from the Graphs palette near the **Partition** node

Connect the **Partition** node to the **Histogram** node

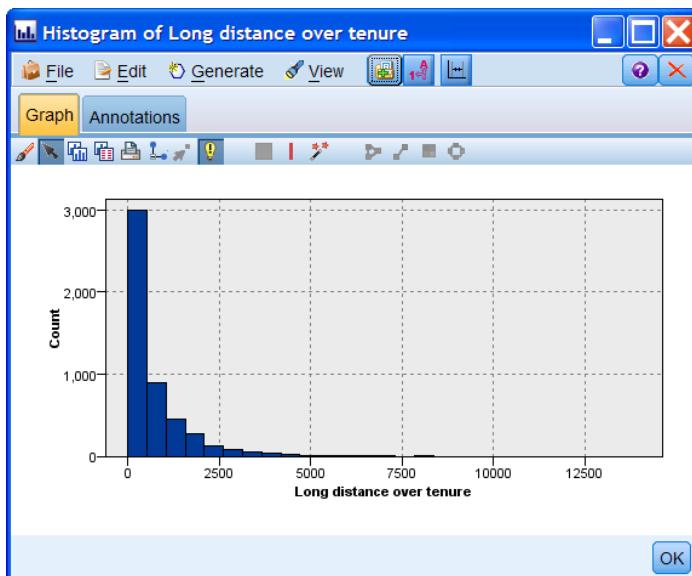
Edit the **Histogram** node

Select **Long distance over tenure** as the Field

Figure 14.4 Requesting a Histogram for Long distance over tenure

Click **Run**

The distribution of *Long distance over tenure* is quite positively skewed. Because of this characteristic, you might try to predict both the original field values and the log of this field (the log would make the distribution more approximately normal). Values which are more normal would work better for regression-based models, for example. In this lesson, we will only attempt to predict the original field.

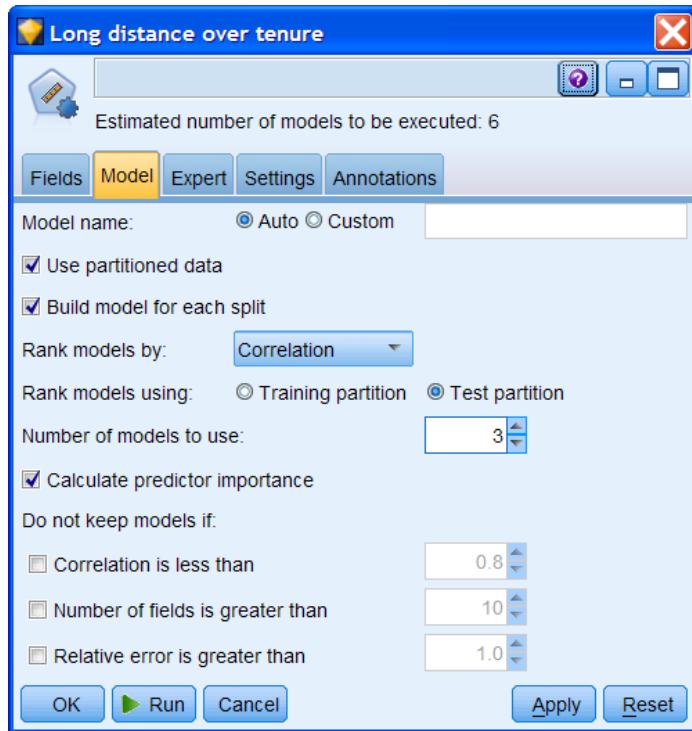
Figure 14.5 Histogram of Long distance over tenure

Close the Histogram window

Add an **Auto Numeric** node from the Modeling palette to the stream to the right of the **Partition** node

Attach the **Partition** node to the **Auto Numeric** node
Edit the **Auto Numeric** node

Figure 14.6 Auto Numeric Node Model Dialog

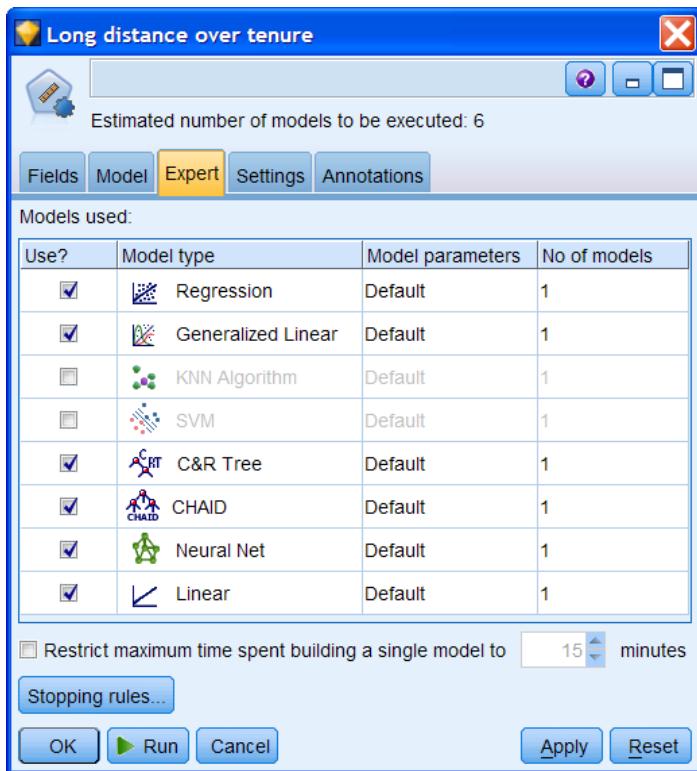


The Auto Numeric node has three choices to compare the models. By default, models are ranked on the correlation between the target and predicted values for the target. Other choices for ranking include the number of fields in the model and the relative error, which is the ratio of the variance of the observed values to those predicted by the model to the variance of the observed values from the mean. The lower the relative error (under 1), the better the model.

Models can be dropped if they don't meet selected criteria—the same three that are used for ranking models.

As with the Auto Classifier, the Auto Numeric builds models on the Training data and tests them on the Test data. The models by default will be ranked on the Test data, but we'll change this. Since many possible models can be developed, the node will keep up to 20 models to summarize and compare.

Click **Training partition** for Rank models using: option
Click the **Expert** tab

Figure 14.7 Auto Numeric Expert Tab

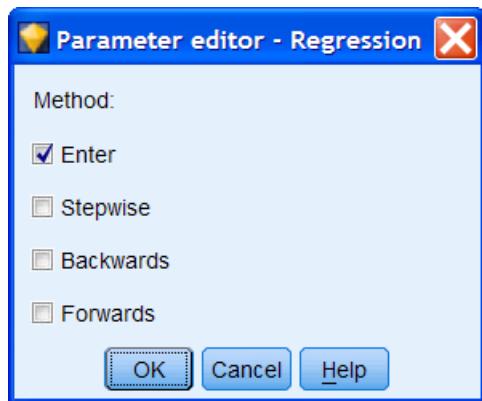
Six different models are available and used by default. KNN is a model introduced in PASW Modeler 13.0. Two were not offered in the Auto Classifier—regression and generalized linear—because they are only used to predict continuous targets. We'll try these six models in this example. The same type of stopping rules and options restrict time to build a single model are available.

Model parameters, such as the depth of a tree or the number of records in parent and child branches, or the entry choice for fields in regression, can be set for each model in the *Model parameters* column by selecting *Specify* from a dropdown list.

There is no option to set misclassification costs because this only applies to categorical targets.

We'll change the regression model parameters to ask for both direct entry and stepwise selection of predictors.

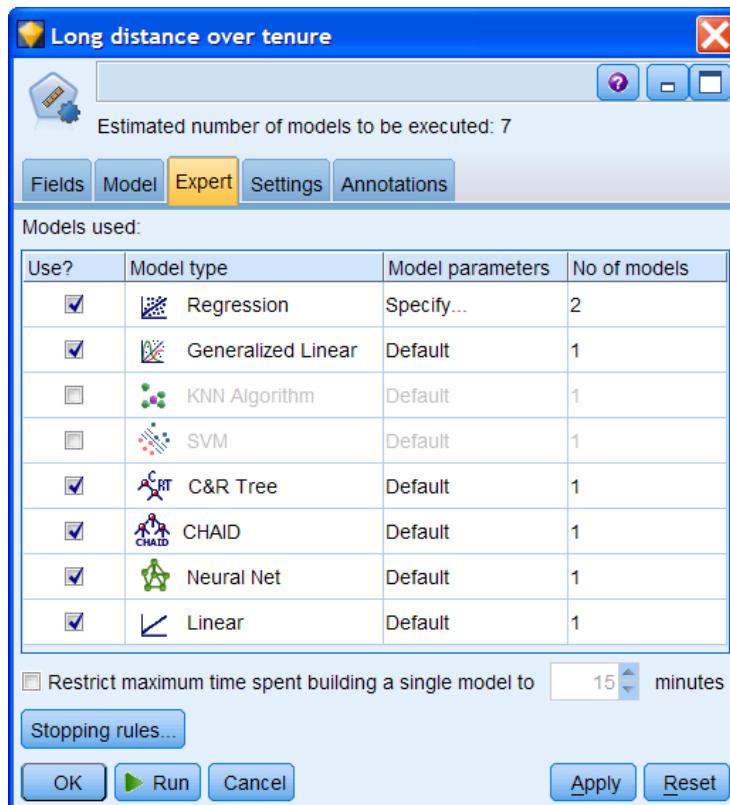
Click in the **Model Parameters** cell for Regression and select **Specify** from the dropdown list
(not shown)
Click in the **Options** cell for the Method parameter and select **Specify**

Figure 14.8 Specifying Regression Methods

There are four types of regressions that can be requested. By default, all the predictors will be entered in one step. The *Stepwise* choice will build a model using the best predictor, then the next best predictor, and so forth until no other potential predictor is significant.

- Click the **Stepwise** checkbox
- Click **OK**
- Click **OK** again

In the Expert tab window, we now see that 2 models will be created with Regression. Seven models will be constructed with the current selections.

Figure 14.9 Two Models Requested Using Regression

Click Run

As with the Auto Classifier, the Auto Numeric node will generate a model nugget in the Models manager palette, and in the meanwhile, the nugget will be connected to the Partition node on the stream canvas.

Double-click the model nugget named **Long distance over tenure** (or right-click the nugget and then select **Edit...** from the context menu)

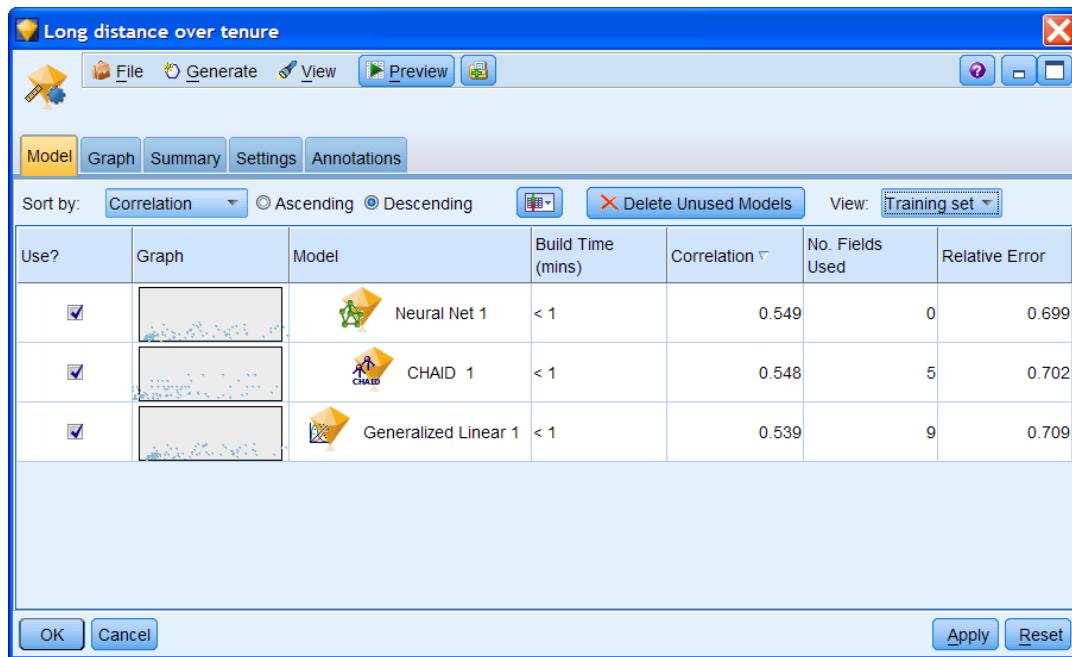
Figure 14.10 Auto Numeric Model Evaluation Report, Testing Set

Use?	Graph	Model	Build Time (mins)	Correlation	No. Fields Used	Relative Error
<input checked="" type="checkbox"/>		CHAID 1	< 1	0.517	5	0.734
<input checked="" type="checkbox"/>		Generalized Linear 1	< 1	0.513	9	0.737
<input checked="" type="checkbox"/>		Neural Net 1	< 1	0.503	0	0.749

The correlation is the correlation between the actual and predicted target values. According to this measure, the best three models are CHAID, GenLin and Neural Net in turn. It is worth noting that the two regression models, Regression and (new in PASW Modeler 14) Linear, are excluded because of the poor performance. As with the Auto Classifier example, all three models have similar correlation, so unless correlation is the only standard, you might choose a winning model based on other criteria, such as relative error.

First let's look at results for the Training set.

Click **Training set** on the View: dropdown

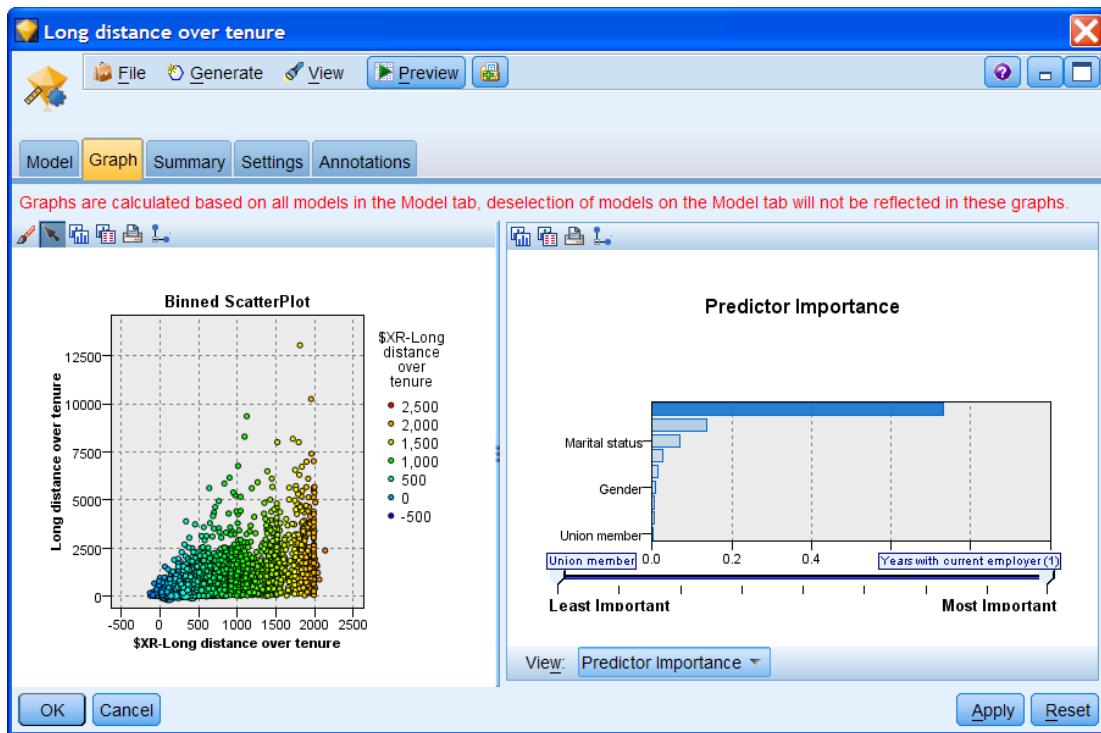
Figure 14.11 Auto Numeric Model Evaluation Report, Training Set

The best model on the Training data is Neural Net, followed by CHAID, and then the GenLin.

Also provided is a thumbnail of a scatterplot showing the actual versus the predicted target values. If the model performs well, the points should fall close to a straight diagonal line from lower left to upper right. The full-sized plot includes up to 1000 records and will be based on a sample if the dataset contains more records than that.

Let's look at the graph for the best models.

Click the **Graph** tab

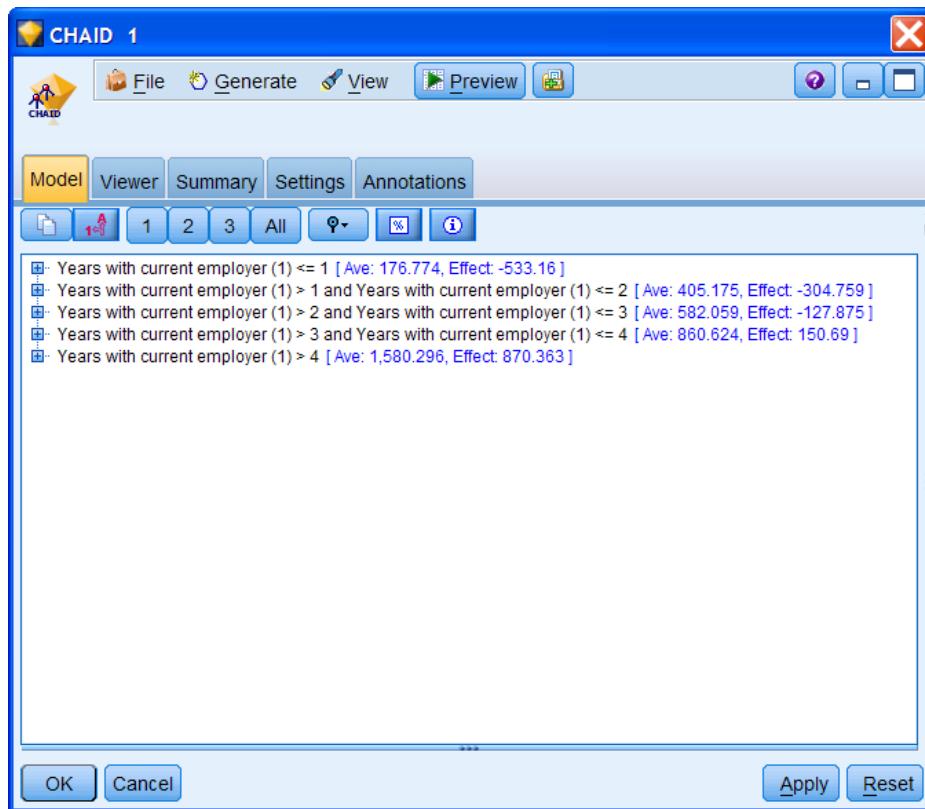
Figure 14.12 Scatterplot of Value of Long distance over tenure

Here, we see a scatter plot and a Predictor Importance plot. From the scatter plot we can see how the models perform in prediction. From the Predictor Importance plot we observe that the most important field is Years with current employer that makes the most contribution in the models.

Because the target field is continuous, evaluation charts cannot be created. We can use the Generate menu to produce modeling nodes to look at the models in more depth. Let's do that for the CHAID model.

Click the **Model** tab
Double Click in the **Model** column for **CHAID 1**

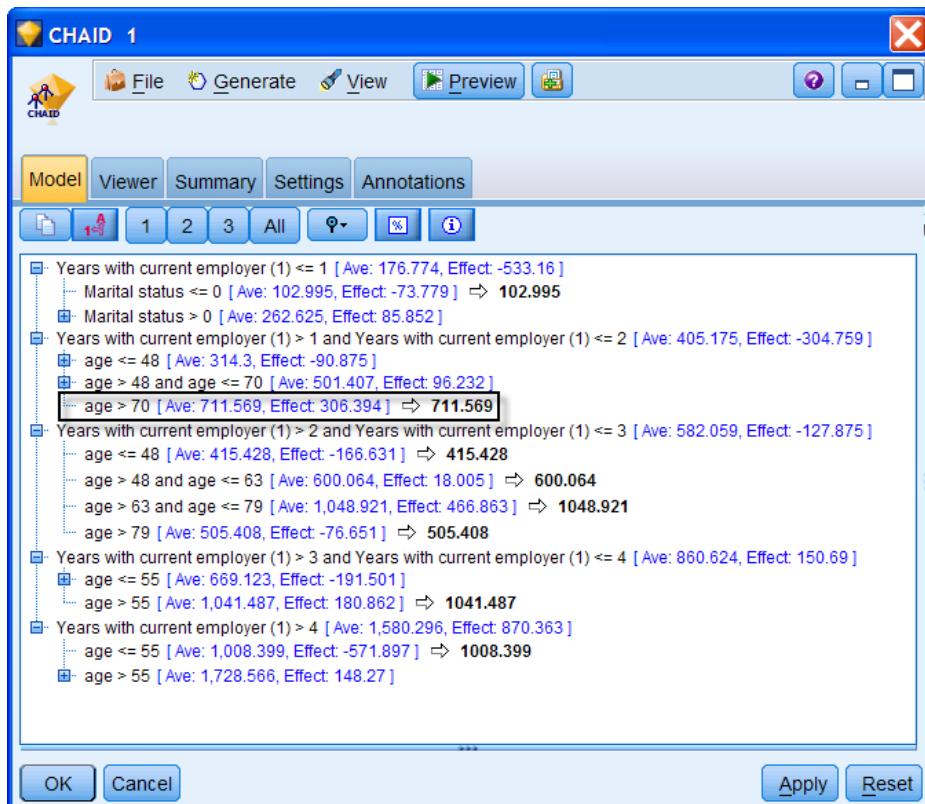
For the CHAID model, the list of rules looks very different than when we used a decision tree to predict a categorical target. For a continuous target, models predict the mean for the target field for a group of records (customers). Also listed is the deviation from the overall average of the parent branch (Effect), which at the top of the tree is equivalent to the overall average for *Long distance over tenure*.

Figure 14.13 CHAID Model Ruleset Top Level

The first split occurs on the field *Years with current employer*. There are no terminal branches shown, which if visible would end with an arrow and a predicted (average) value for that node. To see a terminal node we need to go one level deeper.

Click the **Show level 2** button  on the toolbar

The tree expands by one level. In Figure 14.14 we have outlined one of the terminal branches. In this branch, for customers who have been with their employers for two years and are more than 70 years old, the predicted number of long distance minutes is 711.569.

Figure 14.14 CHAID Model Expanded One Level

This Chaid model can be added to the stream, via **Generate...Model to Palette** (not shown here), e.g. in order to make predictions on new data if we believe that the model is satisfactory.

Summary

In this lesson you have been introduced to the Auto Numeric node. You should now be able to:

- Use the default settings of the Auto Numeric node to predict a continuous target
- Make changes to the model specifications to request additional models
- Use the Auto Numeric Model Results to evaluate model performance and to generate modeling nodes

Exercises

In this exercise, we will use the Auto Numeric node to predict post-campaign visits.

1. Open the *ExerLesson13.str* stream from the previous lesson, or *ExerLesson12.str* from Lesson 12.
2. Review the measurement level for the *Post-campaign visits* field and specify it as a Continuous if it is not already. Change its role to Target. Leave the Role specifications for other fields the same so the predictors will be the same as for the Auto Classifier, but change the role for *Response to campaign* to None.
3. Attach an Auto Numeric node to the Type node. Edit the node to use the Training partition to rank the models and run the node.
4. Review the results for the Training set first. What are the top three models? Now review the results for the Testing set. Are the top three models different than for the Training set?
5. Generate a model nugget on the Models Palette for the best model (in terms of highest correlation on the Testing dataset) and add it to the stream canvas.

Lesson 15: Model Understanding

Objectives

- Use the Analysis node to view model predictions
- Demonstrate how to increase understanding of a model's predictions
 - Use Distribution graphs for categorical fields
 - Use Histograms for continuous fields
 - Use the Means node for continuous fields

Data

Throughout this lesson we will continue using the credit risk data (*Risk.txt*) introduced in the previous lessons.

15.1 Introduction

We developed two models to predict *RISK* in Lesson 12 using C5.0 and CHAID. Both of these nodes created a decision tree model that also provided prediction rules for the categories of *RISK*. The rules are easy to understand and use.

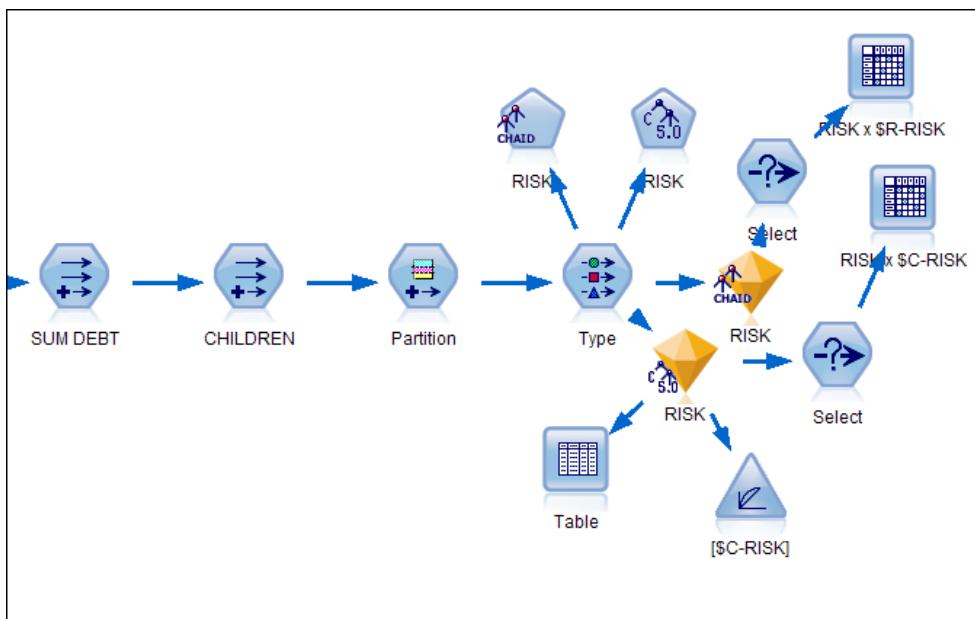
Nevertheless, the rules don't tell the full story of how the model operates. First, except for one rule in the C5.0 model, where *LOANS*=0 leads directly to a prediction (*good risk*), all the other rules involve two or more fields, even in this relatively uncomplicated model. So if we would like to know how predictions are related to, say, the complete distribution of *LOANS*, that can't be read directly from the rules. Second, there are several fields that weren't used by either model, even CHAID, which used more fields than the C5.0 model. You will often want to deepen your understanding of the model by determining how some of these other fields relate to a model's predictions.

We therefore extend our review of a generated model by looking in more depth at the predictions of the C5.0 model in this lesson. We do this to better understand the model and to see if it is acceptable in the way it relates the predictors to the target. We also may determine how the model can be improved, such as by creating new fields.

15.2 Reviewing Model Accuracy with the Analysis Node

We will work with the stream file saved in Lesson 12.

Click **File...Open Stream**, navigate to the **c:\Train\Modeler\Intro** directory
Double-click **Rule Induction.str** (alternatively, use **Backup_Rule Induction.str**)

Figure 15.1 Rule Induction Stream

This stream contains a data source node, a Partition Node splitting the data into Training and Testing Samples, nodes to create two new fields, a Type node with *RISK* specified as the target field, generated C5.0 and CHAID models, plus all the other nodes we used to explore the model. We won't remove any nodes, as all the nodes we use here can be attached to the C5.0 generated model.

When a model seems satisfactory based on performance, fields included, and the relationships between the predictors and the target, the next phase of the CRISP-DM process is model evaluation. Formally, model evaluation is the assessment of how a model performs on unseen data. Since classic data-mining methods, such as decision trees and neural networks, don't test any global hypothesis about a model, with an associated probability value, another method must be used to determine model performance in the target population. The approach, as discussed in Lesson 1, is to apply the model to new, unseen data (unseen during model building) and check its accuracy and performance.

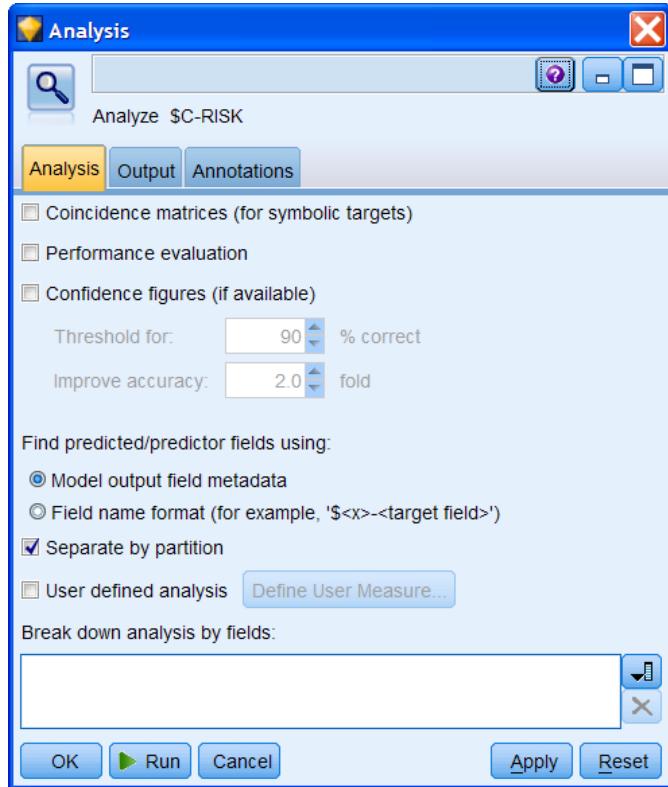
PASW Modeler makes this easy to accomplish by use of the Partition field. We formerly used the Partition node to split the data file into Testing and Training partitions (70% and 30%, respectively, of the full risk data file). In previous lessons we were careful when studying the model not to use the Testing partition (by using a Select node). Doing so would compromise model testing because we would learn how well the model performed on the unseen data. We could then use that information to modify and improve the model. But then we would eventually evaluate the model on the same Testing data we used beforehand. That would be a serious error in model building.

What we will do is add an Analysis node to the stream and attach it to the C5.0 generated model. The Analysis node allows you to evaluate the accuracy of a model, and it organizes output by the partition field values. Analysis nodes perform various comparisons between predicted values and actual values (your target or Out field) for one or more model nuggets, thus allowing you to compare two or more predictive models (see the next lesson). The Analysis node is contained in the Output palette.

When you run an Analysis node, a summary of the analysis results is automatically added to the Analysis section on the Summary tab for each model nugget in the executed stream. The detailed analysis results appear on the Outputs tab of the manager window or can be written directly to a file.

Add an **Analysis** node from the Output palette to the stream near the C5.0 generated model
 Connect the **C5.0 model** to the **Analysis** node
 Edit the **Analysis** node

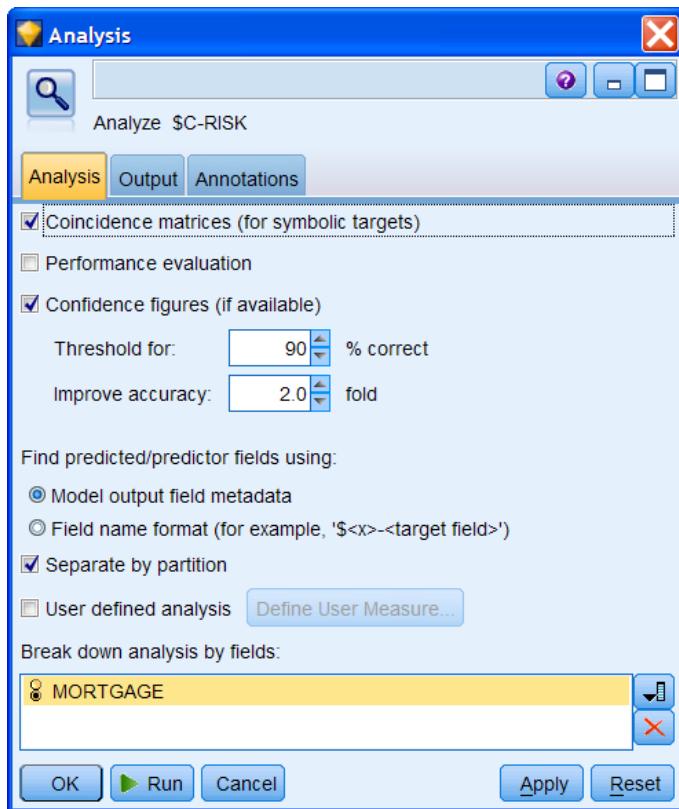
Figure 15.2 Analysis Node Dialog



The Analysis node provides several types of output. *Coincidence matrices (for symbolic targets)* are the equivalent of the crosstabulations we constructed previously with the Matrix node. If a model produced confidence values (which C5.0 does), we can request various measures based on the confidence with the *Confidence figures* check box.

By default, the node will organize output by the partition field. We can also ask that all the output be broken down by one or more categorical fields. In our case, we'll break the results down by **MORTGAGE** (whether the customer has a mortgage).

Click **Coincidence matrices (for symbolic targets)** check box
 Click **Confidence figures (if available)** check box
 Click the field chooser button for **Break down analysis by fields:**
 Select **MORTGAGE** as the field, and then click **OK**

Figure 15.3 Completed Analysis Node Selections

Click **Run**

Many tables are generated in the Analysis Output Browser window, so we will look at them in sections.

The first set of tables shows the overall accuracy of the model in the Training and Testing partitions. When we were examining the C5.0 model, we saw that its accuracy varied widely among the three categories of *RISK*. The model best predicted the *bad profit* group and was worse for the *bad loss* category. But we never calculated overall model accuracy. This was in part because the Matrix node doesn't provide this figure, but also because when predicting a categorical target with three or more categories, overall accuracy may be less critical than accuracy for one or two of the categories. For the risk data and the decision of making a loan to a customer, the most important target might be the *bad loss* group, since that is where money can be lost. So accuracy in other categories might be sacrificed to improve accuracy for this group.

As we can see from the output in Figure 15.4, the accuracy of the C5.0 model on the Training data is 76%. Even when we are interested in overall accuracy, whether this level of accuracy is acceptable will depend on many factors. Accuracy cannot be judged in the abstract.

What is of great interest is how well the model performs on the unseen Testing partition data. The accuracy of the C5.0 model for the Testing partition is the fundamental overall test of the model. If the accuracy on the Testing data is acceptable, then we can deem the model validated.

For these data, the Testing partition accuracy is a (reassuring) 74.9%. The typical outcome when testing data are passed through a model node is that the accuracy drops by some amount. If accuracy

drops by a large amount, it suggests that the model overfit the training data or that the validation data differ in some systematic way from the training data (although the random sampling done by the Partition node minimizes the chance of this). If accuracy drops by only a small amount, it provides evidence that the model will work well in the future.

The small drop of 1.1% in accuracy for the Testing data indicates that the C5.0 model is validated

Figure 15.4 Analysis Node Accuracy Output

The screenshot shows the Analysis Node Accuracy Output window. At the top are 'Collapse All' and 'Expand All' buttons. Below is a tree view with nodes: 'Results for output field RISK', 'Overall Results', 'Comparing \$C-RISK with RISK', and 'Coincidence Matrix for \$C-RISK (rows show actuals)'. The 'Comparing \$C-RISK with RISK' node contains a table:

'Partition'	1_Training	2_Testing	
Correct	2,163	76%	952
Wrong	683	24%	319
Total	2,846		1,271

The 'Coincidence Matrix for \$C-RISK' node contains two tables:

'Partition' = 1_Training		bad loss	bad profit	good risk
bad loss		217	301	84
bad profit		23	1,559	91
good risk		7	177	387

'Partition' = 2_Testing		bad loss	bad profit	good risk
bad loss		116	152	36
bad profit		10	678	46
good risk		3	72	158

Two points to keep in mind:

1. You can explore model predictions in the testing data just as we did in the training data in Lesson 12. However, if the partitions have been created randomly, there is no reason to expect the model to operate differently in the Test partition.
2. When reporting model accuracy, the value from the Test partition should be used. This is the expected performance of the model on new data.

The Coincidence Matrix table will be of special interest when there are target categories in which we hope to make accurate predictions. Recall that the accuracy of predicting *bad loss* in the Training data was about 36%. Although it is not calculated in the output, we can calculate with no difficulty that the accuracy for this group in the Testing partition is $(116 / (116 + 152 + 36)) * 100 = 38.2\%$. This is slightly higher than for the Training partition (thus validating this set of predictions), but still rather low and probably not acceptable for the lending institution.

We can also readily observe that the accuracy in the Testing partition is very high for the *bad profit* group and of moderate accuracy for the *good risk* group, just as in the Training data.

The model predictions can be evaluated in two directions, so to speak, and in the second direction, the accuracy at predicting the *bad loss* group is surprisingly high. If we look at the column for *bad loss* in the Testing partition, note that there are $116 + 10 + 3 = 129$ customers predicted to be a *bad loss*. Of this group, $(116/129) * 100 = 89.9\%$ were correctly predicted. This does not somehow contradict the discussion above. What it does mean is that *when* the model predicts that someone is going to be a *bad loss*, the model is almost always correct. The problem is that the model can't predict (find) all the customers in this group.

Next are two tables, for each partition, reporting on various measures based on the confidence values of the model's prediction. As a reminder, for C5.0 models, and decision trees in general, confidence is calculated in a single node as the percentage of cases in a category of interest. Overall accuracy is then a weighted average of these percentages.

Of special interest are the confidence values when a model makes a correct, or incorrect, prediction. In the Training partition, the mean confidence for correct predictions (.799) is substantially higher than for incorrect predictions (.627). This is the type of difference that we look for, and, no pun intended, it increases our confidence in the model.

In the Testing partition, the mean confidence values are very similar, which we also want to see for a successful model.

Figure 15.5 Confidence Values Report

Confidence Values Report for \$CC-RISK	
'Partition' = 1_Training	
Range	0.447 - 0.892
Mean Correct	0.799
Mean Incorrect	0.627
Always Correct Above	0.892 (0% of cases)
Always Incorrect Below	0.447 (0% of cases)
90% Accuracy Above	Never reached requested level
2.0 Fold Correct Above	0.891 (71.12% of cases)
'Partition' = 2_Testing	
Range	0.447 - 0.892
Mean Correct	0.792
Mean Incorrect	0.624
Always Correct Above	0.892 (0% of cases)
Always Incorrect Below	0.447 (0% of cases)
90% Accuracy Above	Never reached requested level
2.0 Fold Correct Above	0.887 (71.12% of cases)

Scrolling down, we then see the breakdown by the two categories of *MORTGAGE*.

Figure 15.6 Analysis Node Output for Customers without a Mortgage

Output field RISK, splitting by field MORTGAGE	
MORTGAGE = n	
Comparing \$C-RISK with RISK	
'Partition'	1_Training 2_Testing
Correct	427 66.72%
Wrong	213 33.28%
Total	640 277
Coincidence Matrix for \$C-RISK (rows show actuals)	
'Partition' = 1_Training	bad profit
bad loss	135
bad profit	427
good risk	78
'Partition' = 2_Testing	bad profit
bad loss	57
bad profit	193
good risk	27

The model accuracy is much lower for those without a mortgage, 69.68% in the Testing partition. Intriguingly, the Coincidence matrices look odd because there is only one column in the table. This is because the C5.0 model only predicts *bad profit* if a customer doesn't have a mortgage. This was certainly not apparent from our review of the model rules in Lesson 12 since the field *MORTGAGE* wasn't even used in the tree!

This example illustrates the importance of examining a model in more detail, even a decision tree that seemingly provides clear rules that illustrate how its predictions are made. Associations between the predictors can cause unexpected relationships between a predictor and the model outcomes, as here for *MORTGAGE*.

The model accuracy improves quite a bit for customers who hold a mortgage, as seen in Figure 15.7. Practically speaking, this variation in model performance could lead to more caution when using the model for customers who don't have a mortgage (or more confidence for those who do). That is a practical outcome from this type of analysis.

Figure 15.7 Analysis Node Output for Customers with a Mortgage

The screenshot shows the SPSS Modeler interface with the following data:

		'Partition'		1_Training	2_Testing
Correct		1,736	78.69%	759	76.36%
Wrong		470	21.31%	235	23.64%
Total		2,206		994	

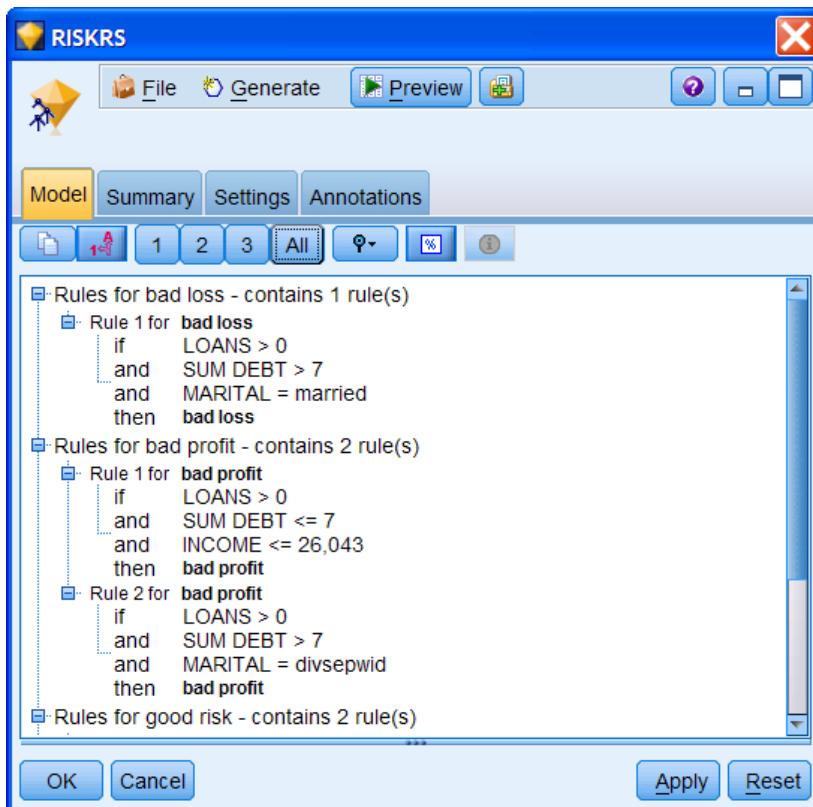
'Partition' = 1_Training		bad loss	bad profit	good risk
bad loss	217	166	84	
bad profit	23	1,132	91	
good risk	7	99	387	

'Partition' = 2_Testing		bad loss	bad profit	good risk
bad loss	116	95	36	
bad profit	10	485	46	
good risk	3	45	158	

15.3 Model Predictions for Categorical Fields

Only four fields of the twelve possible were used by the C5.0 model: *LOANS*, *SUM DEBT*, *INCOME*, and *MARITAL*. The first three are *Continuous*, the last *Nominal*. The measurement level of each field will determine what techniques we use to further investigate model predictions.

As a reminder, Figure 15.8 shows the model predictions in ruleset format.

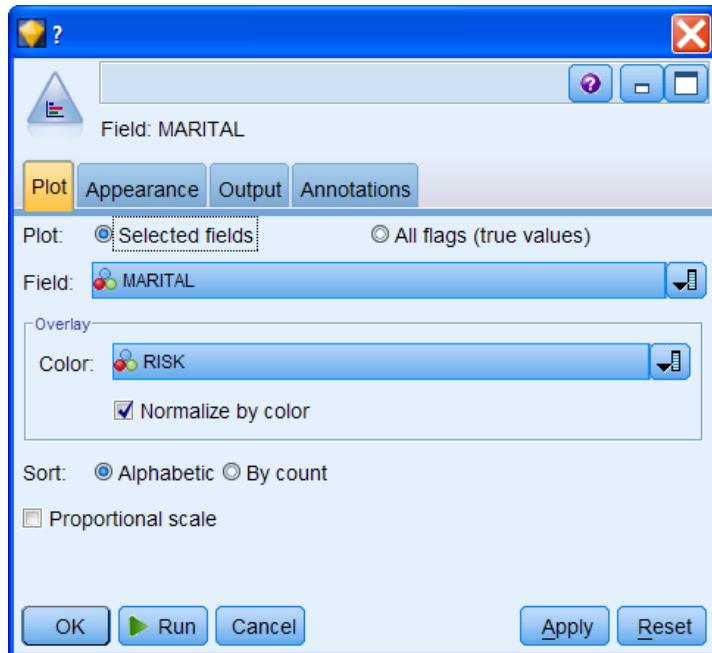
Figure 15.8 Rules From C5.0 Model to Predict RISK

The two rules using categories of *MARITAL* are highlighted. Notice that people who are single do not have a rule, although they do have a prediction. This is because the full tree included a default terminal node for the *single* category where there are no single customers in that branch in training partition (the prediction is *bad loss*). But since there *are* indeed no cases in the training data that match this rule, what will the model predict for this category? Or to put it another way, how are people who are customers who are single related to the model's predictions?

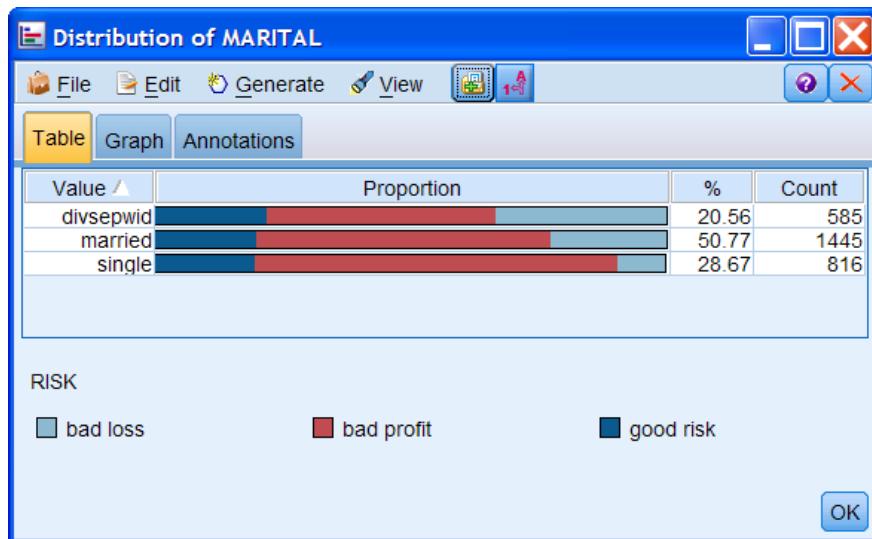
Further, even for the other two categories of *MARITAL*, it isn't at all obvious how they will relate in general to the model's predictions. Married customers are used in only one rule, and even then only when they have one or more loans and a value of *SUM DEBT > 7* (the prediction is *bad loss*). But not everyone who is married is going to have the same value on *RISK* or on *\$C-RISK*.

The Distribution node, which we have used in previous lessons, is a direct way to answer these questions. We will begin with a Distribution graph—a stacked bar chart—of the target field *RISK* with *MARITAL*. This is a good practice to provide a baseline for comparison to the model predictions. Normalizing by color makes it easy to compare the distributions in each category.

- Add a **Distribution** node to the stream from the Graphs palette and attach it to the **Select** node downstream from the C5.0 model
- Edit the Distribution node
- Select **MARITAL** as the Field
- Select **RISK** as the Overlay field
- Click **Normalize by color** check box

Figure 15.9 Distribution Node Request for RISK by MARITAL

Click **Run**

Figure 15.10 Distribution Graph of RISK by MARITAL

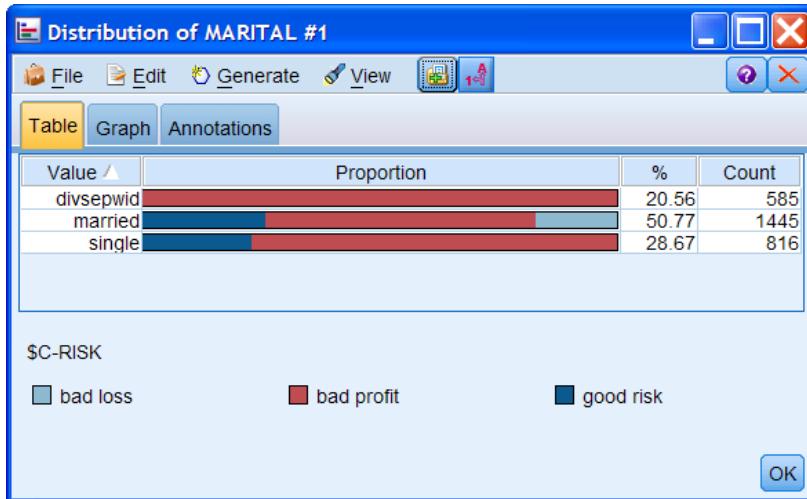
We observe that a majority of single people are in the *bad profit* group, while there is a somewhat equal split across all three categories of *RISK* for those divorced, separated, or widowed. Not surprisingly, all three categories of *RISK* are found in each category of *MARITAL*.

Now we create a similar chart using the model's predictions.

- Close the Distribution window
- Edit the Distribution node in the stream
- Replace **RISK** with **\$C-RISK** in the Overlay field

Click Run

Figure 15.11 Distribution Graph of Predicted RISK by MARITAL



The results are very different than we observed for *RISK*. Before, all three categories of *RISK* were about equally represented in the *divsepwid* group. Now, *everyone* in this category is predicted to be *bad profit*. The distribution for the *single* category is closer to that for *RISK*, but no one who is single is predicted to be a *bad loss* (recall that the node making such a prediction has no cases in the current tree). Meanwhile, all three risk categories are represented in similar proportions for the married customers for *\$C-RISK*.

These predictions of the model make it seem rather odd, especially for the *divsepwid* category. No model can make accurate predictions if it expects that everyone in this group will be in only one category of *RISK* (and recall what we saw above with the Analysis node for customer's without a mortgage). But, of course, the C5.0 model is far from perfect, as it had very low accuracy for the *bad loss* group and only moderate accuracy for the *good risk* group. Models simplify the world to make predictions, which is both their strength and their disadvantage. The C5.0 model contains only five rules and uses only four fields, which makes it very easy to understand. On the other hand, the price paid is that the model has a wide variation in accuracy across categories and makes some predictions that we might be reluctant to apply (if you are divorced, separated, or widowed, you will be a *bad loss*).

In fact, this absolute relationship comes about in this data file because, even more peculiarly, everyone in the *divsepwid* group has 2 or more loans. (How could you check this statement?). This means that the rule "IF *LOANS* = 0, then *good risk*" is never applied to anyone in this category. And in a comparable situation to marital status, all customers without a mortgage have at least one (other) loan, and therefore the simple rule for those with no loans doesn't apply.

This pattern may seem unlikely, but this situation also allows us to emphasize another key aspect of the CRISP-DM methodology. Throughout the data mining process, you should be continually checking on data quality and looking for characteristics of the data that demand further checking or that suggest that the sample may not be fully representative of the target population.

The fact that no one who is divorced, separated, or widowed has less than 2 loans could have been discovered during the data understanding or the data preparation phases of the data-mining project. But unless you are very thorough, it is very difficult to catch everything, especially in real-world

examples with hundreds of fields and very large samples. That is why you keep looking at the data, and a model's predictions, not just for understanding, but also to check for anomalies that bear further study.

These findings do allow us to make another point about the data mining process and field definitions.

Field Definitions: What is a Loan?

One of the important components of data mining is the definition of the fields used in the project. The fields must be understood, and they must be defined in a way that will be useful to both the analyst and the end user. In this instance, the financial institution has measured the number of outstanding loans for a customer, but they have also recorded in a separate field whether that customer has a mortgage on his/her home. Now many of us might consider a mortgage to be a type of loan, and therefore anyone with a mortgage in the data must have at least one loan. But intriguingly, that is not true.

To see the relationship between *LOANS* and *MORTGAGE*, we can't immediately use a Distribution graph or Matrix table because *LOANS*, although it has only a few values, has measurement level continuous. There are at least two ways to check the relationship, but an easy one is to use the Data Audit node.

Add a **Data Audit** node to the stream from the Output palette and attach it to the **Select** node downstream from the C5.0 model

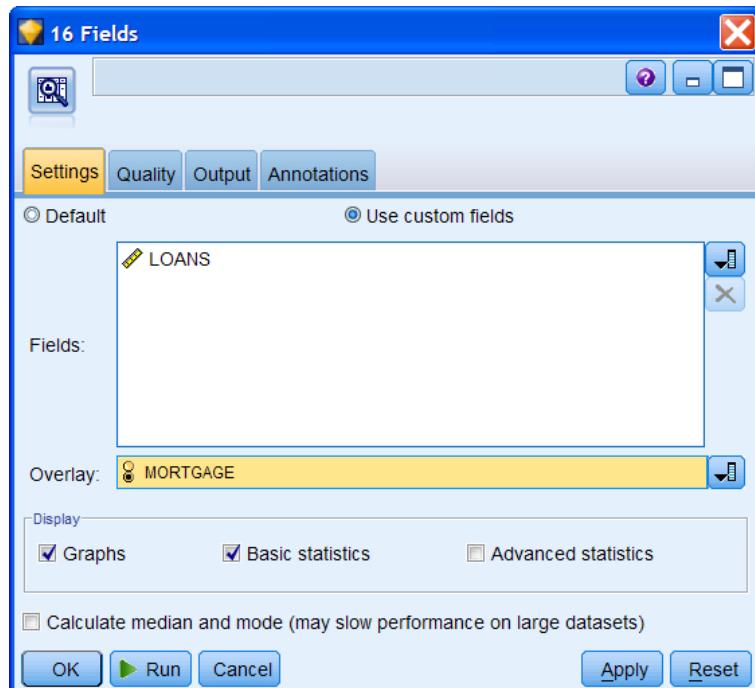
Edit the Data Audit node

Click **Use custom fields**

Select **LOANS** as the Field

Select **MORTGAGE** as the Overlay field

Figure 15.12 Data Audit Node for LOANS by MORTGAGE

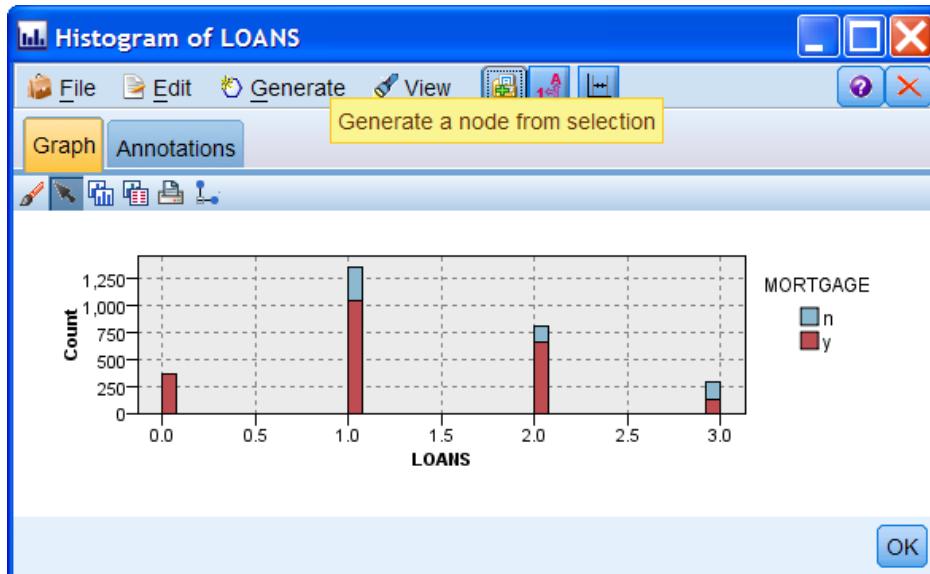


Click **Run**

In the Data Audit Output Browser window,

Double-click on the **Sample Graph for LOANS**

Figure 15.13 Histogram of LOANS with MORTGAGE Overlay



Now we notice something interesting. Everyone with zero loans has a mortgage! This implies that, at least for the risk analysts at the financial institution who prepared this data, a mortgage is *not* equal to a loan. This also implies that this sample of customers is comprised of those who have at least one outstanding loan/mortgage.

We could, though, suggest that a loan and a mortgage be treated as equivalent, so that everyone with a mortgage would have at least one loan. We could, in other words, create another version of the *LOANS* field. Creating alternative versions of fields is a common task in data preparation, but as in this example, sometimes it occurs in the modeling phase of CRISP-DM. You should never be reluctant to do so since this may improve model performance.

Note

Although we won't create the modified version of *LOANS* and rerun the model, if we did so, the model would become more complicated. The split on *LOANS* would not be at 1 loan (the equivalent of the current split at 0), but instead at 2 loans.

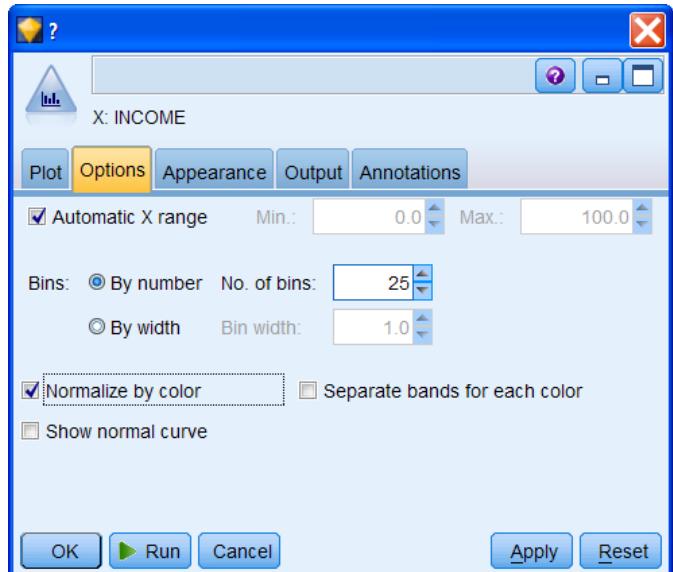
15.4 Model Predictions for Continuous Fields

We turn next to complete an analogous examination of how the model predictions are related to continuous fields. We will focus on *INCOME* and examine its relationship with an overlay histogram (which is what the Data Audit node produced for *LOANS* and *MORTGAGE* in Figure 15.13).

- Close the Histogram Browser window, and then close the Data Audit Browser window
- Place a **Histogram** node from the Graphs palette in the stream and attach it to the **Select** node
- Edit the **Histogram** node
- Select **INCOME** as the Field
- Select **RISK** as the Overlay field

Click **Options** tab
 Click **Normalize by color** check box

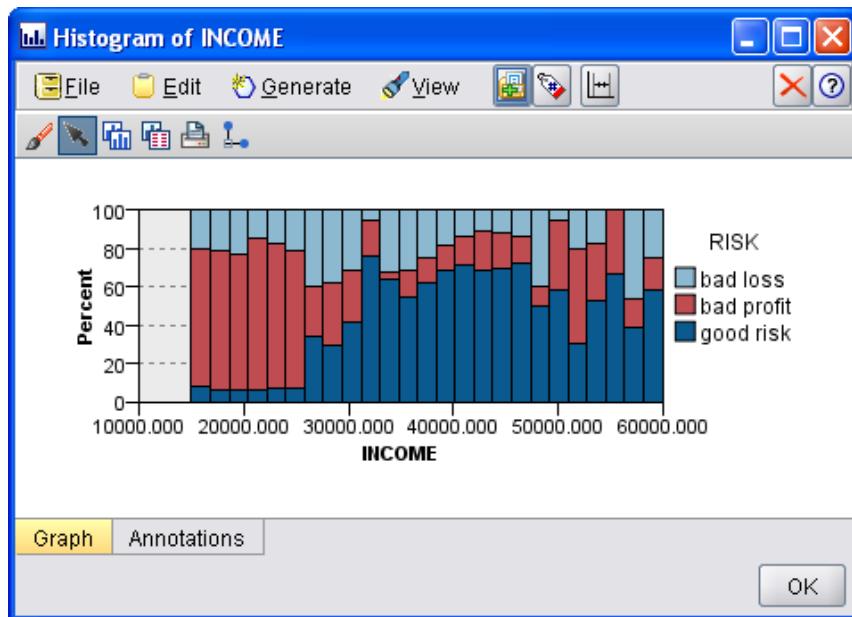
Figure 15.14 Selecting Normalize by Color Option for Histogram



We start by examining the relationship between the actual target field and income. Using *Normalize by color* will have the same effect as with a Distribution graph, making it easy to compare the proportions of each category of *RISK* within income ranges.

Click **Run**

Figure 15.15 Distribution of RISK by INCOME

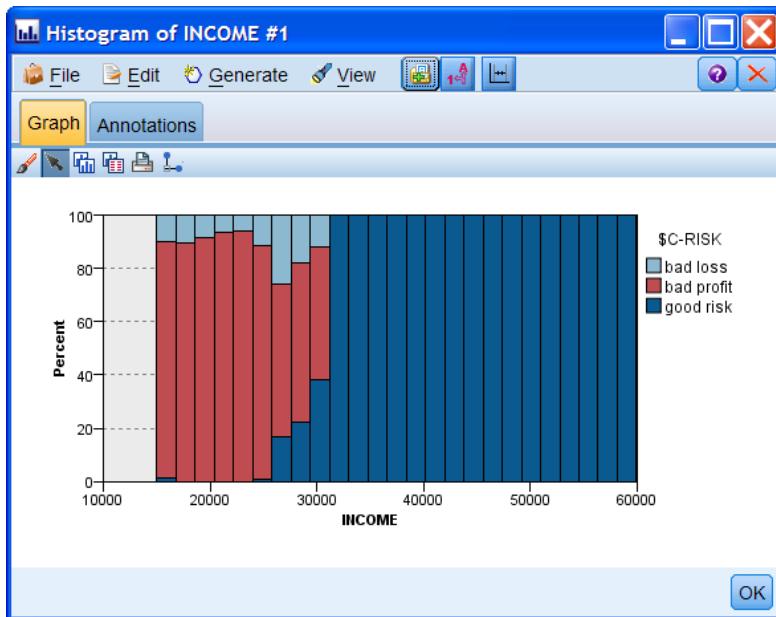


We observe that there are customers in the *bad loss* category throughout the income distribution. Customers in the *bad profit* category tend to have lower income, while customers in the *good risk* category tend to have higher incomes, on average.

Now we do the same for predicted *RISK*.

- Close Histogram Browser window
- Edit the **Histogram** node
- Click **Plot** tab
- Select **\$C-RISK** as the Overlay field
- Click **Run**

Figure 15.16 Distribution of Predicted RISK by INCOME

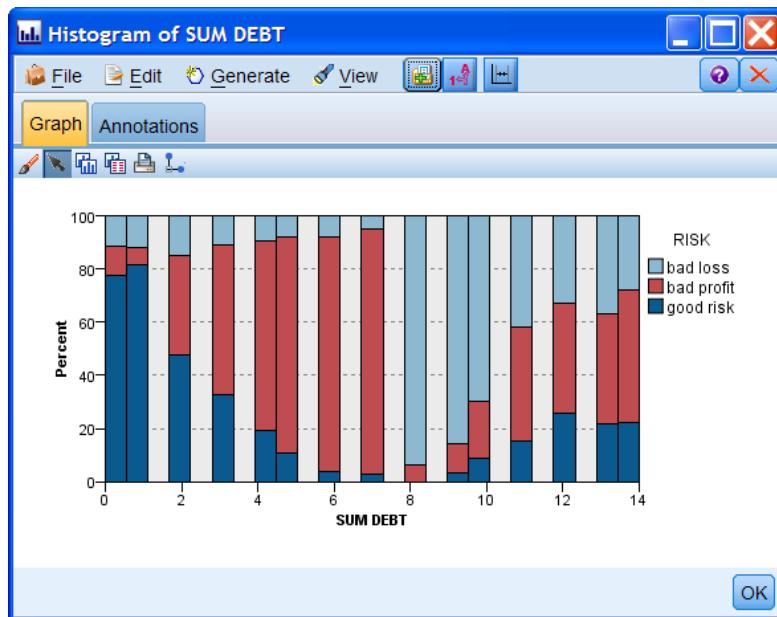


As with categorical fields, the C5.0 model has simplified the complexity of the data. All those predicted to be a *bad loss* or *bad profit* have incomes less than about 30,000, while all those predicted to be *good risk* have incomes above about 26,000.

Although this example may not be entirely realistic because of the sample characteristics we discovered, most models will at least somewhat simplify the relationship between the predictors and the target field.

We repeat this exercise for *SUM DEBT*.

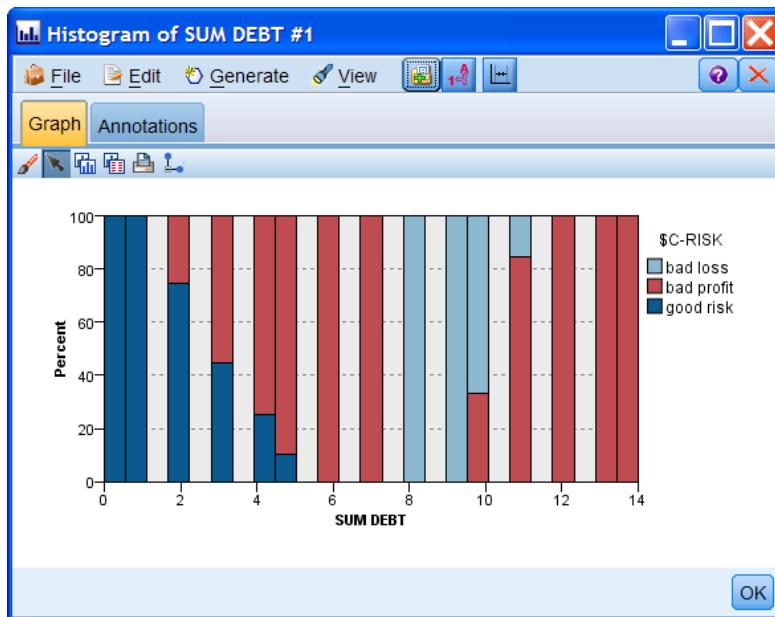
- Close Histogram Browser window
- Edit the **Histogram** node
- Select **SUM DEBT** as the Field
- Select **RISK** as the Overlay field
- Click **Run**

Figure 15.17 Distribution of RISK by SUM DEBT

All three categories of *RISK* are represented across the full range of *SUM DEBT*. There is an interesting break at *SUM DEBT* = 7, where there is a drop in the proportion of *bad profit* customers and a sudden increase in *bad loss* customers. That seems to be a natural value for the model to use for a rule.

Now we create the overlay histogram with predicted *RISK*.

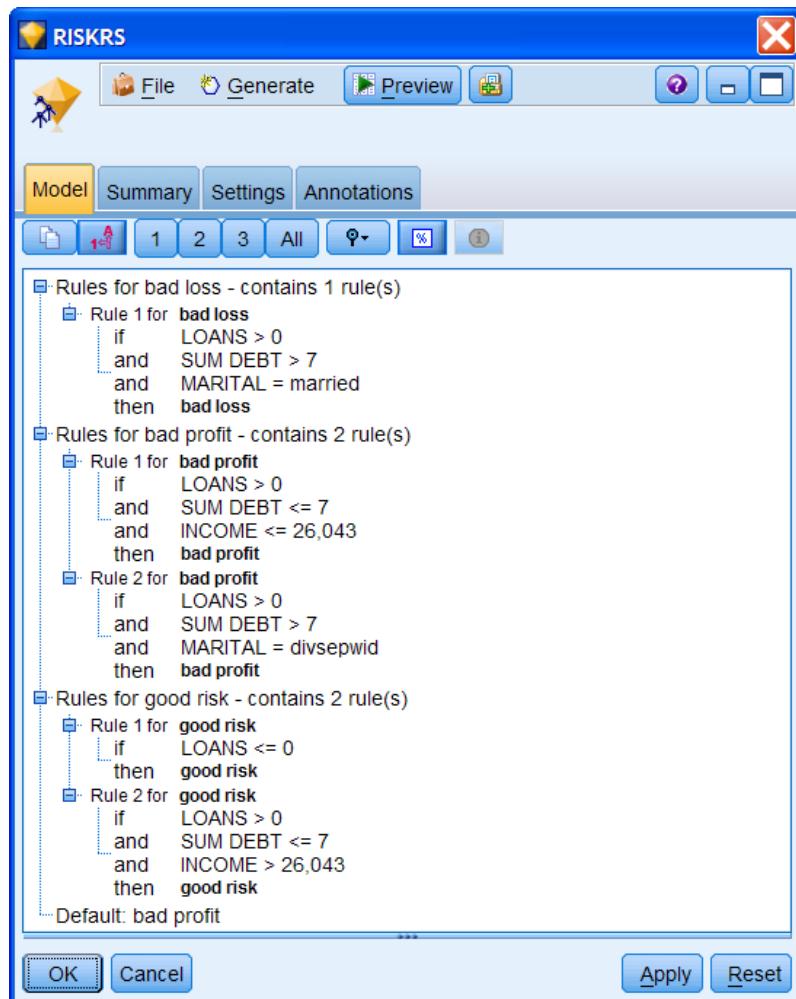
- Close Histogram Browser window
- Edit the **Histogram** node
- Select **\$C-RISK** as the Overlay field
- Click **Run**

Figure 15.18 Distribution of Predicted RISK by SUM DEBT

The histogram is less complex than the actual data. Predictions of *bad profit* continue to come from almost the full range of *SUM DEBT*. But customers predicted to be *good risk* have *SUM DEBT* values of 5 or less, and customers predicted to be *bad loss* have values from 8 to 11. To compare this to the model's rules, we reproduce in Figure 15.19 the ruleset for the C5.0 model.

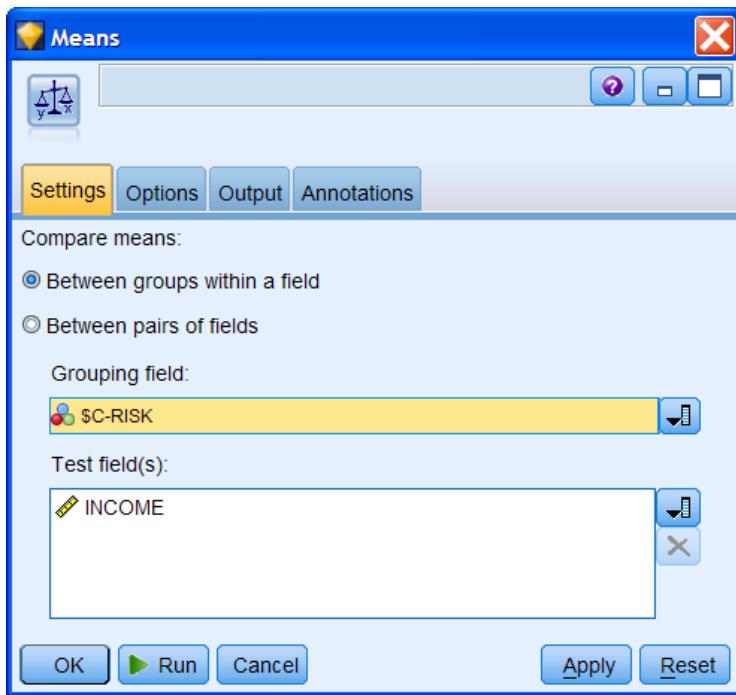
What is quite intriguing is that four of the five rules use *SUM DEBT*, and in every instance the value of 7 is used in the rule. This is a somewhat uncommon case in which by examining the bivariate relationship between a predictor and the target field, we could anticipate the structure of the model.

However, there are still nuances to observe. We don't seem to have any customers with a *SUM DEBT* value of 6 or 7 who are predicted to be *good risk*, yet Rule 2 for *good risk* allows for this possibility. The other fields included in this rule complicate any straightforward relationship between *SUM DEBT* and the *good risk* category.

Figure 15.19 C5.0 Model Ruleset

As a last technique for model understanding, we will look at the mean of *INCOME* by predicted values of *RISK*. This is a quick way to summarize the relationship between a categorical and continuous field in a table.

- Close the Histogram Browser window
- Add a **Means** node from the Output palette to the stream and connect it to the **Select** node
- Edit the **Means** node
- Select **INCOME** as the Test field
- Select **\$C-RISK** as the Grouping field

Figure 15.20 Means Dialog Selections

Click Run

The mean income for those predicted to be *good risk* is far higher (40,091) than for either of the other two categories. Of course, this is commonsensical—you are a better risk for a loan if you have a higher income—but it is worthwhile to see the magnitude of the predicted mean difference between the categories. Compare to the overlay histogram in Figure 15.16.

Figure 15.21 Mean of INCOME by Predicted RISK

Means of [\$C-RISK][INCOME]				
File Edit				
Means Annotations				
Sort by: Field View: Simple				
Grouping field:	\$C-RISK			
*Cells contain:	Mean			
Field	bad loss*	bad profit*	good risk*	Importance
INCOME	23192.085	21805.479	40091.765	1.000 Important

Although the mean for *bad loss* is somewhat greater than for *bad profit*, the difference is small, and it would seem that income is not as critical at distinguishing between these two categories. Look back at Figure 15.19 with the rules for the model. Would you have reached this conclusion from examining the rules for these two categories?

Model Understanding Reconsidered

The examples in this lesson have demonstrated how to more fully understand a model that you are considering as either a possible final model, or to see how a model might be improved for a model that you know is underperforming.

We have kept this example reasonably uncomplicated in the interests of exposition, but in a standard data-mining project these tasks will be more time consuming and complex. You won't necessarily have time to look at all the predictors, or other fields, but instead should focus on an important subset.

Click **File...Close Stream**

Click **No** to save the changes

Summary

In this lesson you have been introduced to some ways of further understanding a model.

You should now be able to:

- Use the Analysis node to look at model performance in the Training and Testing partitions
- Use the Distribution node to look at model predictions for categorical predictors
- Use the Histogram node to look at model predictions for continuous predictors
- Use the Means node to also look at model predictions for continuous predictors

Exercises

In this exercise, we will use other nodes in PASW Modeler to help us understand and evaluate the C5 and CHAID models.

1. Open the *ExerLesson12.str*. In both models (C5.0 and Chaid), *Pre-Campaign Expenditure*, *Pre-Campaign Visits Category*, and *Age* are the best predictors of *Response to Campaign*. So, we will further investigate their relationships.
2. Connect a Distribution node to the right of the Select node attached to the C5.0 model node. Request a normalized by color distribution graph for *Pre-Campaign Visits Category* with *Response to campaign* as the overlay field. Rerun the Distribution node using the predicted response, *\$C-Response to campaign*, as the overlay field. Which categories of visits were most under-predicted? Is the "No Visits" group one that you are trying to predict or might you delete that group?
3. Connect a Histogram node and request normalized histograms of *Pre-Campaign Expenditure* with *Response*, and then *predicted response*. Which levels of expenditure over predict "Responders" and which are under predicted? Is this a similar pattern as with Pre-campaign visits?
4. Finally, use the Means node to look at the means of *Pre-campaign Expenditure* for *Predicted Response*. Be sure to connect the Means node after the C5 Model and Select nodes. Do you expect the mean to be higher for Predicted Responders than actual Responders?
5. *For those with more time:* use the Histogram node and Means node to investigate the relationship of *Age* to both the actual response to campaign and predicted response.
6. Save the stream as *ExerLesson15.str*.

Lesson 16: Comparing and Combining Models

Objectives

- Compare Models with the Analysis Node and Evaluation charts
- Combine Models with the Ensemble Node

Data

Throughout this lesson we will continue using the credit risk data (*Risk.txt*) introduced in previous lessons.

16.1 Introduction

At some point the development of a model will be complete. You will have tried various settings for the modeling algorithms to improve performance, added or subtracted fields, and created new fields for the model. And you will have used the techniques from Lesson 15 to more fully understand a model's predictions.

Given that in data mining it is very easy to develop several models, that this is encouraged in the CRISP-DM methodology, and that PASW Modeler provides the Auto Classifier and Auto Numeric nodes to automate model production, it is also likely that you may have more than one candidate model that performs at acceptable levels. If so, you will want to compare one model to the other to allow a decision to be made as to which is the superior model. Or, again in the spirit of data mining, you may wish to combine the models to make predictions on the presumption that two heads (models) are better than one.

In this lesson we will use the results from the Auto Classifier in Lesson 13 to compare the C5.0 and C&R Tree models developed there. Those two were the best in predicting the recoded *RISK* field (comparing *good risk* customers to the other two categories). We will use the Analysis node, introduced in the previous lesson, an evaluation chart, and other methods of model comparison. Then we use the Ensemble node to automate the combining of two or more models.

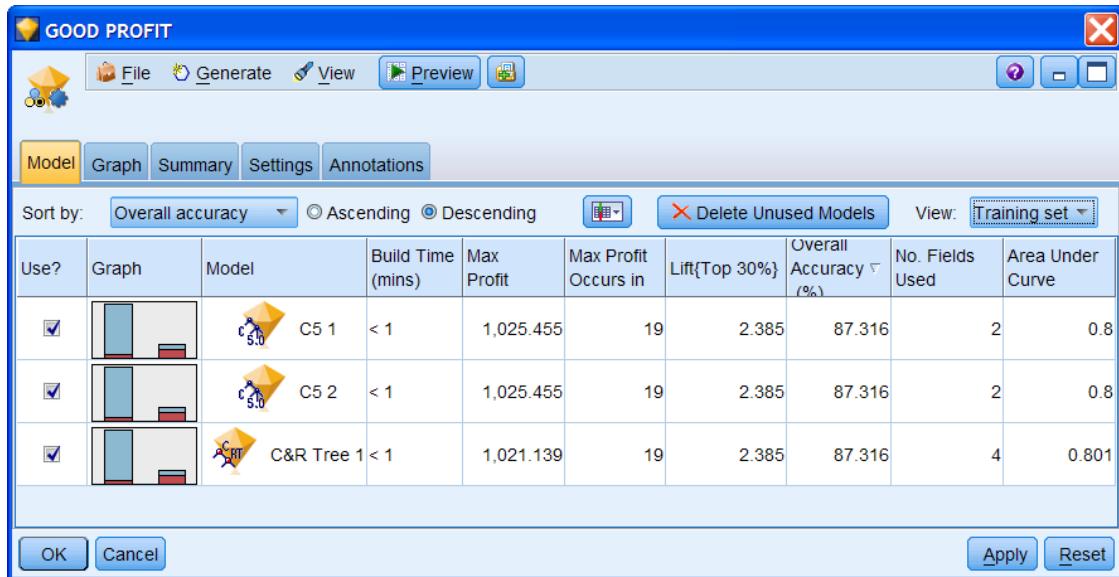
16.2 Comparing Models with the Analysis Node

To compare models, they need to be within the same stream and connected to the same data source node. The stream *Auto Classifier Models.str* from Lesson 13 contains the Auto Classifier node that will allow us to generate model nuggets.

Click **File...Open Stream**
Browse to the c:\Train\Modeler\Intro directory
Double-click **Auto Classifier Models.str**

We need to run the Auto Classifier node to open the browser window from which we can generate the models.

Run the Auto Classifier node labeled **GOOD PROFIT**
Double-click the nugget named **GOOD PROFIT** on the stream canvas
Click the **View** dropdown and select **Training set**

Figure 16.1 Auto Classifier Results to Generate Model Nodes

We can now use the Generate menu to generate model nodes for these two models.

Double-click on the Model column for the **C&R Tree 1** model

In the popup window, Click **Generate...Model to Palette**

Close the **C&R Tree 1** window

Double-click on the Model column for the **C5 1** model

In the popup window, Click **Generate...Model to Palette**

Close the **C5 1** window

Close the Auto Classifier Browse window

The models manager now contains the two models. We'll add them to the stream connected to the Type node.

Click once on each model, then click in the stream below the Type node

Connect the **Type** node to the **C&R Tree 1** model, and then connect that model to the **C5 1** model

The Analysis node can also be used to directly compare the performance of two or more models. When two models are placed in the same stream, they can be jointly assessed with an Analysis node. Additional tables will be produced that report on the number of records where the models agree, and disagree, in their predictions, and the model accuracy when they agree.

Add an **Analysis** node to the stream from the Output palette

Connect the **C5 1** model to the **Analysis** node

Edit the **Analysis** node

Click **Coincidence matrices (for symbolic targets)** check box (not shown)

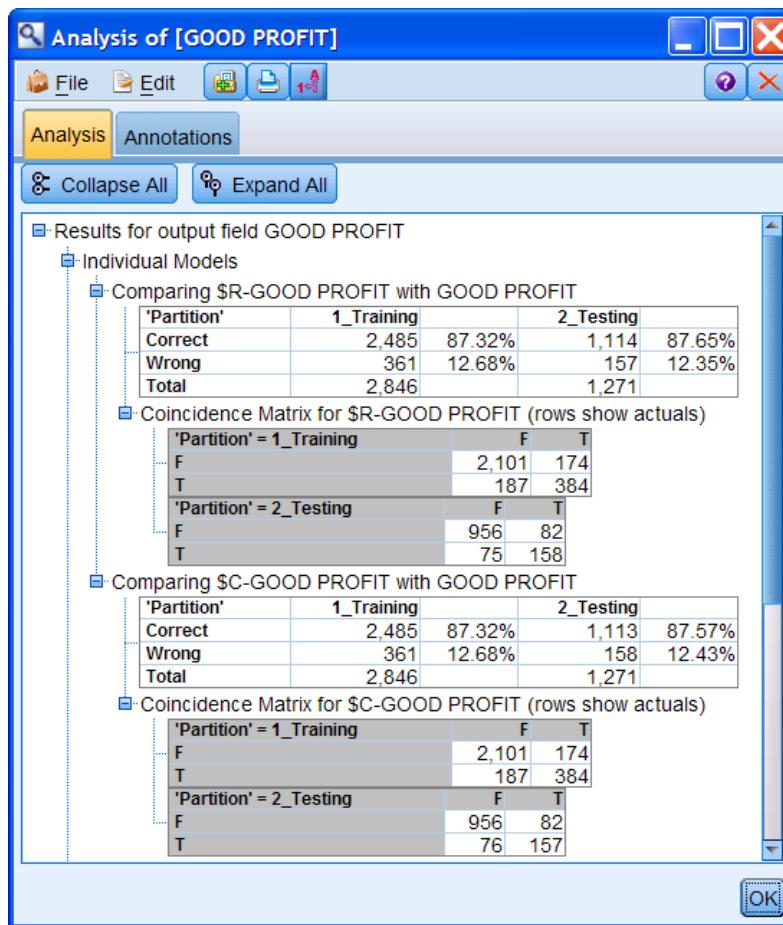
Click **Run**

The first set of output tables repeats the separate model evaluation that we just reviewed in Lesson 15. We'll look at these to put the model comparison in context.

On the Testing partition, the C&R Tree model is accurate on 87.65% of the records. Notice that the output is not labeled by the model type but instead by the field with the model predictions (*\$R-GOOD PROFIT* for C&R Tree). Looking at the Coincidence Matrix (with the predicted categories on the rows), the model is much more accurate on the false category (non-good risk customers). Accuracy on good risk customers is about 67.8% (158/ (75+158)) on the Testing partition, not very impressive.

The C5.0 model does similar good overall on the Testing data (87.57%), but does little worse at predicting good risk customers (67.4%).

Figure 16.2 Model Performance for C&R Tree and C5.0



The new output comparing models is in several tables below these. The first table shows how often predictions of the two models agree in both data partitions. In the Testing partition, the models agree on an astounding 99.9% of the records. In fact, in this partition, there is only 1 case on which they disagree. This is quite high, and the higher the better here in the event that we wish to combine the models (see Combining Models below).

The key question is then, when they agree, how accurate is the prediction? This is answered by the next table (*Comparing Agreement with GOOD PROFIT*), which shows that in the Testing partition, the two models are correct 87.64% of the time when they agree. This is very slightly better than the C5.0 model by itself, although the improvement is modest over the best model. Still, every increase in accuracy can be important in a data-mining project.

Figure 16.3 Coincidence Matrix for Model Agreement

The screenshot shows the Analysis output window for the 'GOOD PROFIT' analysis. It contains three tables:

- Agreement between \$R-GOOD PROFIT \$C-GOOD PROFIT**

'Partition'	1_Training	2_Testing		
Agree	2,846	100%	1,270	99.92%
Disagree	0	0%	1	0.08%
Total	2,846		1,271	

- Comparing Agreement with GOOD PROFIT**

'Partition'	1_Training	2_Testing		
Correct	2,485	87.32%	1,113	87.64%
Wrong	361	12.68%	157	12.36%
Total	2,846		1,270	

- Coincidence Matrix for Agreement (rows show actuals)**

'Partition' = 1_Training		F	T
F	2,101	174	
T	187	384	
'Partition' = 2_Testing		F	T
F	956	82	
T	75	157	

Then a third new table (*Coincidence Matrix for Agreement*) shows predictions for each category of *GOOD PROFIT* separately when the models agree. The accuracy for the *good risk* group in the Testing partition is now 67.7%. This is almost the same with C5.0 model by itself (the improvement in accuracy when using both models comes from the other category).

So if overall accuracy is important, combining the models would be an option; if accuracy for the *good risk* group is most important, then we might not do so.

16.3 Evaluation Charts for Model Comparison

Evaluation charts can easily compare the performance of different models. To demonstrate we will create a gains chart based on the predictions from the C&R Tree and C5.0 models. The target category by default will be *T* (good risk) because true values appear first in the list of categories for flag fields. An evaluation chart will be produced separately for each partition.

Close the **Analysis** output window

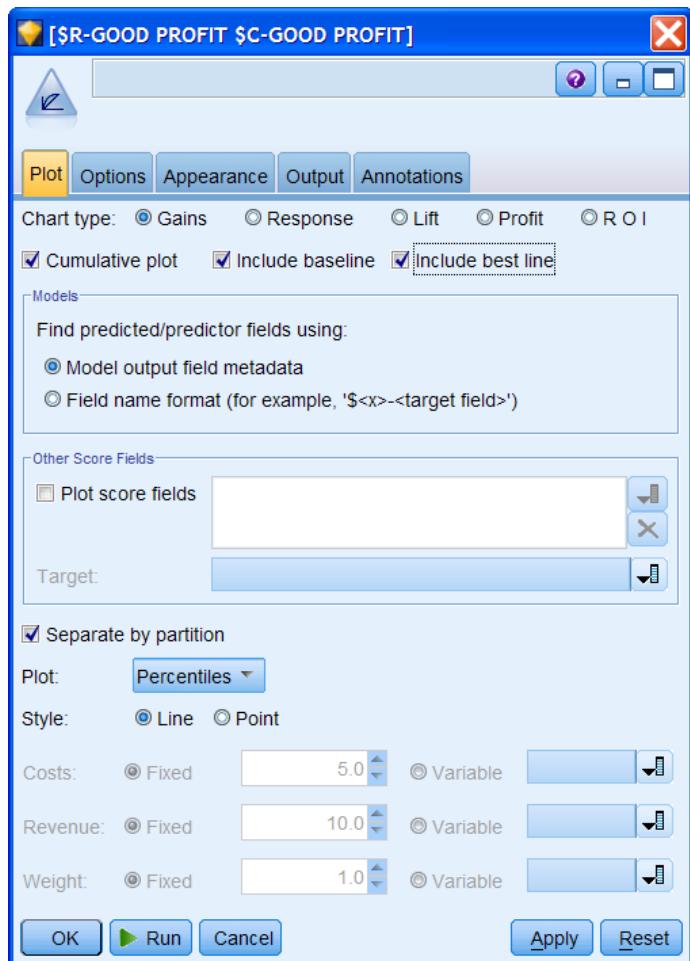
Place an **Evaluation chart** node from the Graphs palette to the right of the generated **C5** model in the Stream

Connect the **C5** model to the **Evaluation** node

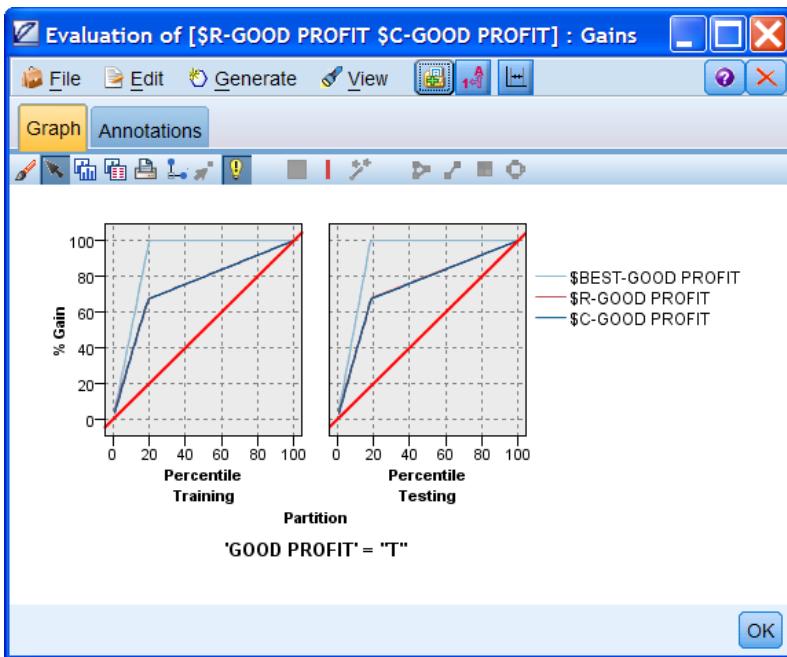
Edit the **Evaluation** node

Click **Include best line** check box

Click **Run**

Figure 16.4 Evaluation Chart Dialog

Now two lines (one for each model) are produced in addition to the baseline. There are also separate charts for the Training and Testing partitions.

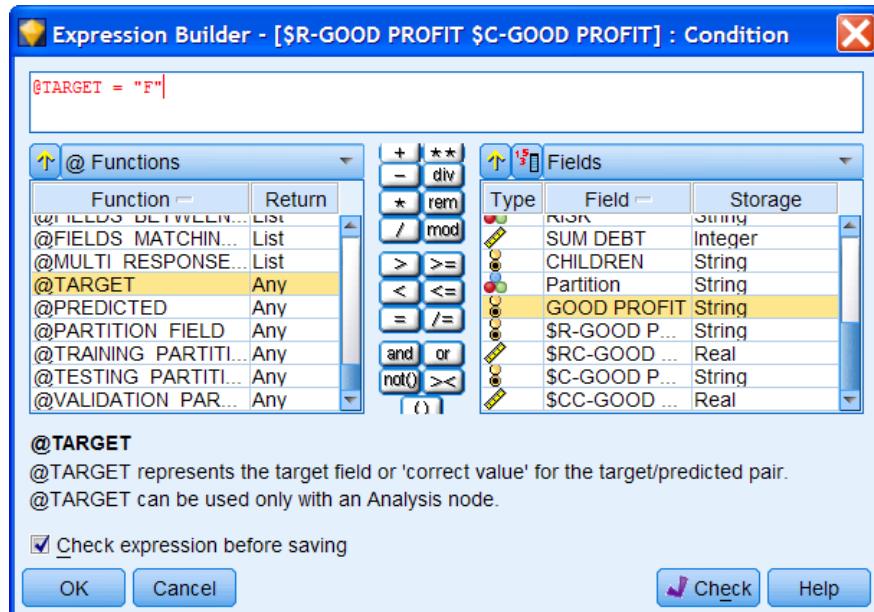
Figure 16.5 Gains Chart for Good Risk Category for the Two Models

The models perform extremely similarly on both partitions. Their lines overlap so closely they can't be separated at this scale. Both models match the theoretical best model performance up to about the twentieth percentile, but fall off rapidly after that.

Close the Evaluation chart window

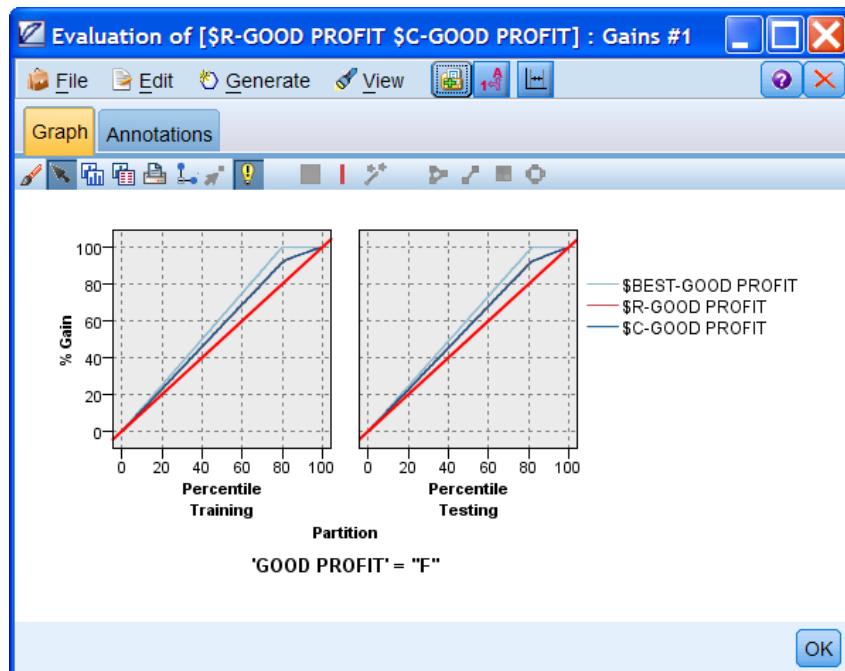
By default, the Evaluation chart uses the first category of the target field to define a hit. Let's change the target category to not good risk (F).

- Double-click the **Evaluation** node
- Click the **Options** tab
- Click the **User defined hit** check box
- Click the **Expression Builder** button in the **User defined hit** group
- Click **@Functions** on the functions category drop-down list
- Select **@TARGET** on the functions list, and click the Insert button
- Click the **=** button
- Right-click **GOOD PROFIT** in the Fields list box, then select **Field Values**
- Select **False: F**, and then click **Insert** button

Figure 16.6 Specifying Not Good Risk as the Hit Condition

Click **OK**
 Click **Run**

Model performance is clearly different at predicting customers who are not good risks. The C5.0 model outperforms the C&R Tree model, and by a similar margin on both the Training and Testing partitions. And both models, but especially C5.0, perform close to the theoretical maximum on a substantial fraction of the sample of customers.

Figure 16.7 Gains Chart for Not Good Risk Category for Two Models

In this way, different models can be compared in gains, lift, profit, or ROI chart form to supplement other model comparisons.

Close the **Evaluation** chart window

16.4 Combining Models

In classical statistical analysis, one model is normally used to predict a target. We may well try several types of models, but there will eventually be one model, whether it is based on logistic regression or hierarchical linear modeling, that we use to make predictions. Many data-mining projects also develop one model, but the spirit and intent of data mining argues for using more than one approach to a given problem.

PASW Modeler makes it straightforward to compare the results of multiple models. It also makes it uncomplicated to combine the results of two or more models, and in fact this is a common practice to improve performance. Doing so is based on the presumption that one technique will be better at predicting a particular outcome, while another technique will be better for a different outcome. Or, one technique will work better for one segment of customers, while another will do better for a different segment.

A basic procedure for combining models follows this logic:

1. Create two or more models and test them
2. When the models agree, use that prediction
3. When the models don't agree, use the model prediction with the highest confidence.

Alternatively, the models can “vote” in step 3, with the final prediction being the outcome predicted by more models. Thus, if there are four models, and three of them predict *good risk* for a customer while the fourth predicts *not good risk*, we predict *good risk*.

To remind ourselves about the confidence for predictions, let's look at the confidence values from the models.

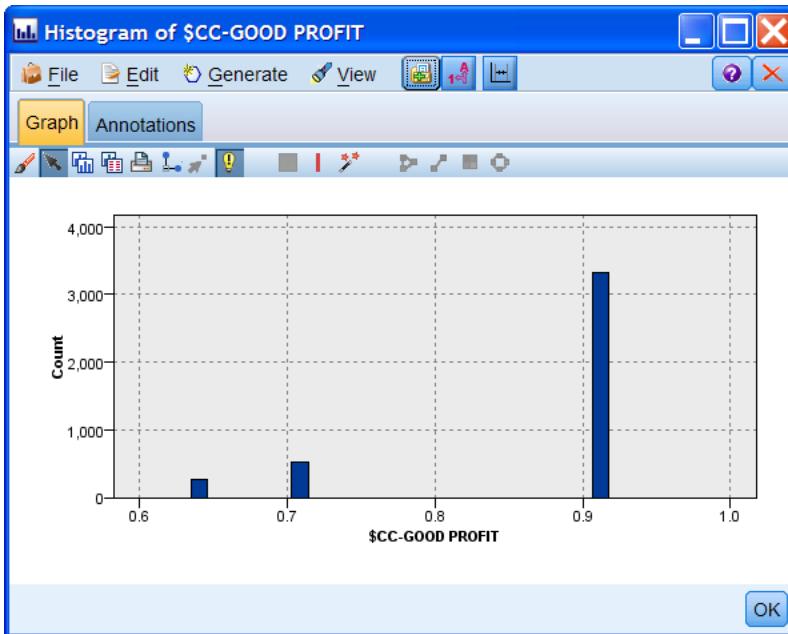
Add a **Table** node from the Output palette near the C5.0 model
Connect the **C5.0 model** to the **Table** node
Run the **Table** node

Figure 16.8 Confidence for the Model Predictions

The screenshot shows a software interface for viewing a dataset. The title bar says "Table (19 fields, 4,117 records)". Below the title bar are standard menu options: File, Edit, Generate, and several icons. There are two tabs: "Table" (which is selected) and "Annotations". The main area is a grid of data with four columns. The first column is labeled with numbers 1 through 20. The second column is labeled "\$R-GOOD PROFIT" and contains mostly "T" values. The third column is labeled "\$RC-GOOD PROFIT" and contains mostly 0.694 values. The fourth column is labeled "\$C-GOOD PROFIT" and contains mostly "T" values. The fifth column is labeled "\$CC-GOOD PROFIT" and contains mostly 0.713 values, with some 0.635 values interspersed. At the bottom right of the grid is an "OK" button.

The field *\$CC-GOOD PROFIT* contains the model confidence for the C5.0 model. The confidence ranges from 0 to 1. For a decision tree model, confidence is the percentage of cases in a terminal node in the predicted (or modal) category. In a flag target, it cannot fall below .50, although all these values are well above that.

Although confidence is measured on a continuous scale, a histogram will show only a few, discrete values because there are only a small number of nodes in the C5.0 tree. We won't bother to create that histogram, but it is displayed in Figure 16.9. There are only three discrete values of confidence because the C5.0 model is simple and has only three terminal nodes.

Figure 16.9 Histogram for Confidence for C5.0 Model Predictions

To automate combining models, PASW Modeler provides the Ensemble node, located in the Field Ops palette (because it combines model prediction fields). An Ensemble node generates a field containing the combined scores. The name is based on the target field and prefixed with \$XF- (for flag), or parallel names for a nominal or continuous target.

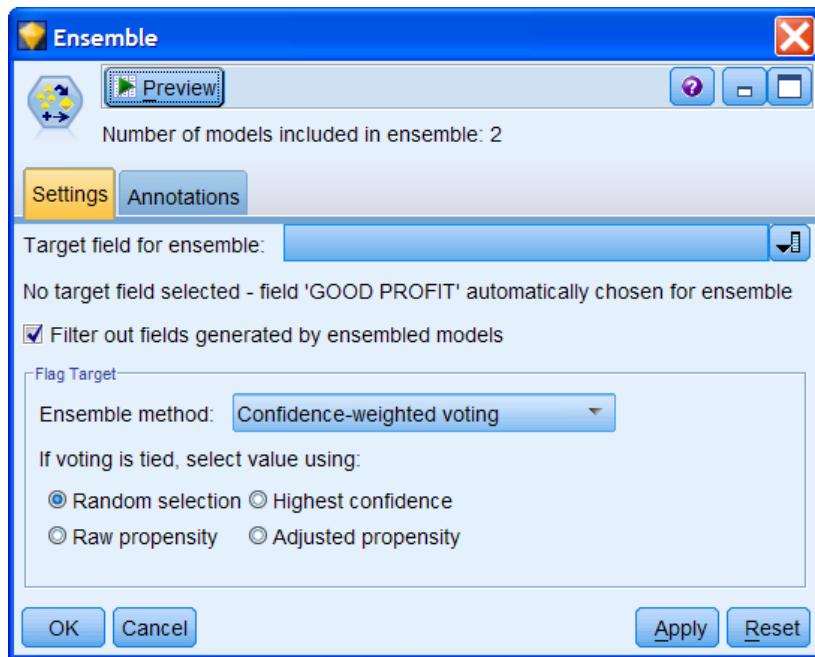
The combined prediction field can be created with several methods for categorical targets, including voting, which works by counting the number of times each possible predicted value is chosen and selecting the value with the highest total. Alternatively, votes can be weighted based on the confidence value for each prediction. The weights are then summed, and the value with the highest total is again selected. The confidence for the final prediction is the sum of the weights for the winning value divided by the number of models included in the ensemble. The default is to use confidence-weighted voting.

For continuous targets, the only available method is to average the predictions.

A quantity called *propensity* can also be used to combine predictions when the target is a flag. Propensity is the likelihood of an outcome (true for a flag target). If the model predicts the true value, then the propensity is the same as P, where P is the probability of the prediction (equal to the confidence for decision tree models). If the model predicts the false value, then the propensity is calculated as $(1 - P)$. Adjusted propensity values can be calculated on the Testing data partition.

In our example, we will use the default method of confidence-weighted voting, although because we are using only two models and have a flag target, the choices are somewhat less critical because several will result in the same prediction.

- Add an **Ensemble** node to the stream near the **C5** model
- Attach the **C5** node to the **Ensemble** node
- Edit the **Ensemble** node

Figure 16.10 Ensemble Node Dialog

The Ensemble node automatically recognizes *GOOD PROFIT* as the target because of the Role setting in the Type node upstream.

There are seven methods of combining predictions for a flag target, with the default being confidence-weighted voting. Votes can be tied, and if so, by default the node will randomly choose a winner. If this seems, well, too random, then three other choices are available.

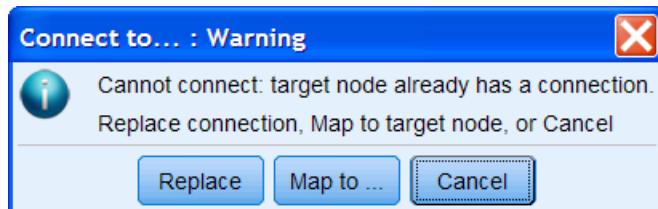
The original fields generated by the models will be filtered out by default so as not to clutter the stream.

There is no *Run* button because the Ensemble node creates a new field but provides no output directly. We'll use a Table node to view the output and turn off field filtering.

Click the **Filter out fields generated by ensembled models** check box to deselect it

Click **OK**

Connect the **Ensemble** node to the **Table** node that is currently connected to the C5 model

Figure 16.11 Warning About Connection Attempt

When we attempt to make this connection, PASW Modeler tells us that the Table already has a connection. We'll replace it.

Click **Replace**

Run the **Table** node

Figure 16.12 Ensemble Prediction and Confidence

The screenshot shows a Windows application window titled "Table (21 fields, 4,117 records)". The menu bar includes "File", "Edit", "Generate", and various icons. Below the menu is a toolbar with "Table" and "Annotations" tabs, where "Table" is selected. The main area is a grid displaying data for 20 records (1 through 20). The columns are labeled: C-GOOD PROFIT, SCC-GOOD PROFIT, \$XF-GOOD PROFIT, and \$XFC-GOOD PROFIT. The first three columns contain numerical values (e.g., 0.713, 0.635), while the fourth column contains binary values (T or F). An "OK" button is visible at the bottom right of the grid.

	C-GOOD PROFIT	SCC-GOOD PROFIT	\$XF-GOOD PROFIT	\$XFC-GOOD PROFIT
1		0.713	T	0.703
2		0.713	T	0.703
3		0.635	T	0.664
4		0.635	T	0.664
5		0.713	T	0.703
6		0.635	T	0.664
7		0.713	T	0.703
8		0.635	T	0.664
9		0.635	T	0.664
10		0.635	T	0.664
11		0.635	T	0.664
12		0.713	T	0.703
13		0.713	T	0.703
14		0.713	T	0.703
15		0.713	T	0.703
16		0.713	T	0.703
17		0.713	T	0.703
18		0.635	T	0.664
19		0.635	T	0.664
20		0.635	T	0.664

The Ensemble prediction is in the column *\$XF-GOOD PROFIT* with the confidence in the next column. For the first record, both the C&R Tree and C5.0 models predict True, so of course the Ensemble prediction is True also. The confidence is the average of 0.694 and 0.713, or .703. To find a record where the original models disagree, we need to scroll down to record 665.

Scroll down until Record 665 is visible.

Figure 16.13 Prediction and Confidence When Models Disagree

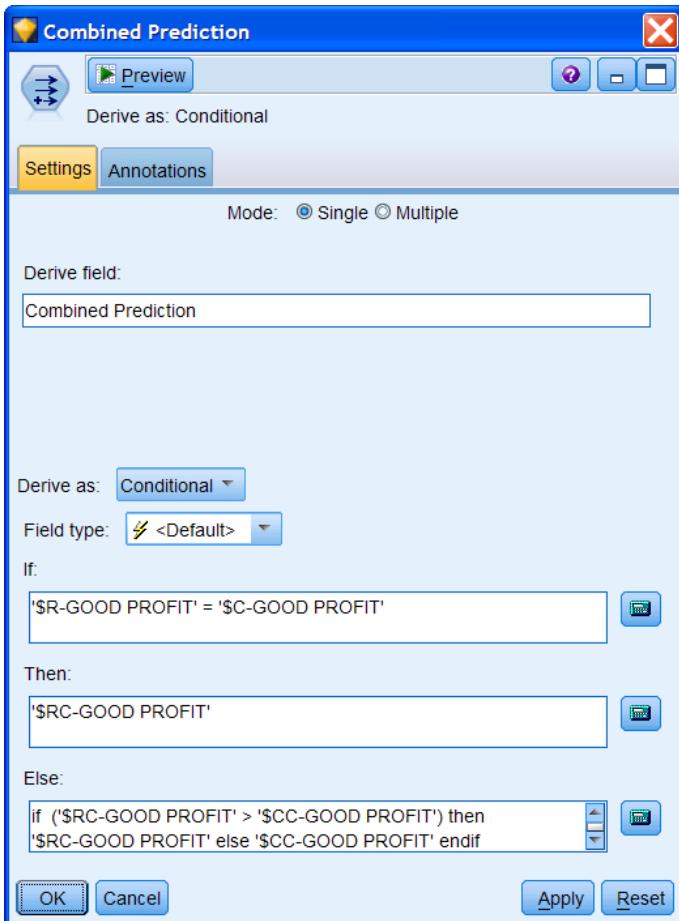
	\$C-GOOD PROFIT	\$CC-GOOD PROFIT	\$XF-GOOD PROFIT	\$XFC-GOOD PROFIT
653	T	0.713	T	0.703
654	T	0.713	T	0.703
655	T	0.713	T	0.703
656	T	0.713	T	0.703
657	T	0.713	T	0.703
658	T	0.713	T	0.703
659	T	0.713	T	0.703
660	T	0.713	T	0.703
661	T	0.713	T	0.703
662	T	0.713	T	0.703
663	T	0.713	T	0.703
664	T	0.713	T	0.703
665	F	0.918	F	0.459
666	T	0.713	T	0.703
667	T	0.713	T	0.703
668	T	0.713	T	0.703
669	T	0.713	T	0.703
670	T	0.713	T	0.703
671	T	0.713	T	0.703
672	T	0.713	T	0.703

For record 665, the actual value of *GOOD PROFIT* is *T*. The C&R Tree model correctly predicts this value, but not C5.0. The confidence of the C&R Tree prediction is 0.694. The prediction of the Ensemble node is *F* because the C5.0 confidence is the highest. Although the C&R Tree prediction is correct, the Ensemble prediction becomes $0.918/2=0.459$. This is quite low, but that is reasonable given that only one original model made an accurate prediction.

Using the Ensemble prediction field, we can now explore how the joint model operates using the same techniques as in Lesson 15.

Creating a Combined Field Manually

You may wish to create a combined field on your own from model predictions. This is easy to do using a Derive node. The expression to create a field is displayed in Figure 16.14.

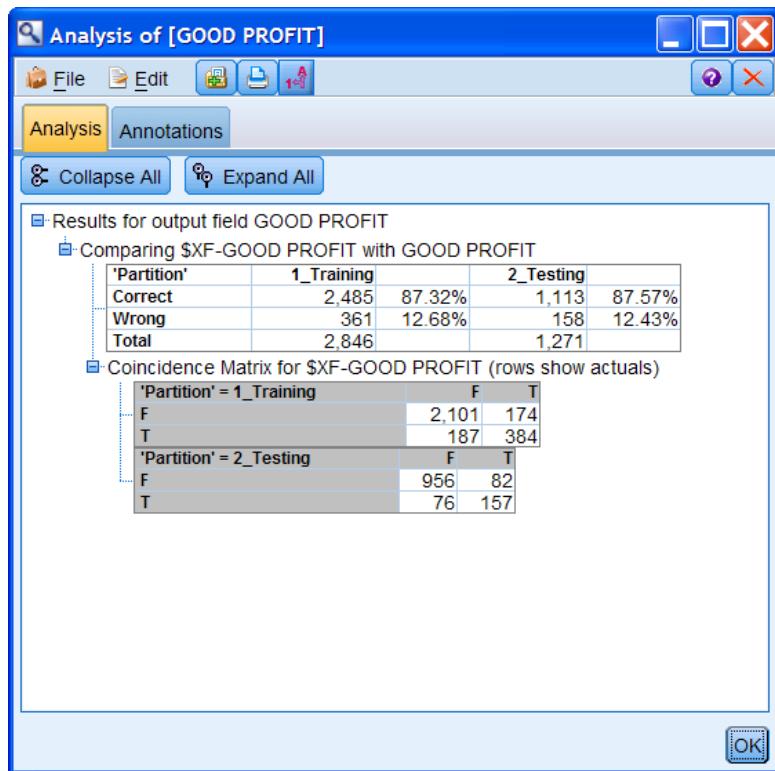
Figure 16.14 Combining Model Predictions

This is a conditional Derive node. When the predictions of each model are the same, then we use the prediction from the C&R Tree model (either prediction could be used).

When the predictions differ, we then need to pick the prediction from the model with the highest confidence for its prediction. This is done in the Else: expression box using the Clem language to create logical comparisons and assignments. The Clem language, as mentioned previously, is case sensitive. The if expression must conclude with an *endif* statement.

We conclude this lesson by using an Analysis node to check the accuracy of the combined models. We'll also filter out the other model fields to reduce clutter in the Analysis node output.

- Close the Table window
- Edit the **Ensemble** node
- Click the **Filter out fields generated by ensembled models** check box (not shown)
- Click **OK**
- Add an **Analysis** node to the stream by the **Ensemble** node and connect the two
- Edit the **Analysis** node
- Click **Coincidence matrices (for symbolic targets)**
- Click **Run**

Figure 16.15 Analysis Output for Combined Models Predictions

The combined model is accurate on 87.57% of the records. The combined model is accurate on 67.4% of the *good risk* customers, essentially the same as estimated by the Analysis node output (differences are caused by the confidence weighting).

Summary

In this lesson you have been introduced to techniques to test and compare models. You should now be able to:

- Use the Analysis node to test model predictions on the Testing dataset
- Use the Analysis and Evaluation Charts node to compare two or more models
- Use the Ensemble node to automatically combine models

Exercises

In this exercise, we will use the Analysis node and Evaluation charts to compare the results of the C5.0 and CHAID models.

1. Open the *ExerLesson12.str* stream and delete all nodes downstream from the C5.0 model and CHAID model nodes.
2. Connect an Analysis node to the C5.0 model and request Coincidence matrices. Review the results.
3. Now, connect the Analysis node to the CHAID model and review the results. Is one model better than the other in overall prediction and predicting Responders?
4. Next compare the two models. Connect the C5.0 model to the CHAID model. Run the Analysis node now at the end of the C5.0 model - CHAID model chain. Further compare the two models by attaching an Evaluation node and request a gains chart with best-fit line. Edit the Target to show Responders in the gains charts.
5. Add an Ensemble node to the stream after the combined models. Then add a Table node to view the combined predictions.
6. Try using different methods of voting in the Ensemble node, and then use the Analysis node to assess the accuracy of the combined models. What is the most accurate method of combining the models? How much better is the combined model than either the C5.0 or CHAID model by itself? On which target category does it have the most improvement?

Lesson 17: Deploying and Using Models

Objectives

- Deploying a Model
- Exporting Model Results
- Model Performance
- Model Lifetime
- Updating a Model

Data

In this lesson we will use the data file *RiskNew.txt* that contains information on new customers on which we would like to make a prediction with the C5.0 model from Lesson 12.

17.1 Introduction

Once a model is developed, whether it is a single model or a combination of models, it must be applied to new data to make predictions. In data mining, the act of using a model on new data is termed *deployment*. There are many options for deploying PASW Modeler models. In this lesson, we focus on a broader view of the process, providing general advice on using models, model performance, updating models, and model lifetimes.

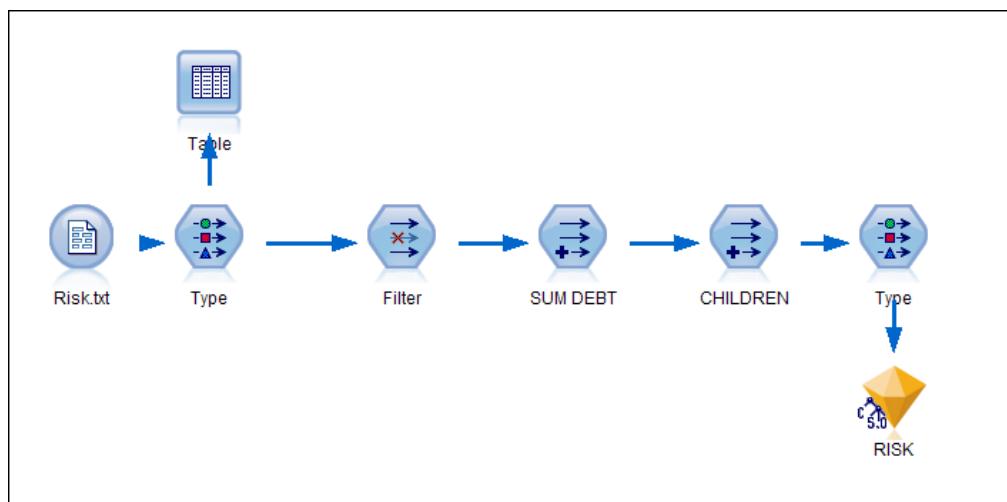
17.2 Deploying a Model

As we have seen in previous lessons, models that PASW Modeler generates can be placed in a stream and used to make predictions. But models just as easily can be used in a similar stream to make predictions for new data. To illustrate this process, we have modified the *Rule Induction.str* stream file to create a typical deployment stream, labeled *Deploy.str*. This stream uses only the C5.0 model.

Let's open and examine this stream.

Click **File...Open Stream** (navigate to the c:\Train\ModelerIntro folder if necessary)
Double-click **Deploy.str**

Figure 17.1 Stream Used for Deployment of C5.0 Model



This stream contains only the essential nodes, including a Source node for the data (not referencing the new data yet), a Type node, Derive nodes to create new fields, another Type node, and the C5.0 model. If we would like to use PASW Modeler to make predictions on new data, this stream will be adequate, with a few changes.

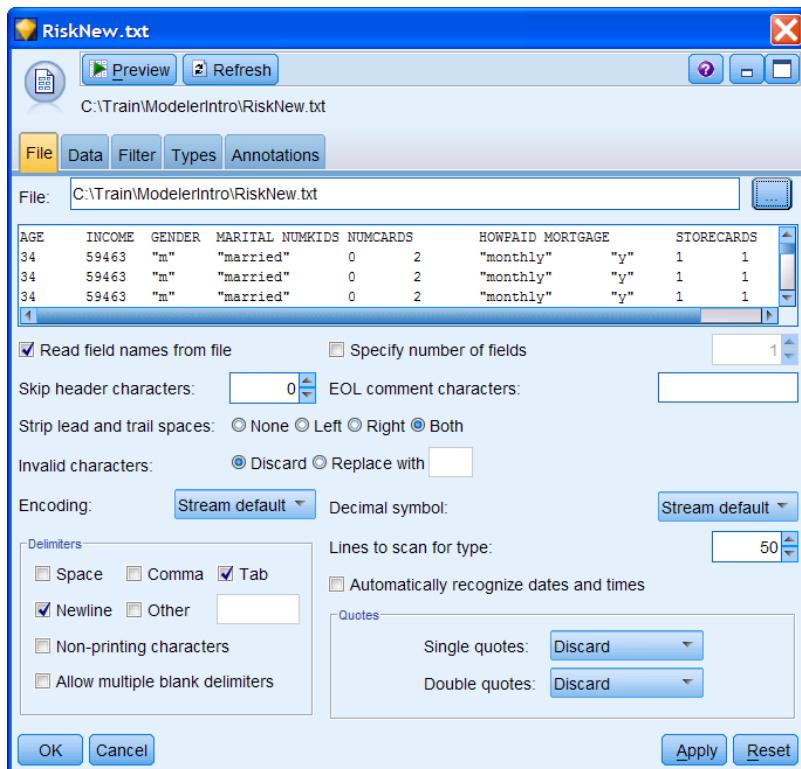
Suppose that we are given new customer data by the financial institution on which we want to make predictions on *RISK* for each customer. These new data are in the file *RiskNew.txt*, which is a text file with the same structure as *Risk.txt*. Of course, there is no *RISK* field in these data because that information is precisely what we wish to predict. This data file is larger than either the training or testing data.

The Type node directly after the data source node contains information for *RISK*. Will this be a problem when we read in the new data? We'll answer this question by changing the source and executing the Table node.

Edit the **Risk.txt** source node

Change the data file to **RiskNew.txt** (you can either change the text or locate the file and double-click on it)

Figure 17.2 Changing to RiskNew Data File



Click the **Preview** button

Figure 17.3 RiskNew Data File

The screenshot shows a software window titled "Preview from RiskNew.txt Node (10 fields, 10 records)". The window has a menu bar with "File", "Edit", "Generate", and "Help". Below the menu is a toolbar with icons for "Table", "Annotations", "OK", and "Cancel". The main area is a table with 10 rows and 8 columns. The columns are labeled: AGE, INCOME, GENDER, MARITAL, NUMKIDS, NUMCARDS, HOWPAID, and MORTGAGE. The data shows values for each field across 10 records.

	AGE	INCOME	GENDER	MARITAL	NUMKIDS	NUMCARDS	HOWPAID	MORTGAGE
1	34	59463	m	married	0	2	monthly	y
2	34	59463	m	married	0	2	monthly	y
3	34	59463	m	married	0	2	monthly	y
4	34	59463	m	married	0	2	monthly	y
5	34	59463	m	married	0	2	monthly	y
6	34	59463	m	married	0	2	monthly	y
7	34	59463	m	married	0	2	monthly	y
8	34	59463	m	married	0	2	monthly	y
9	34	59463	m	married	0	2	monthly	y
10	34	59463	m	married	0	2	monthly	y

There is no *RISK* field in this data file. And PASW Modeler does not report an error even though we ran the data through a Type node with that field defined. We'll edit the Type node to check its status.

- Close the Preview window
- Click **OK** to return to the stream canvas
- Edit the **Type** node

Figure 17.4 Type Node Without RISK

The screenshot shows a software window titled "Type". The window has a toolbar with "Preview", "OK", and "Cancel". Below the toolbar is a tab bar with "Types", "Format", and "Annotations", where "Types" is selected. The main area is a table with columns: Field, Measurement, Values, Missing, Check, and Role. The table contains the following data:

Field	Measurement	Values	Missing	Check	Role
AGE	Continuous	[18,50]	None	None	Input
INCOME	Continuous	[15005,59...]	None	None	Input
GENDER	Flag	mf	None	None	Input
MARITAL	Nominal	divsepwid,...	None	None	Input
NUMKIDS	Continuous	[0,4]	None	None	Input
NUMCARDS	Continuous	[0,6]	None	None	Input
HOWPAID	Flag	weekly/mo...	None	None	Input
MORTGAGE	Flag	y/n	None	None	Input
STORECARDS	Continuous	<Read>	None	None	Input
LOANS	Continuous	[0,3]	None	None	Input

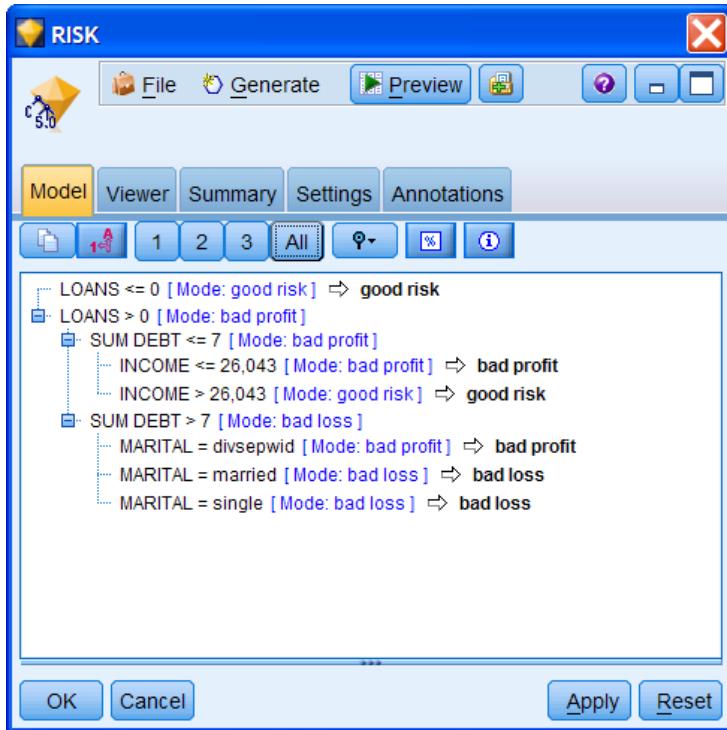
At the bottom, there are two radio buttons: "View current fields" (selected) and "View unused field settings". There are also "OK", "Cancel", "Apply", and "Reset" buttons.

Conveniently, PASW Modeler has deleted the non-existent field *RISK* from the Type node, without any work on our part. All the other fields are defined.

If we review the Values column, we see that the range of values in the new customer data—such as from 0 to 3 for *LOANS*—is the same as for the Training data. Ideally, when developing a model, you want to include the full possible range of data for each field as there could be in future data. This can be impossible sometimes, and we might encounter a new customer in the future with, say, a value of 5 for *LOANS*. Will the C5.0 model handle this customer? Can it make a prediction?

To answer this question, we can review the model's rules, which we have repeated in Figure 17.5.

Figure 17.5 Rules for C5.0 Model to Predict RISK



The rules use a cut point of 0 for *LOANS*. And note there is no upper limit to the rules for customers with *LOAN* values above 0. These rules would apply equally to someone with 2 loans or 10 loans. In most cases with fields of measurement level Continuous, as with *LOANS*, values higher, or lower, than the range of data on which the model was developed will still be covered by the rules.

This isn't necessarily true for categorical fields. Imagine that, for some reason, we had no married people in the Training data. We would get different rules for the model since one rule uses married respondents to make a prediction of *bad loss*. But if the new data *did* have married customers, the model still wouldn't use that category to make predictions, except as a default (as it does for those who are *single*, but not because there were no customers who were single in the data).

So if categorical fields have new values in the future, this can cause difficulties. The model will still make a prediction for every customer, but the predictions might be more accurate if the missing category had been included in the training data. (It should also be noted that, even for continuous fields, values too far outside the original data range can be of concern. If a customer had a *SUM DEBT* value of 30, would we still be comfortable using the model to make a prediction?)

It is too late to solve this problem when developing the model; instead, careful attention to representative data is a necessity at the earliest stages of a data-mining project.

Click Cancel

We've changed the data file and run data through the first Type node. We'll run data through the second Type node, which will also be adjusted by PASW Modeler to match the data file.

Add a **Table** node from the Output palette to the stream near the second Type node
 Connect the **Type** node to the **Table** node
 Run the **Table** node
 Close the Table window

We are ready to make predictions for the new customer data. There is nothing special to do other than to run the data through the C5.0 model. We'll look at the predictions with a Table node.

Connect the **C5.0** model to the **Table** node
 Click **Replace** to the Modeler message
 Run the **Table** node

Figure 17.6 Predictions for RISK for New Customers

SE	STORECARDS	LOANS	SUM DEBT	CHILDREN	\$C-RISK	\$CC-RISK
1	1	1	4	F	good risk	0.634
2	1	1	4	F	good risk	0.634
3	1	1	4	F	good risk	0.634
4	1	1	4	F	good risk	0.634
5	1	1	4	F	good risk	0.634
6	1	1	4	F	good risk	0.634
7	1	1	4	F	good risk	0.634
8	1	1	4	F	good risk	0.634
9	1	1	4	F	good risk	0.634
10	1	1	4	F	good risk	0.634
11	1	1	4	T	good risk	0.634
12	1	1	4	T	good risk	0.634
13	1	1	4	T	good risk	0.634
14	1	1	4	T	good risk	0.634
15	1	1	4	T	good risk	0.634
16	1	1	4	T	good risk	0.634
17	1	1	4	T	good risk	0.634
18	1	1	4	T	good risk	0.634
19	1	1	4	T	good risk	0.634
20	1	1	4	T	good risk	0.634

Two fields have been added to the data, the model prediction (**\$C-RISK**), and the model confidence (**\$CC-RISK**).

Model deployment can be as simple as this with PASW Modeler. This type of deployment is often called *batch scoring* because we are making predictions for a group of records, rather than scoring individual records one-at-a-time.

Close the Table window

17.3 Exporting Model Results

Once you have predictions on new data, you will need to use them in some fashion, either to send out a mailing or email, place them in a database so they can be accessed with other tools, or create a report on the predictions. In many instances, you may want to export the results as a data file to another format. PASW Modeler has several types of file exports.

Click the **Export** tab in the Palettes area

Figure 17.7 Export Nodes

In all instances, you can attach one of these nodes after a generated model node and use it to write data to an external file.

The Database export node writes data directly into an ODBC-compliant relational data source. In order to write to this type of source, it must exist and you must have write permission for it (and, very likely, you will need to coordinate this with the staff that maintains databases for your organization).

The Flat File export node writes data to a delimited text file (with data values separated by commas, tabs, or other characters). A wide variety of other software can read this type of file.

The Statistics Export node writes data in PASW Statistics file format (.sav). This file can be read by PASW Statistics and other SPSS products. This is also the format used for PASW Modeler cache files.

The Data Collection Export node saves data in the format used by SPSS® PASW Data Collection market research software, based on the PASW® Data Collection Data Model. This format distinguishes case data—the actual responses to questions gathered during a survey—from the metadata that describes how the case data is collected and organized.

The SAS Export node writes data in SAS format to be read into SAS or SAS-compatible software. Three file types are available, including SAS for Windows/OS2, SAS for UNIX, or SAS Version 7/8/9.

The Excel export node saves data in Microsoft Excel format (.xls). You can also choose to launch Excel automatically and open the exported file when the node is run.

The XML Export node, new to PASW Modeler 14.0, enables you to output data in XML format, using UTF-8 encoding. You can optionally create an XML source node to read the exported data back into the stream.

All of these export nodes also contain a Publish tab. Publishing is done with the full stream, and it allows you to export entire PASW Modeler streams in order to embed the streams in your own external applications. This permits you to use that stream in a production environment.

17.4 Assessing Model Performance

To measure model accuracy/performance, we must wait for an outcome to occur, and then use these data to assess the model. Typically, outcomes will be added to an existing data warehouse or database (where an outcome can be a customer purchase, customer revenue, etc.). You can then read these data into PASW Modeler, using the existing stream file that was used for deployment. Now you have both a prediction and an outcome, just as when the model was being trained. If the outcome has the same field name, you can run the data through the generated model and use all the tools of PASW Modeler, such as the Analysis node, to measure performance.

Alternatively, you can add the model's predictions to a database, and then use other tools to assess the model once the outcome has been added to the database.

There can be a bit of a chicken-and-egg problem to testing how well a model performs when that model is used to determine how a customer is treated, e.g., which offer he or she receives. Consider two customers, A and B. Suppose our model predicts that A will not be a large source of future revenue, but that B will be. Therefore, we send an offer to B, but not to A. After we wait some specified time period, we can measure the amount of revenue received from B and see whether the model was accurate, but, what about A? How do we *really* know that customer A would not have been a good source of revenue?

The answer to these questions is that, not surprisingly, we don't know what would have happened if the offer had been given to customer A. Overall revenue per customer might increase after we begin using the model, but even that doesn't guarantee that customer A should not have received an offer.

In the case of the financial institution, they likely will not offer loans to customers who are predicted to be a *bad loss*, but as with our simple example of customers A and B, they won't know for sure that a customer predicted to be a *bad loss* would have been one if given a loan.

Many data miners don't concern themselves with this problem since it seems a bit intractable. There is a technique that can be used to further assess the model before it is fully deployed, and this process is used by some companies where the stakes are high because of cost or projected revenue.

A model can be trained and tested, and then it can be used to make predictions for a specified time period, but during that period, the model's predictions are not used for customers. Instead, the old system of operation is used to make offers to customers, and then the results from that system are compared to what the model predicted. So in this arrangement, the model might predict that a customer will be a *bad loss*, but if they are still offered a loan, what happens?

Understandably, following this plan further delays the implementation of a model, but in mission-critical applications, the delay can be worth it. Also, there are some projects in which the outcome occurs too far in the future to successfully implement this approach (and loans to customers might fall into that group). But this method will further validate and assess the model in a way that the testing data cannot.

Model Accuracy

When a model is finally fully deployed, it is critical that statistics be maintained of the model's accuracy. The time period you use to report on accuracy can be anything convenient and meaningful for your business and the data. You will also want to use other statistics to measure model performance, especially if the outcome you are predicting is a continuous field, such as customer revenue or sales (examples are the absolute error in prediction, the standard deviation of the error, and the correlation between the actual revenue and the model prediction).

You will probably also want to track error in important segments or subgroups. Although the model may not have been developed separately for customer segments, it is still reasonable to measure how well it is performing in important segments. You may discover that accuracy varies considerably across segments, which could eventually lead to refinement of the model (see below).

It is important to establish criteria in advance for assessing model performance, rather than constructing them after first seeing some results. This is partly to get buy-in from key players and

groups within the organization. There should be agreement on what constitutes a successful data-mining project.

Eventually all models will be replaced, so keeping track of model performance will hopefully give you clues about what changes should be made when developing the next generation model.

17.5 Model Lifetime

As just noted, no model lasts forever. Model performance deteriorates for many reasons. These include:

1. A changing competitive environment
2. A change in the type of customer using your products or services
3. A change in the products and services of your firm
4. A change in marketing and sales strategies by your firm
5. A change in the general economy that impacts your firm

This list could be extended, but is comprehensive enough to illustrate how fragile a model can be. On the other hand, many models may have a long shelf-life. A book-of-the-month type club which has adjusted to the challenges of the Internet and online bookstores may develop a model that will accurately predict who will buy books from the club. This model might be a satisfactory performer for several years, since the factors that lead people to regularly buy books are fairly stable.

Whatever the situation in your industry, you need to constantly watch for changes that might impact a model's performance. Of course, if you are faithfully measuring model performance, when these changes cause the model to perform poorly (rarely do changes lead to improvement), you'll know that there is a problem with the model. But it is much better to anticipate changes as soon as possible and begin developing a new model.

There is no absolute rule for the lifetime of a model, since that depends on so many factors. For example, the airline industry is very competitive with rapid changes, so we can imagine that models there might be constantly updated. In some industries, the ultimate version of this is a model that can re-estimate itself as new data is added a few cases at a time (PASW Modeler provides the Self-Learning Response Model node with this capability).

Model lifetime becomes particularly tricky when the outcome from a model—such as the ultimate disposition of a loan—will take several years to come about. In such a case, you may decide to redo models on a regular schedule and not wait for good evidence about how well a model is doing.

17.6 Updating a Model

When you have decided to develop a new model, you follow the same general steps as when the current model you are using was developed. If the data have not changed (in terms of available data sources and fields), then the process may be fairly quick. But having to develop a new model is an opportunity to look for additional data, so keep that point in mind. Don't presume that the same algorithm (C5.0) will work just as well on the most recent data. Unless you are under a severe time and resource crunch, the new model should be developed using all appropriate modeling techniques.

Developing the model should proceed through the same steps, and you will need both training and testing data files. A question that naturally arises is the appropriate time window from which to construct these two datasets. Going back too far in time will include data from when the current

model was performing adequately. Using only very recent data may compromise model development because there may not be sufficient data to create large enough data files to construct a robust model.

As with the matter of model lifetime, there is no absolute or correct answer to the appropriate time period. You need to balance the opposing factors and choose a sampling period best for your situation. One bit of general advice is that in situations where the environment is changing rapidly, shorter and more recent time periods should be favored.

Interestingly, when updating a model, we have a chance to test model accuracy with a method that wasn't available when the model was first developed. When a model is to be updated, it means that there is an existing model that has been deployed. And that means that the two models can be compared on the testing data to see which does better. In data mining the existing model is called the *Champion* and the potential new model is called the *Challenger*. You develop the Challenger per the methods we have used in this course guide, and you test it on the testing data. But because the test data are historical, you will also know how well the Champion did on these same records. This allows you to see which model is more accurate on the testing data, whether measured by overall accuracy or accuracy for important categories (such as *bad loss* for the financial institution).

If the Challenger does better than the Champion, it can be substituted and deployed on new data and the Champion retired for good. If the Challenger doesn't do better, then it is back to the drawing board, and the Champion is retained, at least temporarily.

Summary

In this lesson you have been introduced to deploying and using models. You should now be able to:

- Use a generated model in a PASW Modeler stream to make predictions on new data
- Understand the options to export results and data from PASW Modeler
- Understand how model performance can be measured
- Understand factors affecting model lifetime
- Plan for updating an existing model.

Exercises

In this exercise, you will do the simple batch deployment of the C5.0 model that you developed in Lesson 12 on a new set of data stored in the PASW Statistics data file, *CharityNew.sav*.

1. We have edited the *ExerLesson12.str* stream to contain just the nodes needed to deploy the C5.0 model. Open this stream, *CharityDeploy.str*.
2. Connect a Table node to the Type node to read and instantiate the data. Review the Type node to assure that the range of values and field names are the same as in the data that generated the model.
3. Connect a Table node to the C5.0 model and review the two new fields.
4. Although you can not properly evaluate the results of the scoring until the outcome is known, you might find it interesting to look at the percentage of "Responders" predicted in this new scored data versus the original data on which the model was built. For example, you could run a distribution graph on this scored data and compare to the actual number of responders in your original data. Hint: you will need another source node, or you can open a previously saved stream.

Appendix A: PASW Modeler Options and Stream Properties

Objectives

- To provide an overview of important PASW Modeler program options
- To provide an overview of stream properties

Data

None needed.

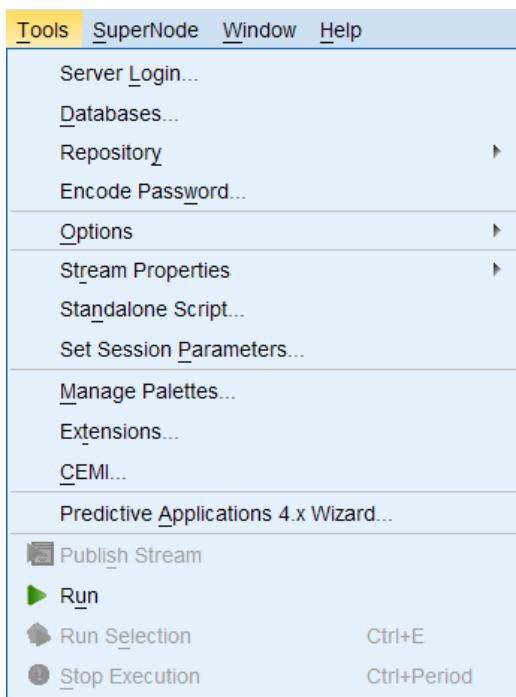
A.1 Setting PASW Modeler Options

Like most software programs, there are various default settings that control the operation of PASW Modeler, and many of these can be modified as suitable for your own purposes and style of using PASW Modeler. For example, changes can be made to:

- Memory usage
- Language
- Fonts and color
- Optimization of stream processing.

These options are generally accessed from the Tools menu. We won't need a stream to view or modify these options.

Click **Tools**

Figure A.1 The Tool Menu Choices

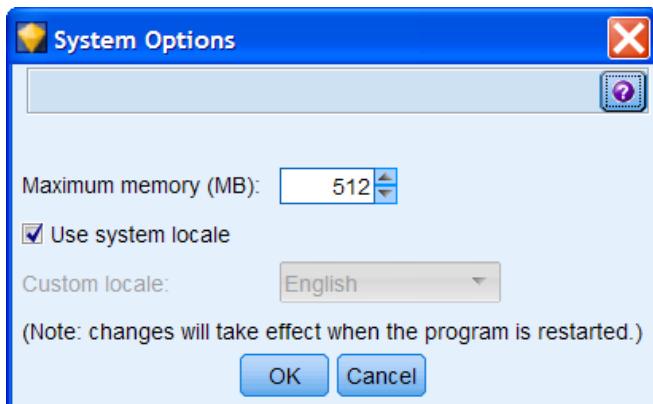
The Tools menu contains a variety of choices and options, only some of which we will discuss in this appendix. We will be concerned in this section with the System Options and User Options choices.

System Options

We'll first examine the available system options.

Click Options...System Options

The Maximum memory setting imposes a limit on PASW Modeler's memory usage (this is real memory, not virtual, or disk-based memory). You probably don't need to change this parameter unless you get an "out of memory" message.

Figure A.2 System Options Dialog

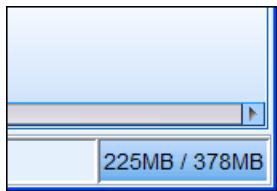
The preferred language locale for PASW Modeler is English by default. If you deselect the *Use system locale* check box you can choose from at least 9 other locales.

Changes made in this dialog box only take effect when PASW Modeler is restarted.

Click **Cancel**

Another method to manage memory is to force PASW Modeler to free up any available memory by clicking in the lower right corner of the PASW Modeler main window where the memory that PASW Modeler is using and the amount allocated are displayed.

Figure A.3 Current Memory Usage in PASW Modeler



In Figure A.3 the first number (225MB, or megabytes) refers to the amount of memory that is currently being used. The second number (378MB) is how much memory is allocated.

Clicking in this area turns it a darker shade, and after a short wait, it will turn back to grey and the memory allocation figures will drop, if possible.

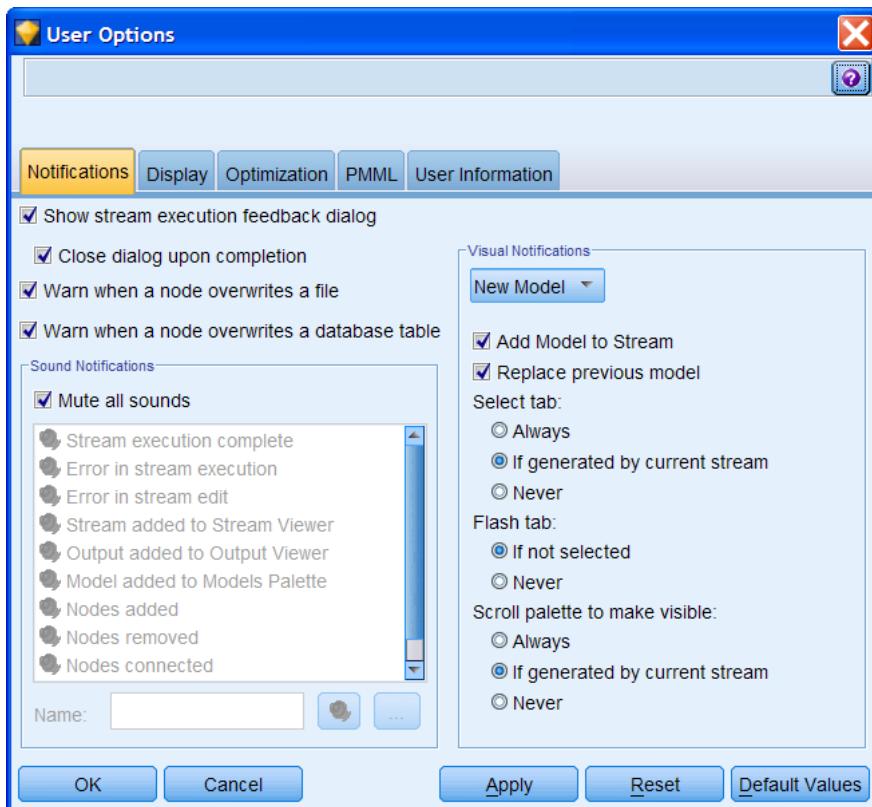
We'll look next at some of the available User Options.

User Options

User options, like system options, apply to all your streams. Most take effect immediately.

Click **Tools...Options...User Options**

The User Options dialog has several tabs to organize the alternatives. The first tab is Notifications. In this tab you can control options for the occurrence and types of warnings and confirmation windows. You can also change the behavior of how the Managers window handles new output and models.

Figure A.4 Notifications Tab

We mention just some of the more useful or critical options here.

Show stream execution feedback dialog. Normally PASW Modeler displays a progress indicator when stream execution has been in progress for three seconds or more. You can turn this off if you wish.

Warn when a node overwrites a file. Normally PASW Modeler will warn you with an error message when node operations overwrite an existing file. It is usually best to leave this turned on. The same applies to the warning when PASW Modeler overwrites a database table.

The options on the right side of the dialog box are used to specify the behavior of the Outputs and Models managers tabs when new items are generated. Select New Output or New Model from the drop-down list to specify the behavior of the corresponding tab.

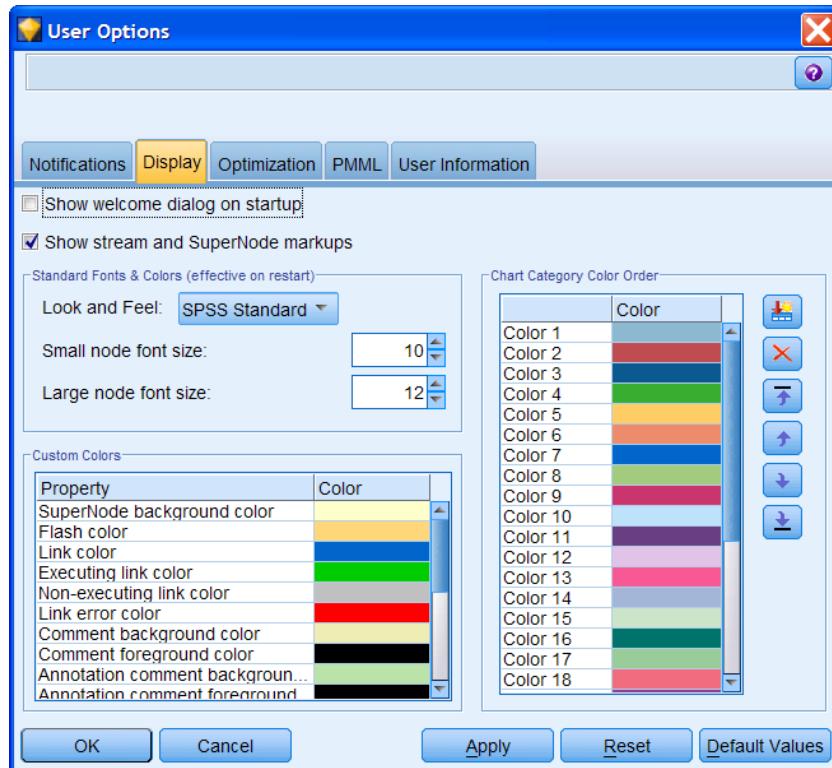
Scroll palette to make visible (New Model only). This is used to select whether to automatically scroll the Models tab in the Managers window to make the most recent model visible. That is normally a good idea, but you might not want to do this if you are running a script and generating several models.

Replace previous model (New Model only). In data mining it is common to work iteratively with a model, changing various settings and running the model several times before doing validation. Every time you run a modeling mode again, it will generate another model, and by default, it will overwrite the existing node with that name. Otherwise, you would have many instances of models in the manager window, including ones that you have found to be poorer performers. However, if you want

to be able to readily compare the various models you generate, then you may wish to retain them in the manager. You can either rename the model yourself each time, or you can deselect this check box.

Click **Display** tab

Figure A.5 Display Options



Most of the settings in this dialog are self-explanatory. The Custom Colors choices control the color scheme of PASW Modeler and the size of fonts used for display. These options are not applied until you restart PASW Modeler.

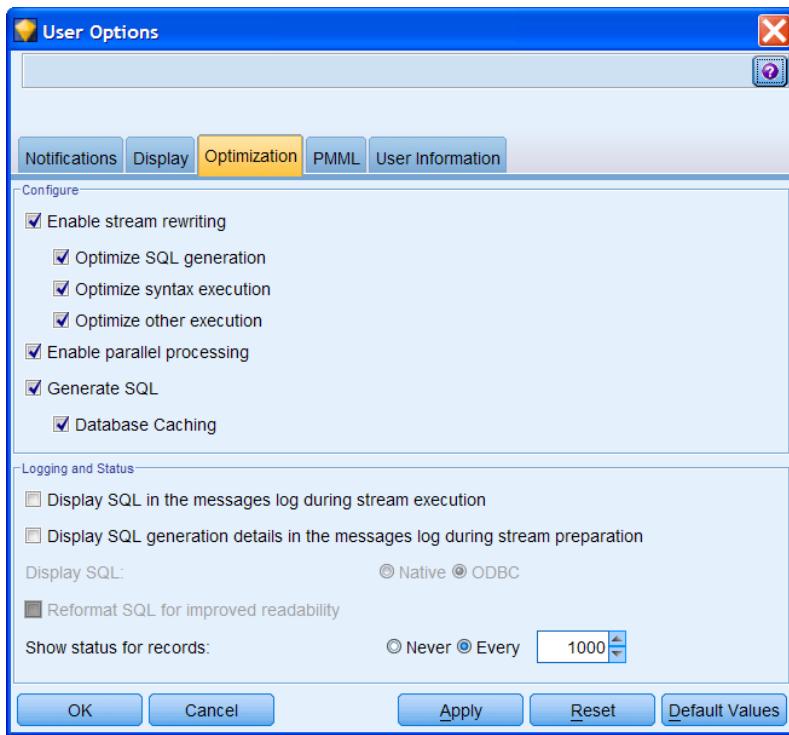
There is a general choice between the PASW Modeler blue-themed interface and a Windows standard color scheme.

For each of the items in the Custom Colors table, you can pick from a small list of other colors, or scroll to the bottom of the color drop-down list and select Color. This opens another dialog that allows you to choose a color from three color systems, including RGB.

The Chart Category Color Order table lists the currently selected colors used for display in newly created graphs. The order of the colors reflects the order in which they will be used in the chart. For example, if a set field used as a color overlay contains four unique values, then only the first four colors listed here will be used.

Click the **Optimization** tab

The Optimization tab in the User Options dialog box allows you to optimize PASW Modeler performance during stream execution..

Figure A.6 Optimization Options

Enable stream rewriting. Two types of stream rewriting (really reorganizing) are available, and you can select one or both. Stream rewriting reorders the nodes in a stream behind the scenes for more efficient execution by PASW Modeler.

Optimize SQL generation. This option applies when data are to be read from a database and allows PASW Modeler to reorder nodes in the stream so that more operations can be pushed back using SQL generation for execution in a database. When it finds a node that cannot be rendered into SQL, the optimizer will look ahead to see if there are any downstream nodes that can be rendered into SQL and safely moved in front of the problem node without affecting the stream semantics. Not only can the database perform operations more efficiently than PASW Modeler, but such pushbacks act to reduce the size of the dataset that is returned to PASW Modeler for processing..

Optimize other execution. This method of stream rewriting increases the efficiency within PASW Modeler of operations that cannot be delegated to the database. Optimization is achieved by reducing the amount of data in the stream as early as possible. While maintaining data integrity, the stream is rewritten to push operations closer to the data source, thus reducing data downstream for costly operations, such as joins.

Enable parallel processing. When running PASW Modeler on a computer with multiple processors, this option allows the system to balance the load across those processors, which may result in faster performance. In particular, streams with multiple terminal nodes may benefit from parallel processing, as will the following individual nodes: C5.0, Merge (by key), Sort, Bin (rank and tile methods), and Aggregate (using one or more key fields).

Generate SQL. Select this option to enable SQL optimization, allowing stream operations to be pushed back to the database by using SQL code to generate execution processes, which may improve performance. To further improve performance, Optimize SQL generation can also be selected,

allowing PASW Modeler to maximize the number of operations pushed back to the database. When operations for a node have been passed back to the database, the node will be highlighted in purple during execution.

Show status for records. PASW Modeler normally reports after every 1000 records as they arrive at terminal nodes. In large files you may want to change this setting to something greater so you receive less frequent updates.

Click **Cancel**

We turn next to stream properties in PASW Modeler.

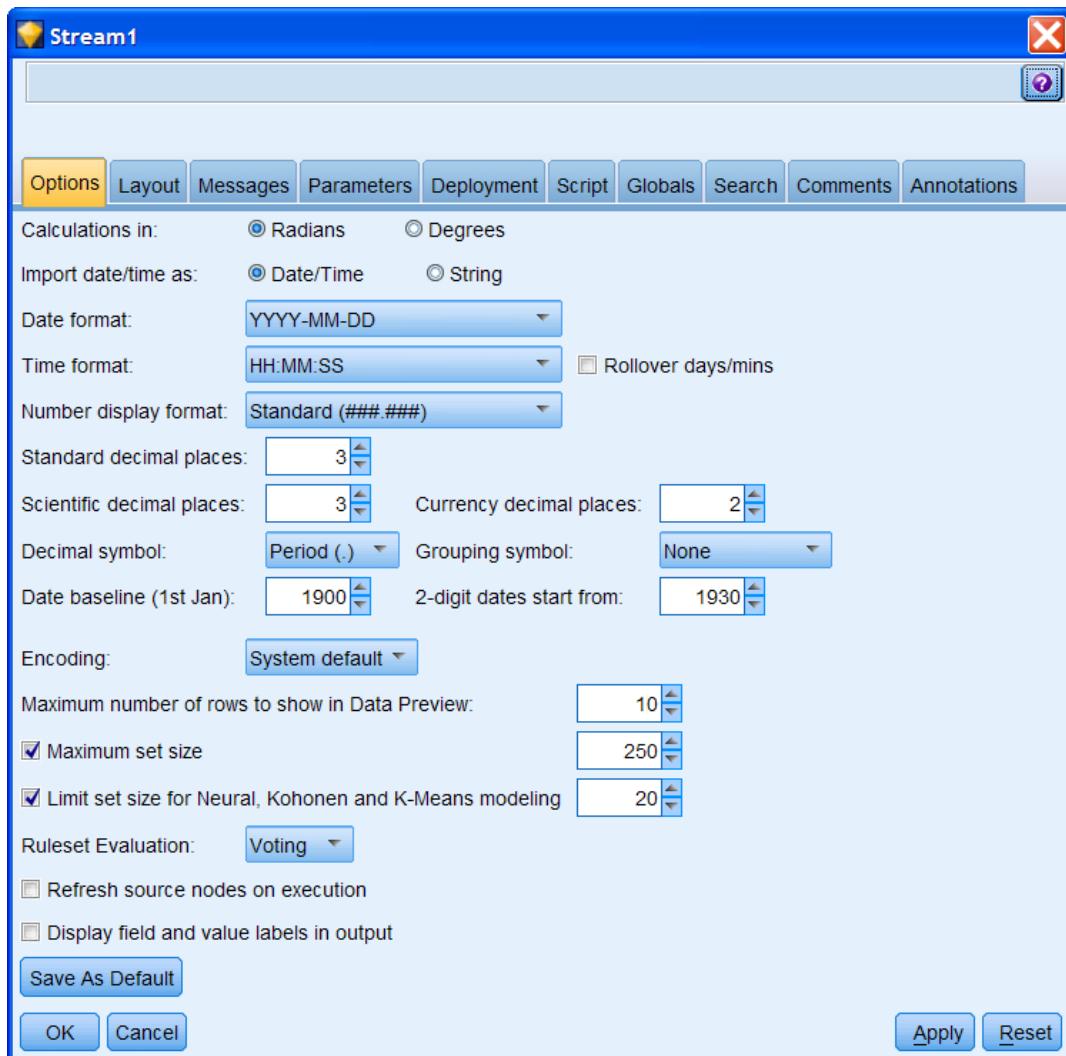
A.2 Stream Properties

Streams can be usefully thought of as PASW Modeler programs, and they have their own properties that control their execution. Since several streams can be used in a PASW Modeler session, the current stream (the one displaying on the Stream canvas) is affected by changes to the stream properties.

Click **Tools...Stream Properties...Options**

It may appear that there are several dialog boxes available under Stream Properties, but for ease of access, each of the tabs in the Stream Properties dialog has its own submenu choice.

There are many settings in this dialog, so we discuss only a few.

Figure A.7 Stream Options Tab

Dates and times are handled in PASW Modeler by setting a master format for date and time fields. Any string fields that conform to that format will be interpreted as a date, or time, respectively. Also, date and time fields read from databases will be set to these formats (if they have date or time storage).

If you have date or time fields that don't conform to these formats, you can use PASW Modeler transformations to modify the string accordingly.

The Date baseline (1st Jan) specifies the baseline year (beginning January 1) to be used by CLEM date functions that work with a single date. If you use 2 digit years, you can also control the cutoff year to add century digits. For example, specifying 1950 as the cutoff year will roll over 05/11/09 to the year 2009. The same setting will use the 20th century for dates after 50, such as 05/11/73 (1973).

For number display formats, you can specify the number of decimal places to be used when displaying or printing real numbers, in either standard or scientific notation.

Maximum set size is set to 250 by default and specifies the maximum number of values for set (categorical) fields. If there are more distinct values than this, the type of the field becomes typeless and will not be used in modeling. This property prevents you from using large sets, which can both slow stream execution significantly and degrade model performance. You can experiment with this if you have large set fields.

The Display field and value labels in output check box is deselected by default. This choice controls the displays of field and value labels in tables, charts, and other output. If labels don't exist, the field names and data values will be displayed instead. You can toggle labels on an individual basis elsewhere in PASW Modeler. You can also choose to display labels on the output window using a toggle button available on the toolbar.

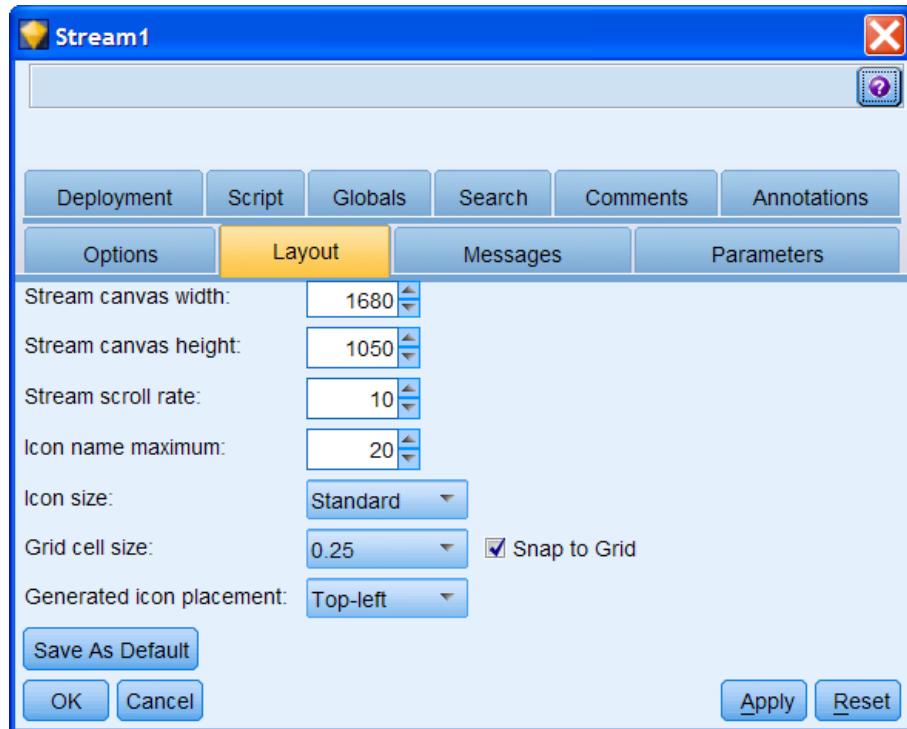
As mentioned above, changing settings in this dialog changes them only for the current stream. To make these settings the default for all streams, click Save as Default button.

Click **Layout** tab

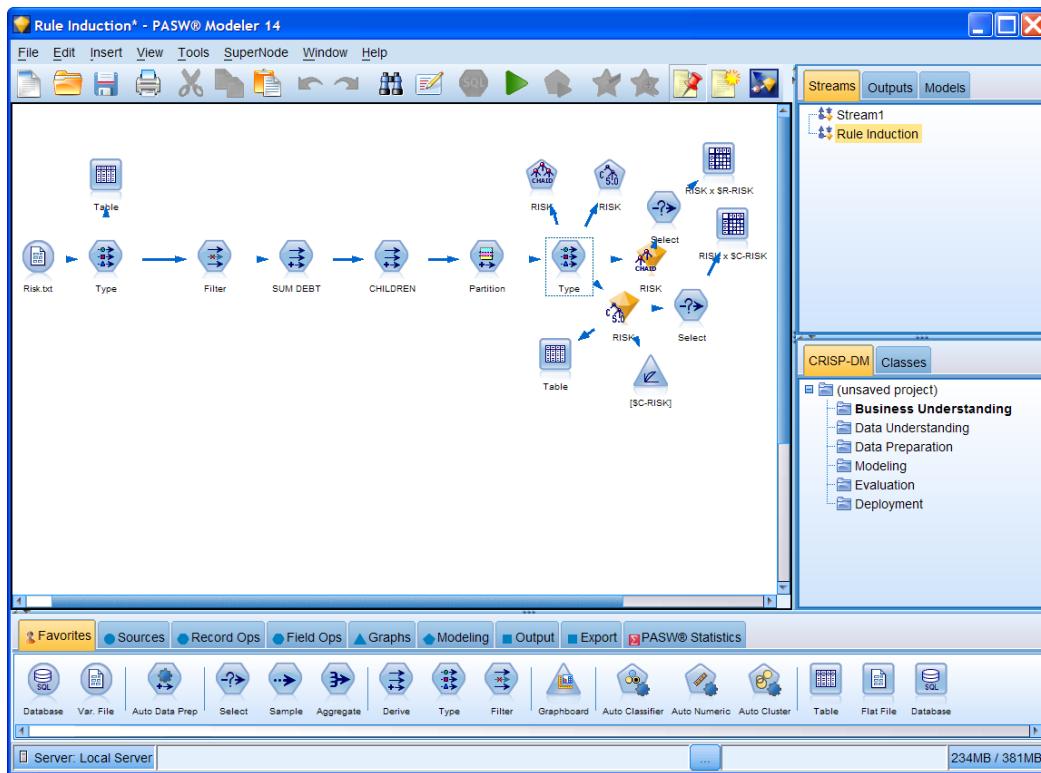
The Layout tab allows you to change options for the display and use of the stream canvas.

The stream canvas has a standard width and height (in pixels). If you create large streams, you will need more space and so can increase these values.

Figure A.8 Layout Options



As a further method to get more nodes on a canvas, you can set the Icon Size setting to Small, which will reduce the size of icons. Figure A.9 shows a stream with small icons.

Figure A.9 Stream with Small Icons

If you find it helpful to name your icons with something that readily describes their function or associated fields, you may want to increase the limit on the number of characters for node names. Of course, longer names will add more clutter to the stream canvas.

Finally, if you want exact control over the position of icons, you can turn off Snap to Grid, which works like it does in drawing programs by placing icons on an invisible grid (note that the grid cell size can also be modified). If you turn off Snap to Grid, though, your streams may not be as neat and tidy.

There are other tabs in the Stream Properties dialog, but they mostly cover more advanced topics, such as scripting.

Appendix B: Running Statistics Commands from PASW Modeler

Objectives

- To provide an introduction to running Statistics commands from PASW Modeler
- To explain differences in how field (variable) definition is handled in PASW Modeler versus Statistics

Data

A data file containing information on the credit rating and financial position of 514 individuals in Statistics format (*smallsample.sav*).

B.1 The Statistics Output Node

Many users of PASW Modeler software also have PASW Statistics installed on their PCs. If these users have had PASW Statistics (formerly known as SPSS® Statistics Base) for any significant period of time, they may have: a.) learned PASW Statistics syntax (its command language), and b.) have PASW Statistics syntax programs saved that could be helpful when analyzing new data with PASW Modeler.

For these users, and for anyone else who would like to use a PASW Statistics procedure (a command that reports on, graphs, or analyzes data) from within PASW Modeler, the Statistics Output node provides this functionality. You can view the results from the Statistics commands in a browser window or save results in the Statistics output file format.

Although PASW Modeler has a wide variety of Graph, Modeling, and Output nodes, there may be certain operations that you either prefer to do in PASW Statistics (e.g., crosstab tables) or others that PASW Modeler doesn't provide (e.g., ordinal regression).

There is some help available in PASW Modeler for PASW Statistics commands, but the Statistics Output node does not provide a dialog-driven interface to create a command. You do need to know PASW Statistics command syntax to use the Statistics Output node.

PASW Statistics must be installed and licensed on your computer to use this node. Otherwise, you don't need to do anything special in PASW Modeler to use PASW Statistics syntax with the Statistics Output Node (to see more details, you can use the Statistics Helper Applications dialog, available from the menus from Tools...Options...Helper Applications).

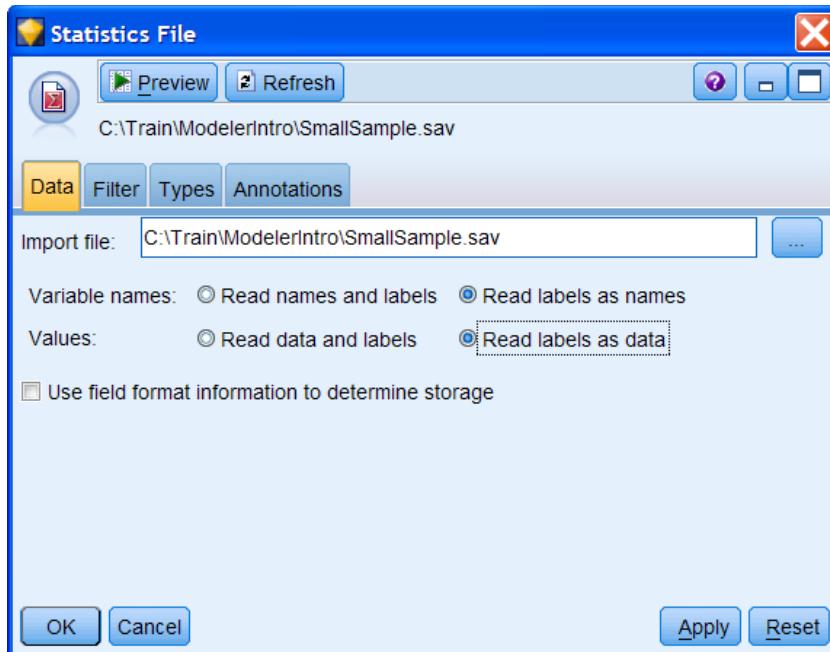
The Statistics Output node is a terminal node, so data cannot pass downstream from the node. As such, this node was designed to be most useful for executing PASW Statistics procedures, but most PASW Statistics commands, including transformation commands such as Compute or Recode, can be run from this node. The modifications made to the data file will only exist within this node and can't be used elsewhere in the stream, but this may often be satisfactory.

To illustrate the use of the Statistics Output node, we'll create a stream that reads a PASW Statistics data file (the data do not need to be from PASW Statistics to use this node) and then look at the options available.

With PASW Modeler open:

- Place a **Statistics File** node on the stream canvas
- Edit the node and specify the import file as **SmallSample.sav** from the c:\Train\ModelerIntro folder
- Click **Read labels as names** and **Read labels as data** option buttons

Figure B.1 Statistics File Node



These choices will create variable names with spaces and odd characters so we can review how these are handled in Statistics. PASW Modeler doesn't complain about the names.

Click the **Preview** button

Notice that some of the field names contain spaces or begin with the pound sign (#). Neither of these is allowed in PASW Statistics, as we will see below.

Figure B.2 Data in Small Sample File

Preview from SmallSample.sav Node (12 fields, 10 records)

The table shows the following data:

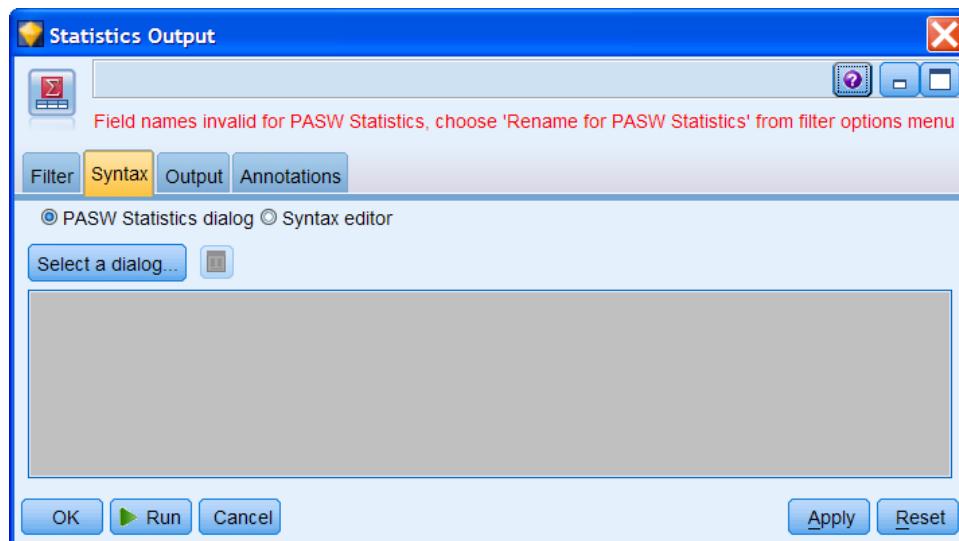
	Income	Gender	Marital Status	# of d...	# of cr...	paid ...	Mort...	# stor...	# oth...	Credit...
1	59193	Female	married	1	2 monthly	y		1	1	good r...
2	58381	Male	married	1	1 monthly	y		1	0	good r...
3	57388	Female	married	0	1 monthly	y		1	0	bad loss
4	56470	Male	married	0	2 monthly	y		1	0	bad loss
5	55554	Female	married	0	1 monthly	y		1	0	good r...
6	54792	Male	married	1	1 monthly	y		2	0	good r...
7	53983	Female	married	1	2 monthly	y		2	0	bad pr...
8	53550	Male	married	1	1 monthly	y		1	1	bad loss
9	52973	Male	married	1	1 monthly	y		1	0	bad pr...
10	52495	Female	married	1	2 monthly	y		1	1	good r...

OK

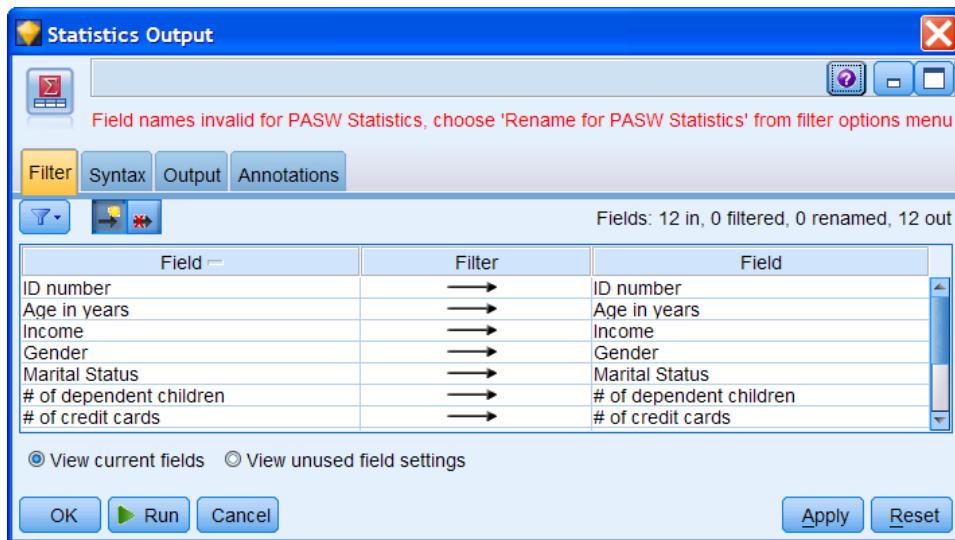
B.2 Using an Existing Syntax File

For an example we will use the syntax file *Appendix B.sps*. Statistics syntax files have the default extension .sps.

- Close the preview window
- Click **OK** to return to the Stream Canvas
- Place a **Statistics Output** node on the stream
- Connect the **Statistics File** node to the **Statistics Output** node
- Double-click the **Statistics Output** node to edit it

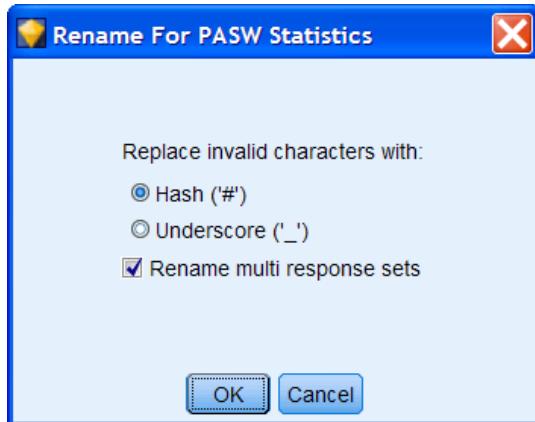
Figure B.3 Statistics Output Node

Click the **Filter** tab

Figure B.4 Filter Tab in Statistics Output Node

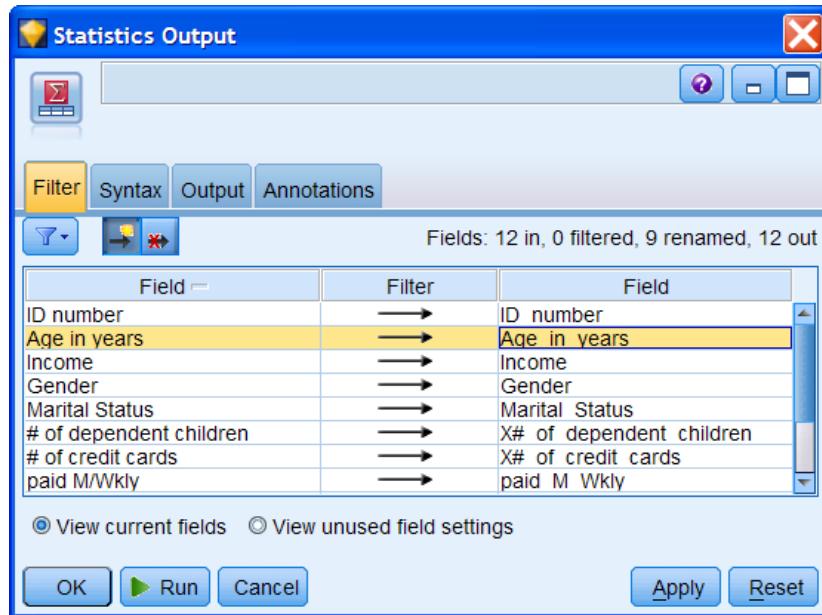
This is a standard Filter tab that you have seen before in Source node dialog boxes. Field names can be edited in the right-hand column as necessary. We will take the easy route and have PASW Modeler do the work.

Click the button and select **Rename for PASW Statistics**

Figure B.5 Rename for PASW Statistics Choices

The invalid characters, such as a space, can be replaced with either a hash mark (#) or an underscore (_). We choose the second option.

Click **Underscore (_)** option button
Click **OK**

Figure B.6 Renamed Fields in Statistics Output Node

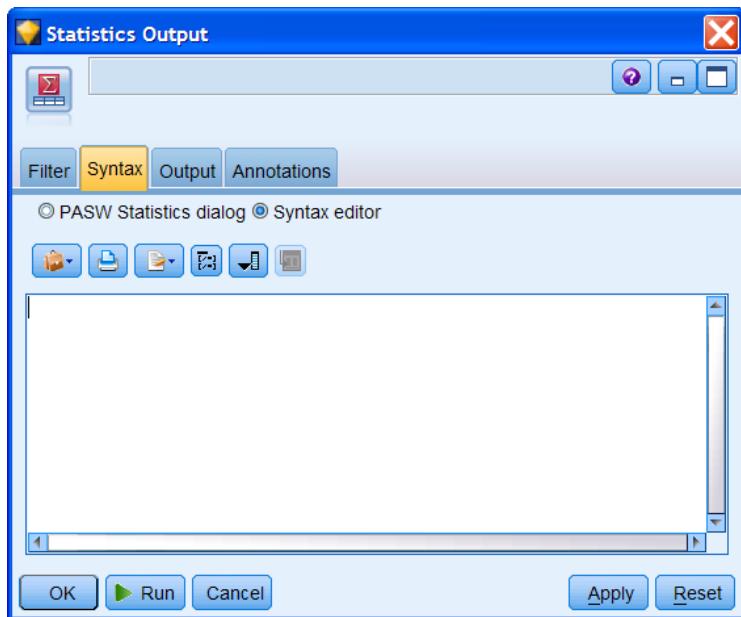
We'll now open the existing syntax file to show how to run some PASW Statistics commands.

Click the **Syntax** tab,
Check the **Syntax editor** button,

This tab provides a text box in which to create the PASW Statistics commands, or to read in commands from a file. There is also a Field Chooser button to select field names to use in a command, and a Statistics Syntax Help button that will provide syntax help for the command at the cursor's location.

To open a syntax file you choose Open from the File options button. To insert previously saved syntax without replacing the current contents, choose Insert from the File menu. This will paste the contents of a syntax file at the point specified by the cursor.

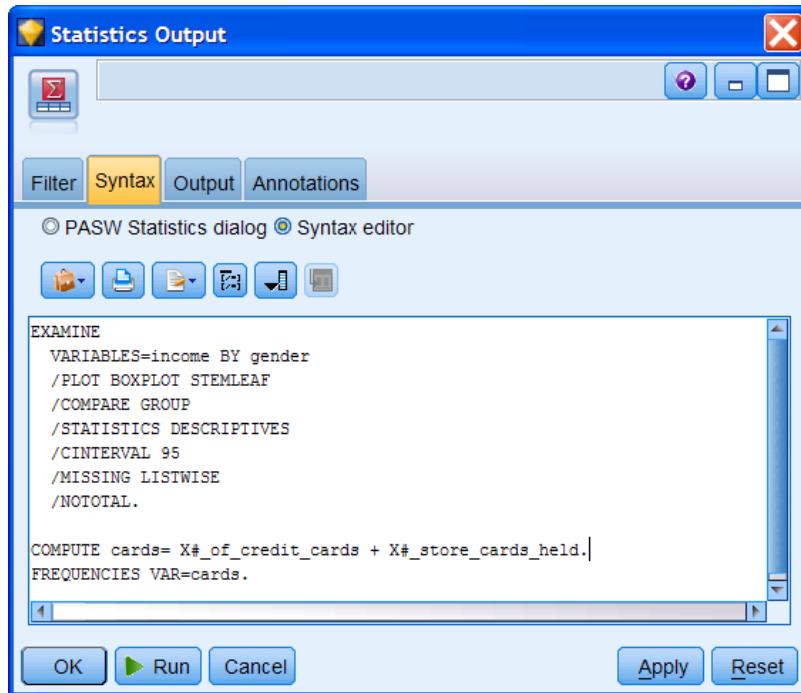
Once you have created syntax here, you can save the syntax into a file by choosing Save or Save As from the File options button.

Figure B.7 Syntax Tab in Statistics Output Node

Click the **File Options** button and then select **Open**
Select the file **Appendix B.sps** to open it

PASW Statistics commands all end with a period. In this simple example, our commands will do the following:

1. Run the Examine command (Explore from the menus) that does data exploration, including the creation of boxplots. The Examine command in Statistics is used in a similar manner to the Data Audit node.
2. Use the Compute command to create a new field (*cards*) that is the sum of the number of credit cards and store cards. Note that because the Statistics Output node is a terminal node, the newly created field will not be available outside of this node.
3. Run a Frequencies command on *cards* to review its distribution.

Figure B.8 Appendix B Syntax File in Statistics Output Node

For those familiar with the Statistics Syntax Editor window, you will recall that only portions of a syntax file need to be run. By comparison, in the Statistics Output node, clicking the Run button will run *all* the syntax.

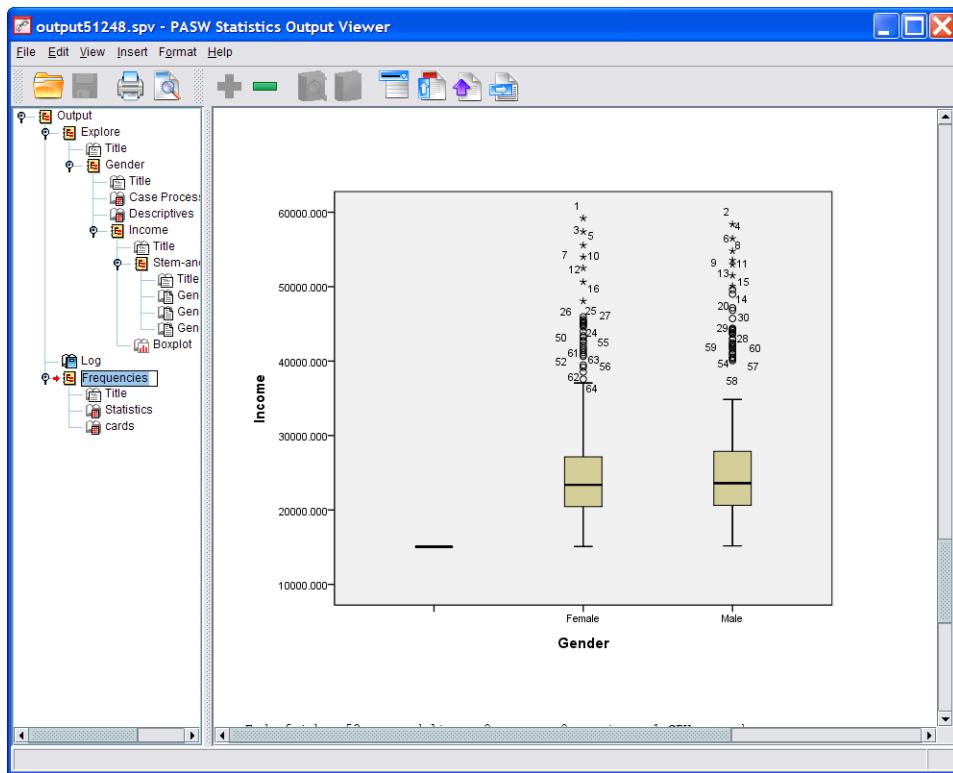
Click Run button

By default, the output from the Statistics Output node is placed into a separate browser window. There is a large amount of output from the Examine command, so we focus only on the Boxplot chart.

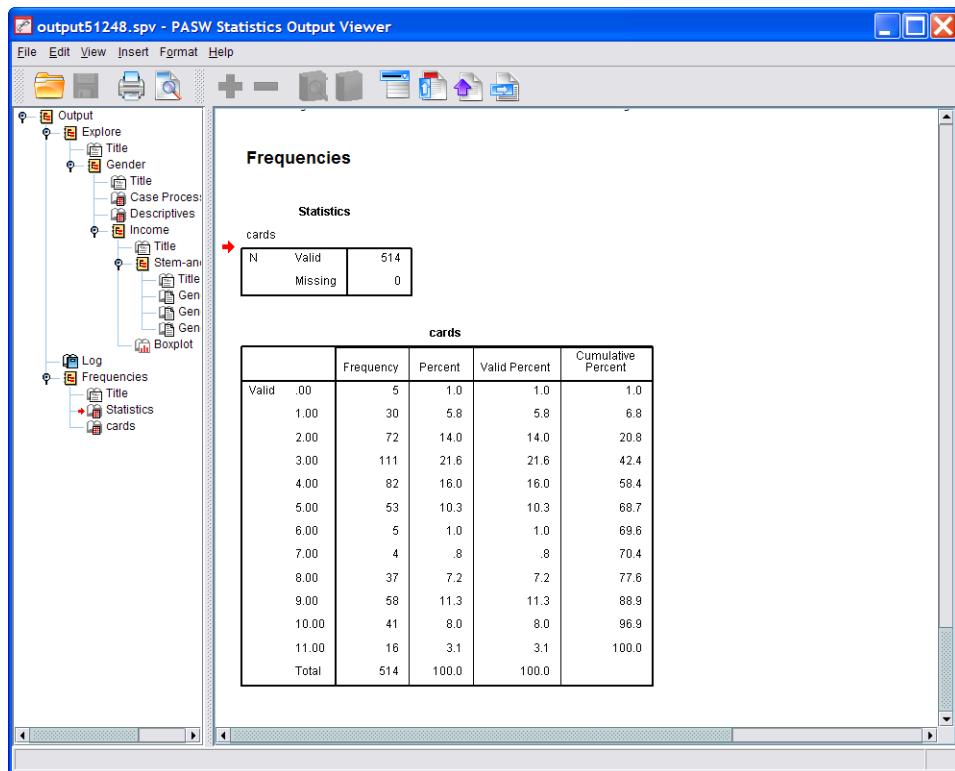
Scroll down until the Boxplot is visible

The boxplot is a succinct method of displaying graphically the distribution of a variable (here *Income*) within the categories of a categorical field (here *Gender*). *Gender* has three categories because of a few blank values (recall that the labels were read as data).

We can see that the boxplots look very similar for males and females, which means the distribution of income is similar for both genders.

Figure B.9 Boxplot from Examine Command

We next computed the new field *cards* and then requested Frequencies for the field. In Figure B.10 we see the standard PASW Statistics Frequencies display, with the first table listing the amount of missing data (see a discussion of this in the next section), and then the frequencies table itself. The range in number of cards owned is from 0 to 11, with 3 being the most common value.

Figure B.10 Frequencies Output for Cards

Missing Data

PASW Modeler and PASW Statistics handle missing data differently. Although both programs allow you to declare data missing, PASW Statistics will usually not honor the missing value designations from the Type tab (in a Source node) or from a Type node. This means you need to be careful when running procedures with PASW Statistics, especially statistical procedures (e.g., Regression), and you should review the number of valid cases carefully.

Typically, the best way to handle missing data is to do so in PASW Modeler before the Statistics Output node with the usual methods, such as filtering out the missing records. However, you can use the PASW Statistics Missing Values command in the list of syntax to handle missing data on the fly.

Data Mining References

Berry, Michael J. and G. Linoff. 2004. *Data Mining Techniques For Marketing, Sales and Customer Support*. (2nd edition). New York: Wiley.

Berry, Michael J. and G. Linoff. 2001. *Mastering Data Mining: The Art and Science of Customer Relationship Management*. New York: Wiley.

Berry, Michael J. and G. Linoff. 2002. *Mining the Web: Transforming Customer Data into Customer Value*. New York: Wiley.

Berson, Alex and S. J. Smith. 1997. *Data Warehousing, Data Mining, & OLAP*. New York: McGraw Hill.

Fayyad, Usama M., Piatetsky-Shapiro, G., Smyth, P. and R. Uthurusamy. 1996. *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA.; Cambridge, Mass: AAAI Press and MIT Press.

Han, Jiawei and M. Kamber. 2005. *Data Mining: Concepts and Techniques*. (2nd ed.). San Francisco: Morgan Kauffman.

Hand, David J. 1998. "Data Mining: Statistics and More?" *The American Statistician*. Vol. 52, No. 2.

Heikki, Mannila, Smyth, P. and D. Hand. 2001. *Principles of Data Mining*. Boston: MIT Press.

Larose, Daniel T. 2004. *Discovering Knowledge in Data: An Introduction to Data Mining*. New York: Wiley –Interscience.

Mena, Jesus. 2001. *Webmining for Profit: E-Business Optimization*. Burlington, MA: Butterworth-Heinemann.

Pyle, Dorian. 2003. *Business Modeling and Data Mining*. San Francisco: Morgan Kaufmann.

Westphal, Christopher and T. Blaxon. 1998. *Data Mining Solutions*. New York: Wiley.

Version 1.0 of the CRISP-DM (Cross-Industry Standard Process for Data Mining) process document can be downloaded from the following website: www.crisp-dm.org.

IBM