Visualization for Classification and Clustering Techniques

Overview

- Importance of Data Visualization in the Knowledge Discovery and Data Mining (KDD) Process
- Understanding and Trust
- Visualization techniques
  - Classification
  - Clustering
- Future Directions

KDD Process

- Selection
  - Obtain data from all of sources
- Preprocessing
  - After selecting the data, clean it to make sure it is consistent
- Transformation
  - After preprocessing the data, analyze the format/amount of data
- Data Mining
  - Once the data is in a useable format/content, apply various algorithms based upon the results trying to be achieved
- Interpretation/Evaluation
  - Finally, present the results of the data mining step to the user, so that the results can be used to solve the business need at hand

Importance of Data Visualization

The final step in the KDD process :

- Highly dependent on the Data Visualization technique
- Bad/inappropriate technique may result in misunderstanding
- Misunderstanding may cause an incorrect (or no) decision

It is important to consider that the KDD process is useless if the results are not understandable

Current Issues w/Data Visualization

- The literature suggests a significant reliance on expert users
- General lack of data visualization support in many data mining tools [Goebel99]
- These are significant problems if KDD/DM/Data Visualization will expand at the rates suggested
  - Data visualization tool market – $2.2 billion by 2007 [Nuttall03]

Suggested Direction

- Need to determine techniques that balance simplicity with completeness
- If this can be done for non-expert users
  - Simplicity & Completeness → Understanding
  - Understanding → Trust
  - Trust → more use of KDD/DM
  - Result will be:
    - Better business value
    - Higher ROI

Classification and Clustering

- Classification/clustering are classical pattern recognition/ machine learning problems
- Classification, also referred to as categorization
  - Asks "what class does this item belong to?"
  - *Supervised learning* task
- Clustering
  - Asks "how can I group this set of items?"
  - *Unsupervised learning* task
- Items can be documents, emails, queries, entities, images
- Useful for a wide variety of search engine tasks

Common Visualization Techniques

- Visualization techniques dependent upon
  - The type of data mining technique chosen
  - The underlying structure and attributes of the data

Classification

- Decision Trees
- Scatter Plots
- Axis-Parallel Decision Trees
- Circle Segments
- Decision Tables

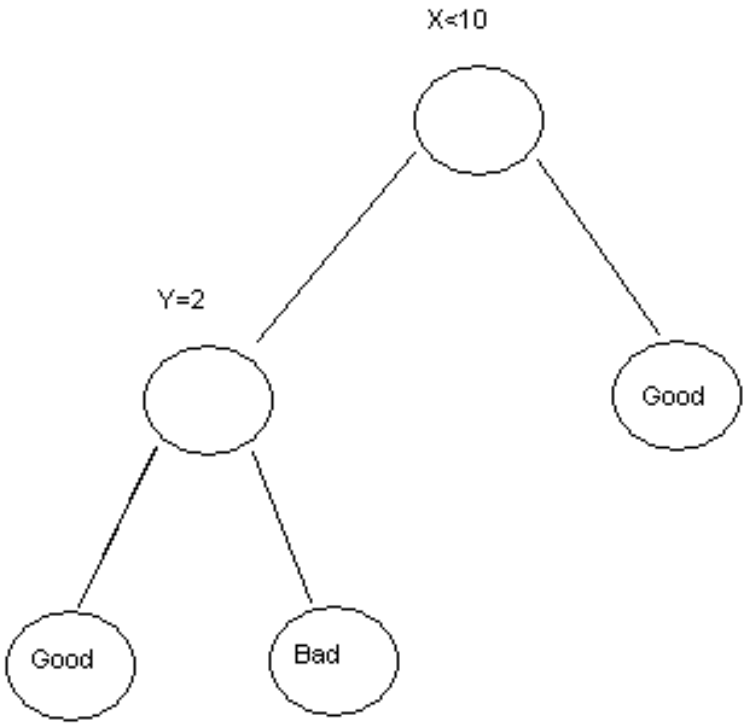
Clustering

- Scatter Plots
- Dendrograms
- Smoothed Data Histograms
- Self-Organizing Maps
- Proximity Matrixes
- K-means

Classification

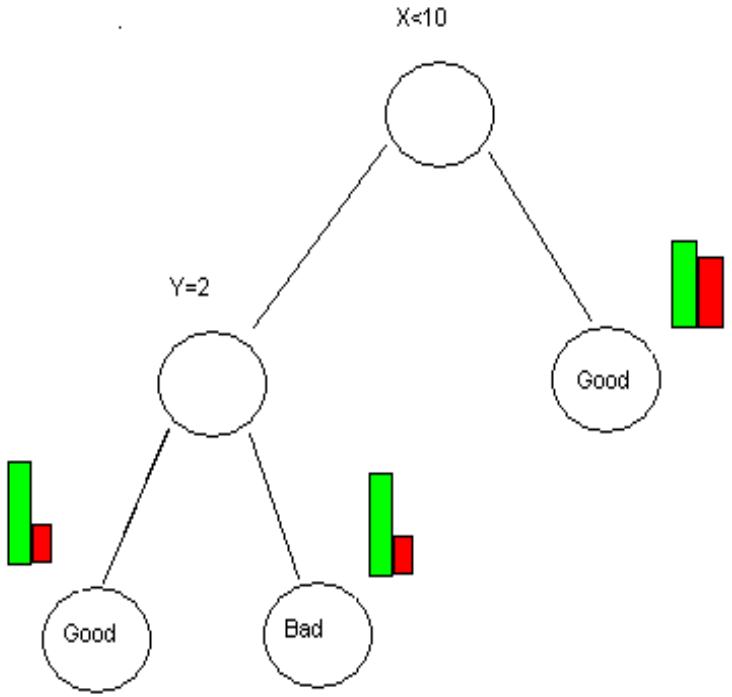
Decision Tree

- Information limited to
  - Attributes
  - Splitting values
  - Terminal node class assigr



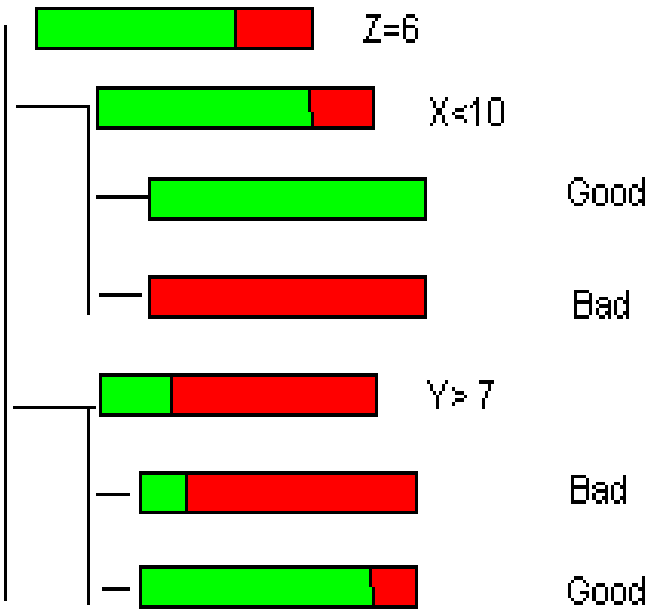
Decision Tree with Histograms

- Data mining rarely classify 100% of the data correctly:
  - Include the success of properly classifying the data - histogram added for each terminal node
  - Percentage of data that was classified correctly/incorrectly
  - Assists users in determining if the classification is ‘good enough’



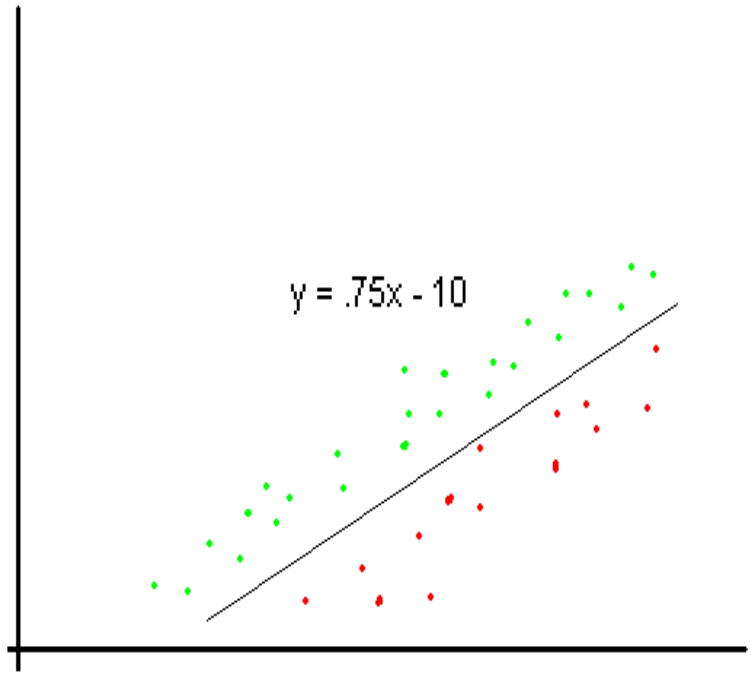
Decision Tree - Different Format

- Vertical representation - allows for easy user interaction
  - Combines the split points ar classification accuracy compactly
  - Key difference - colors a matched with a specif classification



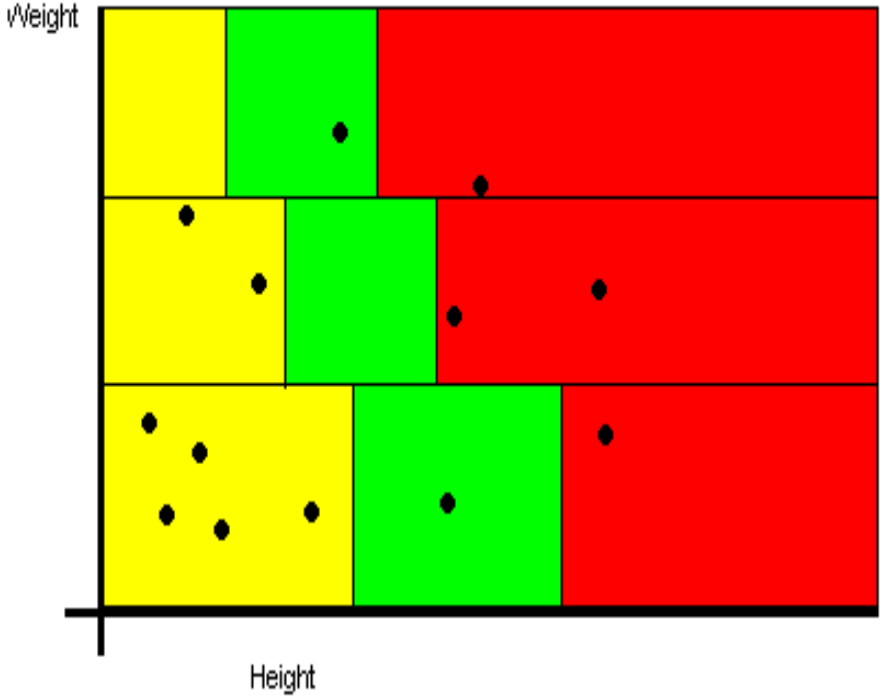
Scatter Plot with Regression Line

- Excellent way to view 2-dimensional data
- Familiar to anyone who has taken high-school algebra
- Regression lines provide descriptive techniques for classification



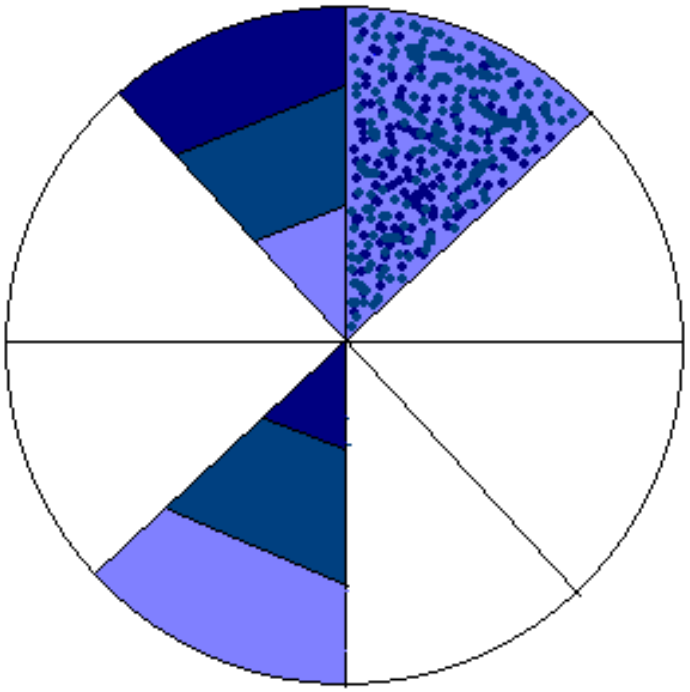
Axis-Parallel Decision Tree

- Combination Scatter Plot and Decision Tree
- Areas divided in parallel regions on the axis
- Well suited for classification problems with two attribute values
- High visibility into the impact of outliers



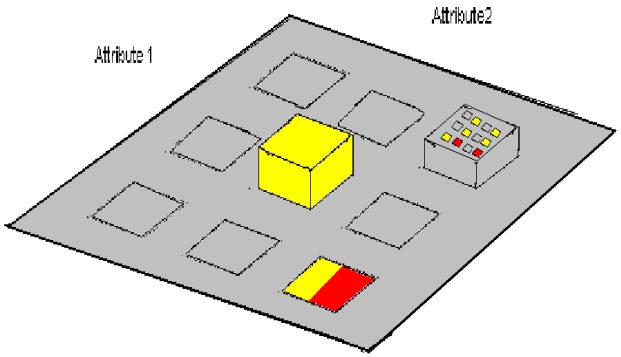
Circle Segments

- Multi-dimension data
- Maps dataset with *n* dimensions onto a circle divided by *n* segments
  - Each segment is a different attribute
  - Each pixel inside a segment is a single value of the attribute
  - Values of each attribute are then sorted (independently) and assigned a different colors based upon its class

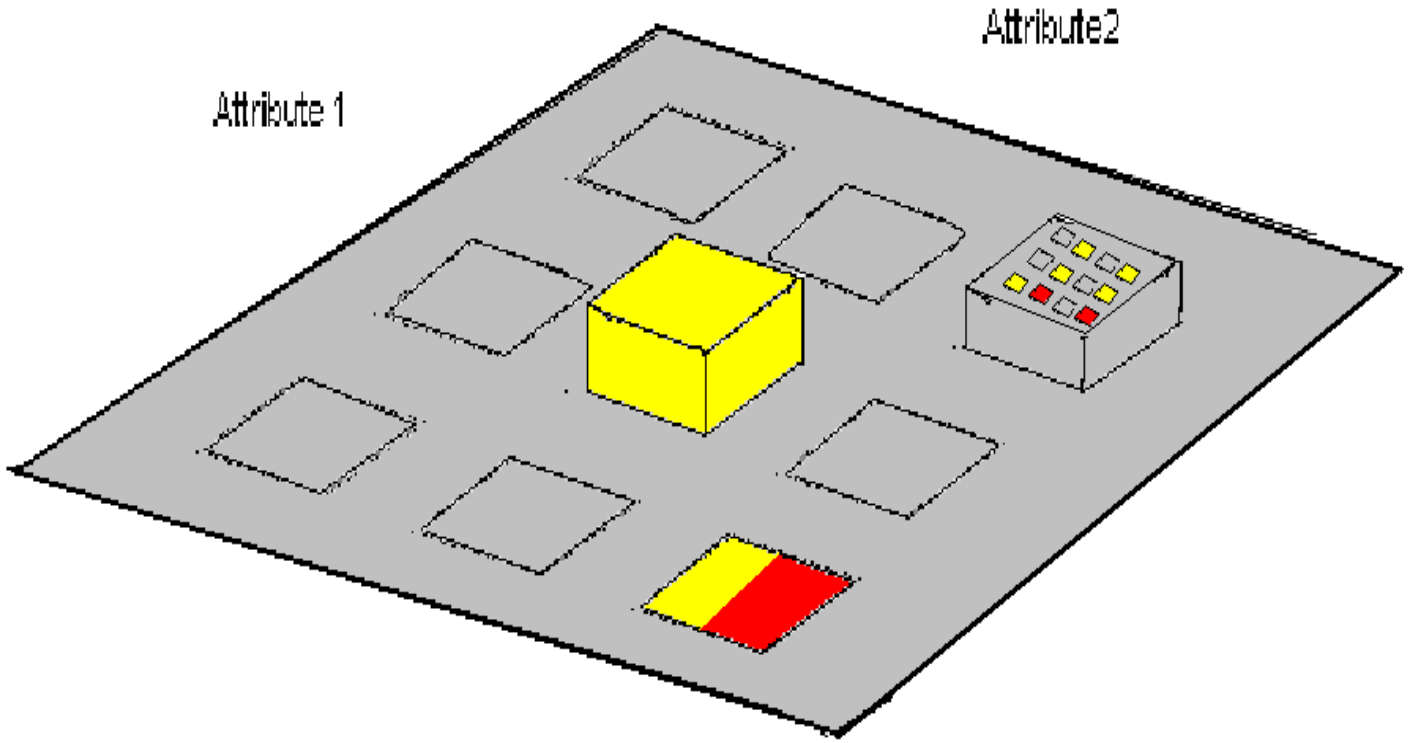


Decision Table

- Interactive technique
- Maps attribute data to a 2D hierarchical matrix
- Levels can be drilled down - another set of attributes
- Height of a cell conveys the number of data entities
- Cells color coded
  - Neutral color → no data in that intersection point
  - Color coded by class (percentage)



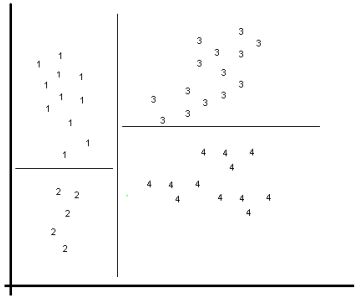
Decision Table



Clustering

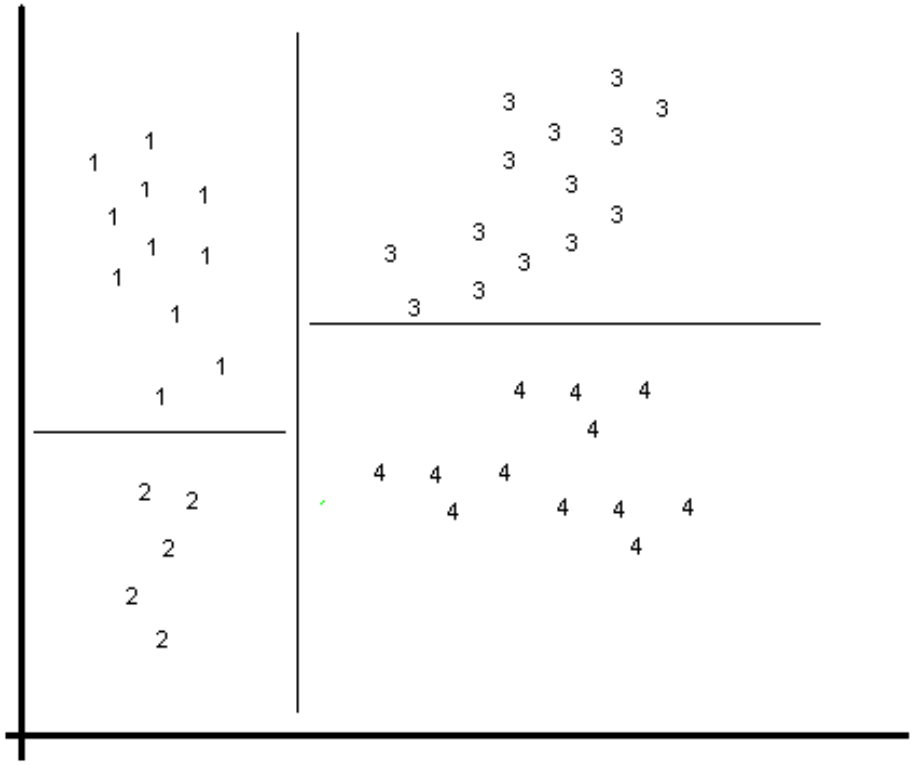
Scatter Plot

- Extensions include, displaying points in:
  - Various sizes and colors to indicate additional attributes
  - Shading of points to introduce a third dimension
  - Using different brightness levels of the same color to represent continuous values for the same attribute
  - Using various points or classification identifiers (i.e., numbers, symbols)
  - Using various glyphs to display additional attributes

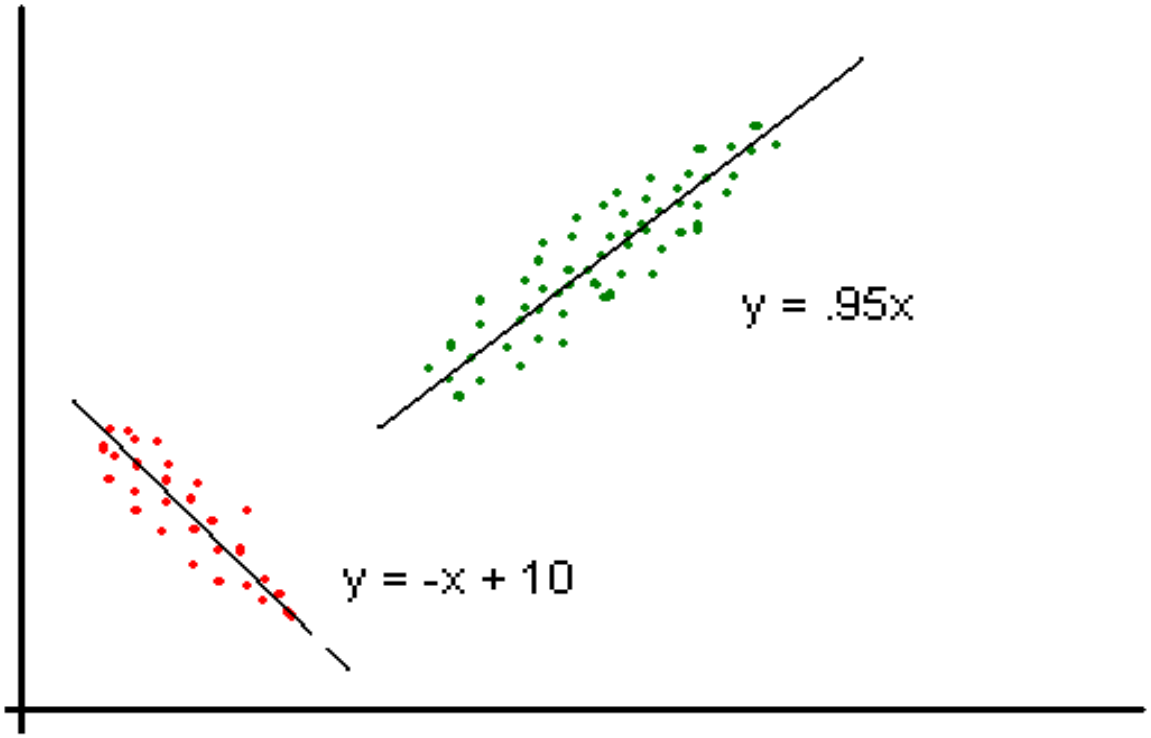


Scatter Plot

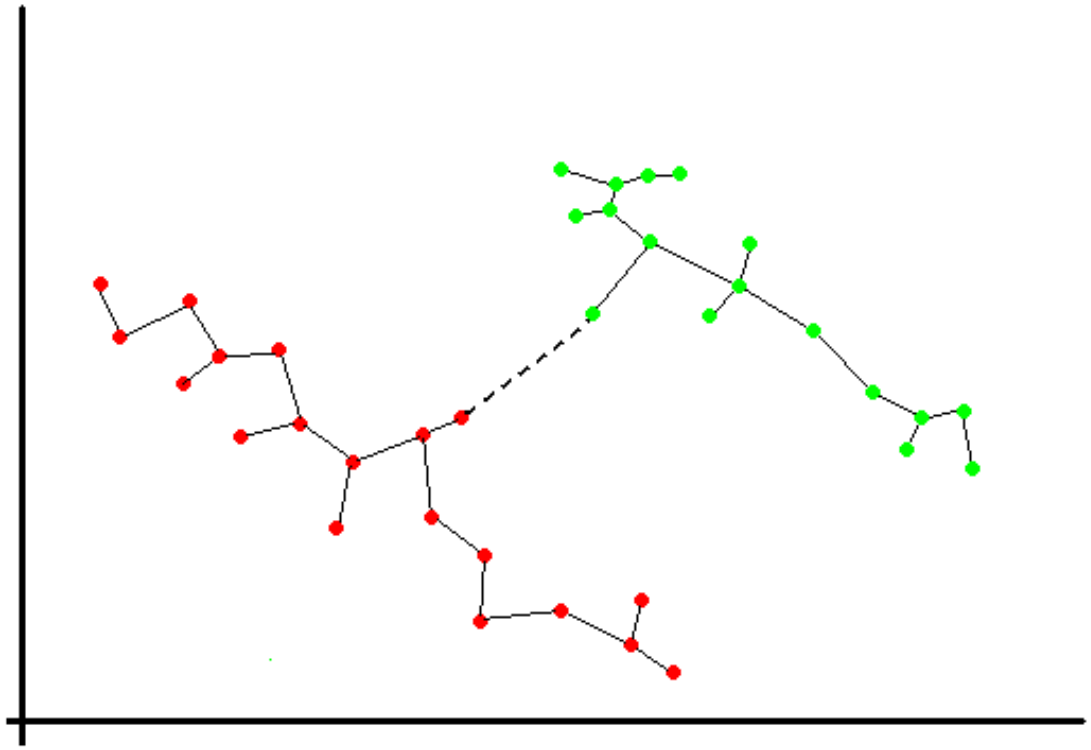
- Map decision trees on top of scatter plots to describe clusters



Scatter Plot with Regression Lines

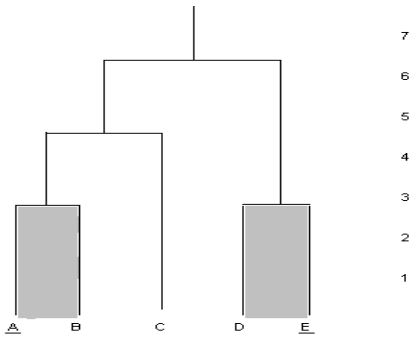


Scatter Plot w/Min Spanning Tree



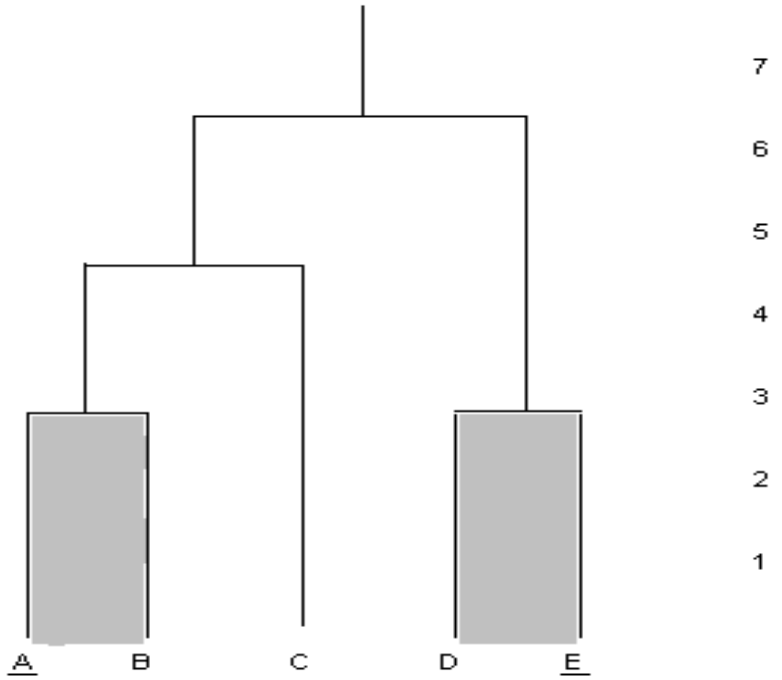
Dendrogram

- Intuitive representation - hierarchical decomposition of data into sets of nested clusters.
- From an agglomerative perspective:
  - Each leaf - a single data entity
  - Each internal node - the union of all data entities in its sub-tree
  - The root - the entire dataset
  - The height of any internal node - the similarity between its ‘children’.



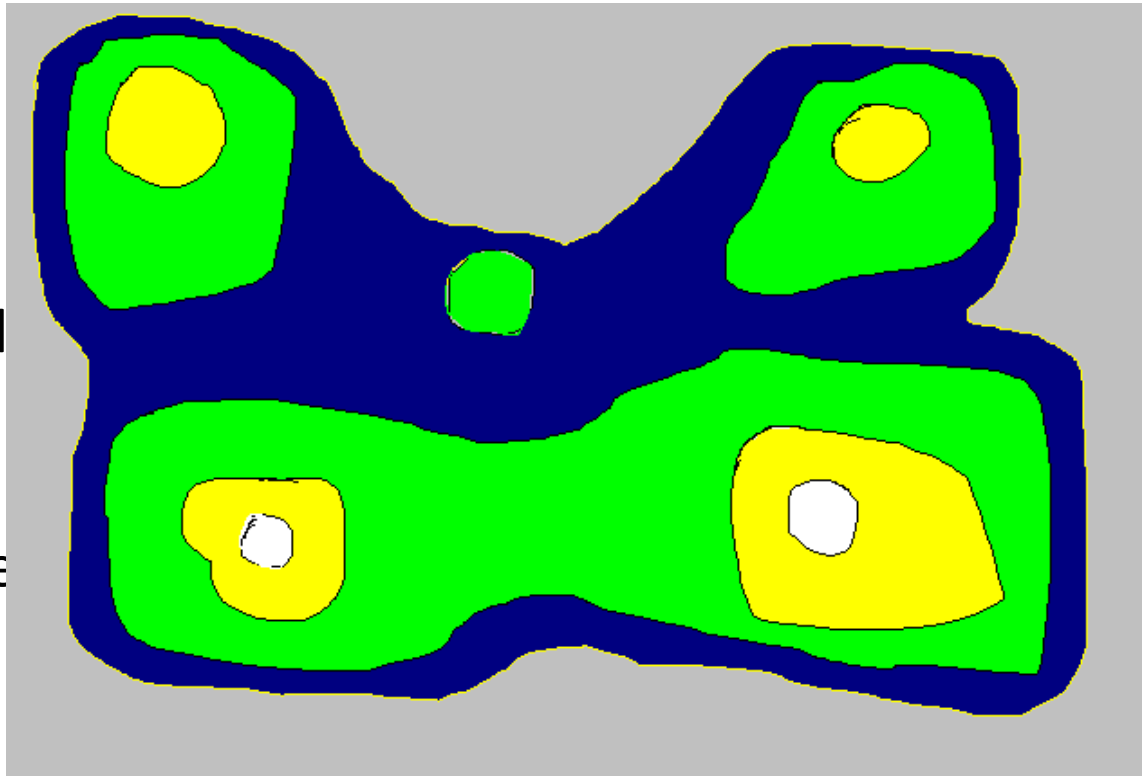
Dendrogram with Exemplars

- The “most typical member of each cluster” [Wishart99]
  - Underlined labels of the leafs
  - Done in combination with shading to identify the clustering level

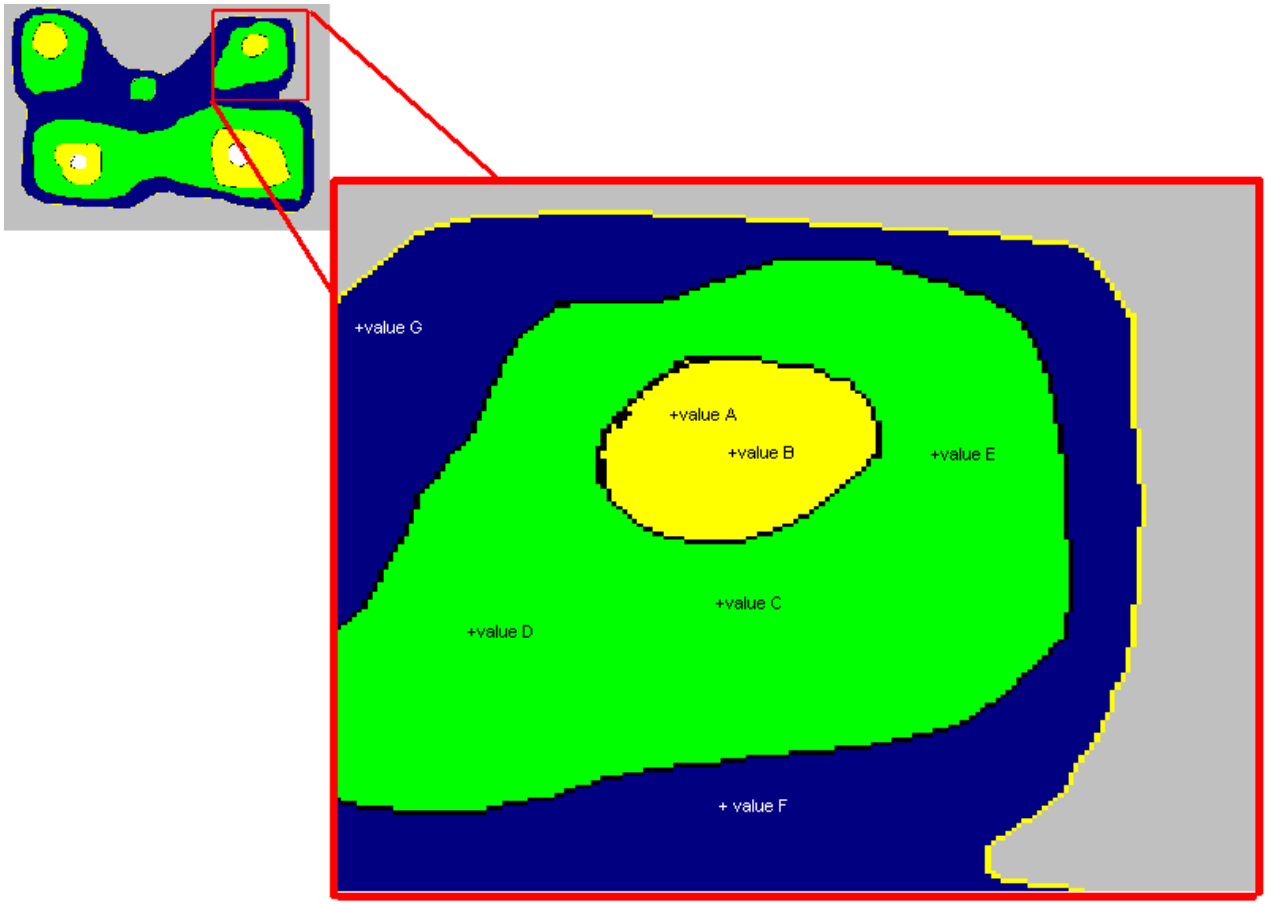


Smoothed Data Histogram

- Represents data on a ‘display map’
- Similar data items are located close to each other
- More defined the clusters – lighter colors



Smoothed Data Histogram - Detail



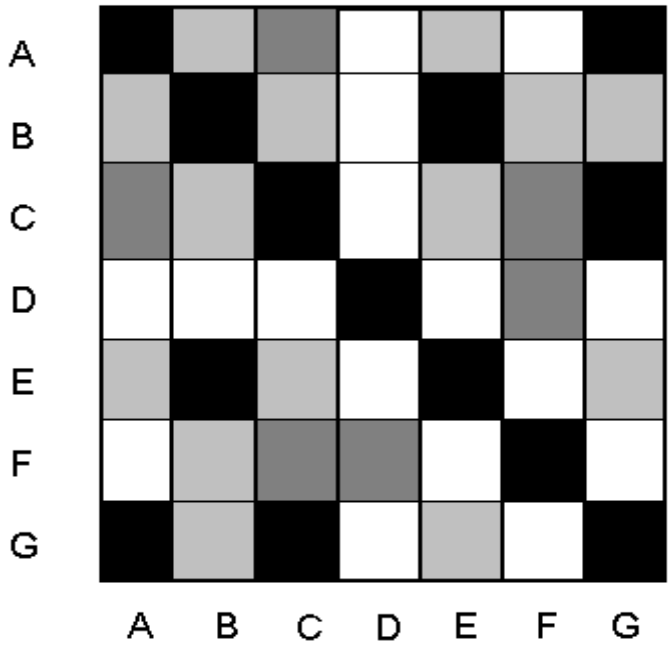
Self-Organizing Map ‘Grid’

- Source of Smoothed Data Histogram
- Numbers indicate most ‘common’ cluster

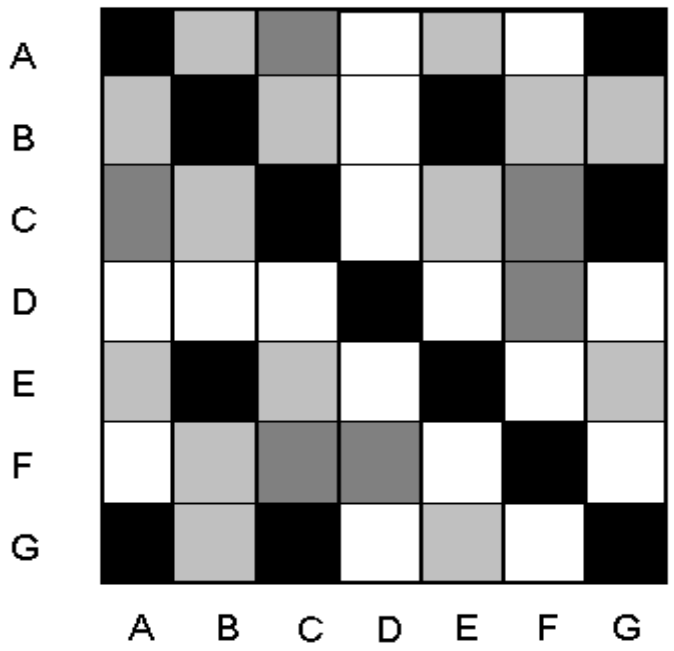
1					*5*	
2	*3*	*2*		*5*	*6*	*5*
2	*2*	*2*	*4*	*5*	*5*	*5*
7	*1*	*1*	*1*	*5*	*7*	
7	*8*	*7*	*7*	*7*	*10*	*7*
7	*9*	*7*	*7*		*11*	*7*
	8			*7*	*10*	*7*

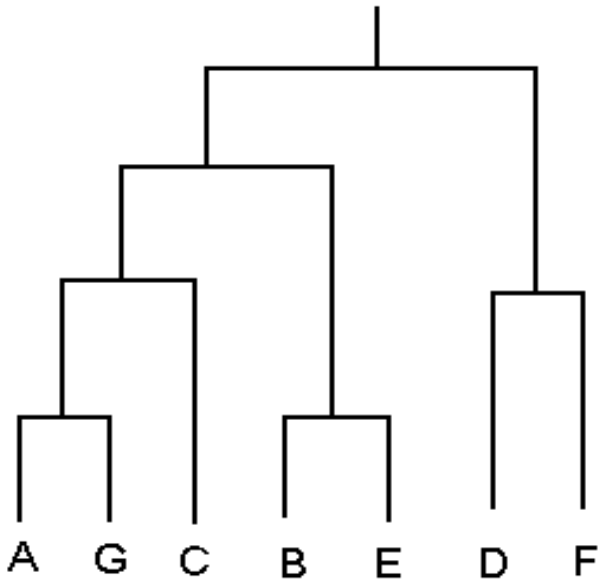
Proximity Matrix

- Graphically display the relationship between data elements
- Usually symmetric, but can be sorted by the strength of relationships



Proximity Matrix and Dendrogram





Summary

- Data visualization techniques are extremely important for understanding the KDD process
- A balance of simplicity and completeness is important
- The techniques discussed allow average users to understand the results of the KDD process
- Understanding → KDD results to be interpreted/trusted by non-expert users → extending the business value
- If data visualization techniques do not establish a high level of trust in the KDD process, the process will fail

Future Direction

- Significant effort will be spent on improving data visualization techniques in the next few years
  - KDD process and data mining are becoming more widespread
  - Business will expect tools to become more ‘user-friendly’ and support the varied level of skills
- Trends are moving to a more interactive mode
  - Static reporting techniques (i.e., standard decision trees, standard circle segments) are being replaced
  - Interactive techniques (i.e., smoothed data histograms, interactive circle segments and decision tables)
- Very interactive data models → ‘virtual reality’ are also being considered/proposed

Advanced Analytics: Trends, Forecasts, Clusters, and other Statistical Tools

In Tableau Desktop, you can run statistical analyses with only a few clicks. Tableau's advanced analytics tools include distribution bands, trend lines, forecasts, and clustering. With these tools, you can tackle analytical questions that cannot be answered with simple descriptive visualizations. But they don't return simple summary tables as results; instead, as Tableau is primarily a visual analytics tool, the output is added to the charts you create.

In addition to the built-in analytics tools, it is possible to integrate Tableau with different programming languages for more sophisticated, customized analytics projects. This integration of Python, R, and MATLAB will be covered in the second half of this chapter.

TREND LINES

Line charts can sometimes be difficult to interpret when the data is very granular. Trend lines help you to see the pattern in your data, by tracing out the fundamental evolution of the measure in question.

The following trend line models are available:

- Linear
- Logarithmic
- Exponential
- Polynomial
- Power

Adding Trend Lines

To add a trend line, drag the Trend Line item from the Analytics pane to your view. While you are moving the mouse, Tableau will show a selection of the available types of regression models (see Figure 7.2).

Figure 7.2 Selection fields with different types of trend line models.

Choose the model type that is most appropriate for your use case by dropping the Trend Line on the respective field. Models that are not compatible with your data are greyed out and can't be selected.

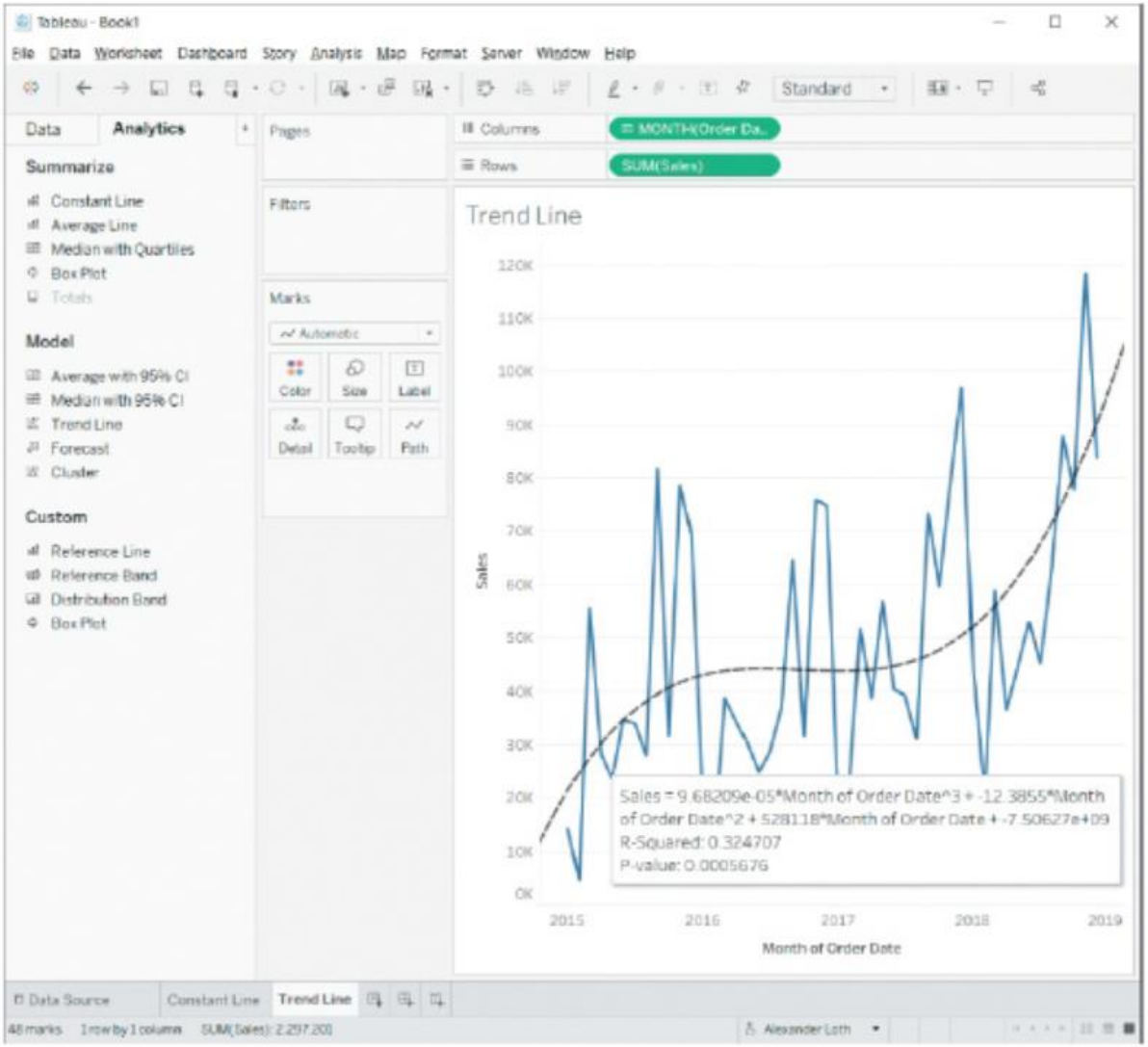


Figure 7.3 Line chart with an added trend line and its tooltip.

Figure 7.5 Description of a trend line.

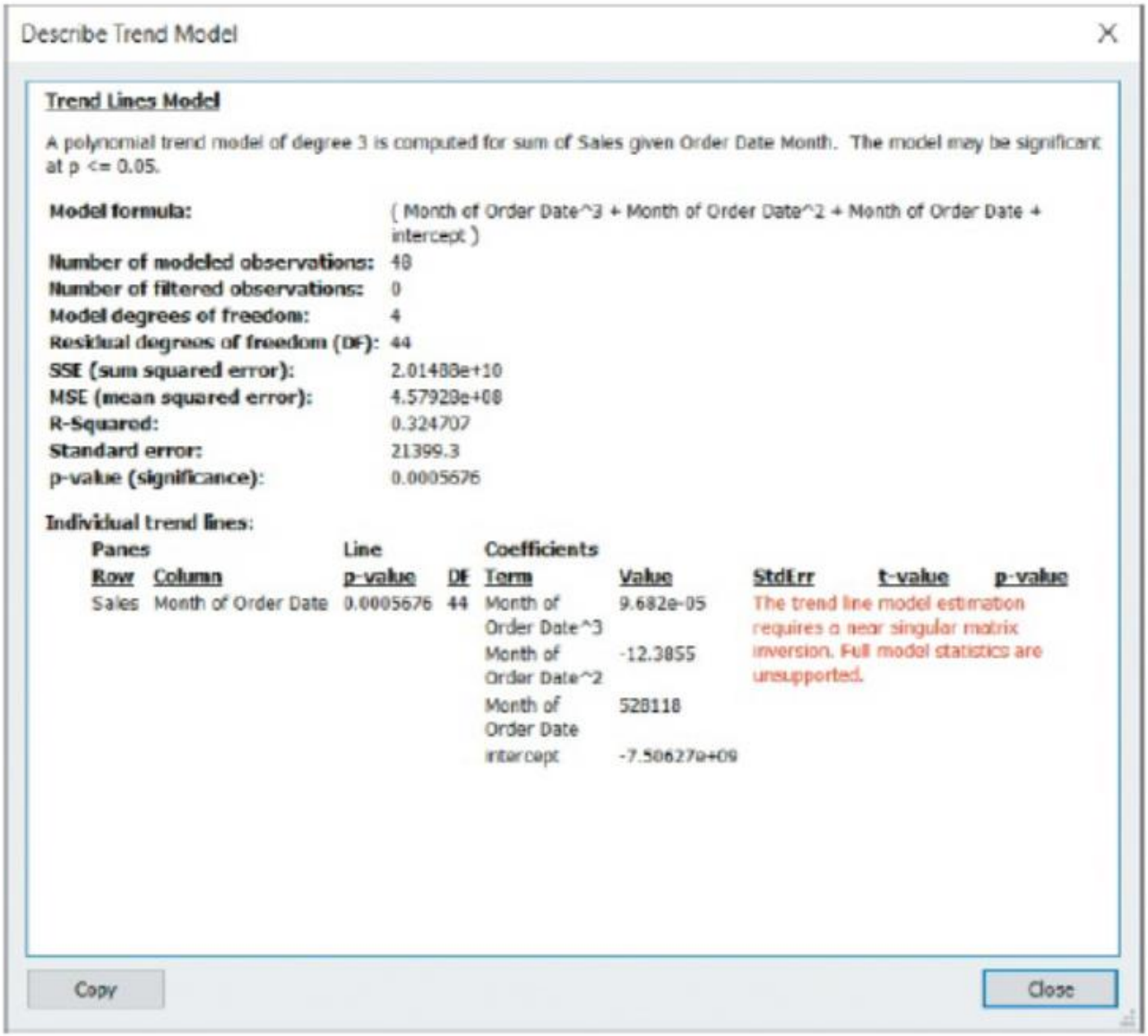


Figure 7.6 Trend line model.

FORECASTS

Forecasting models extrapolate future values of a time series based on its historical values, allowing you to attempt to predict the evolution of a measure. Many different mathematical models can be used for such endeavors, each with its own advantages and drawbacks. Tableau's forecasting tool uses what is called *exponential smoothing*.

In such models, more recent data points are assigned greater weights than older observations. They work reasonably well to capture both long-term trends and any potential seasonality that may be present in the time series. The resulting forecast is shown directly in the chart.

Because we are talking about time-series data, ensure that your view contains a time or date field as well as a measure.

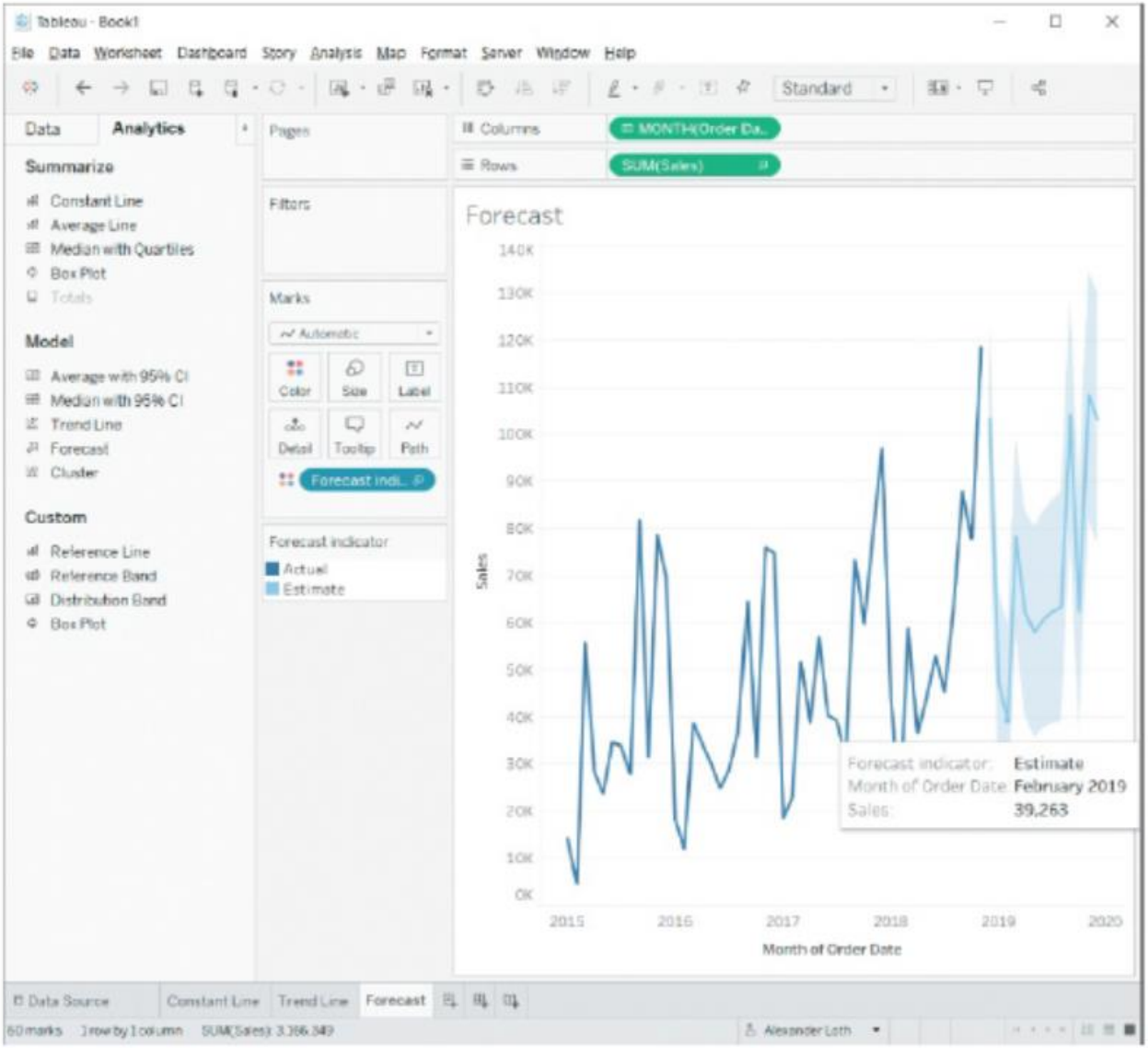


Figure 7.7 Line chart with a forecast.

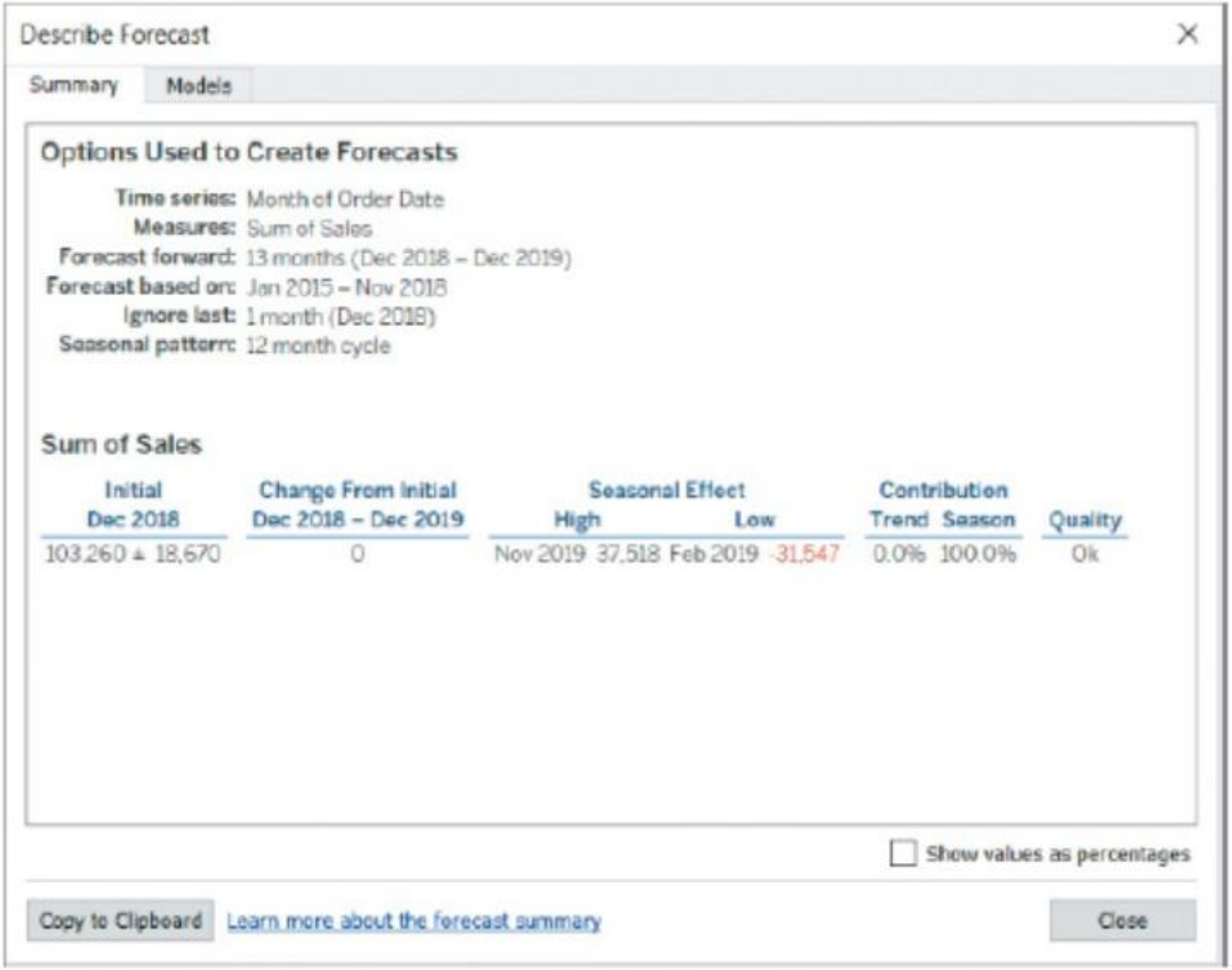


Figure 7.9 Model summary.

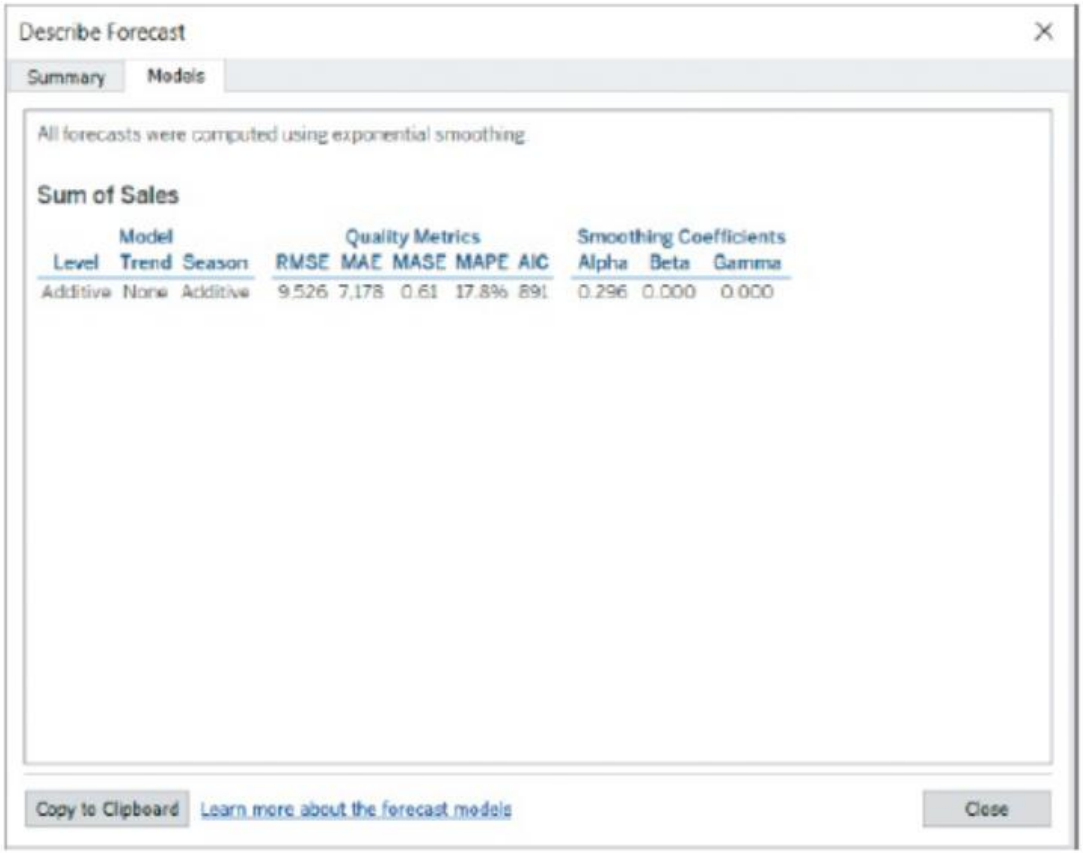


Figure 7.10 Model type, quality metrics, and smoothing coefficients.

The forecast models used by Tableau assign greater weight to more recent observations. The smoothing coefficients determine the extent to which this happens. A value close to 1 indicates that only recent values influence the forecast. A value of 0 means all historical data enters the equation equally (maximum smoothing). The alpha coefficient refers to the level forecast, the beta coefficient to the trend forecast, and the gamma coefficient to the forecast of the seasonality.

CLUSTER ANALYSIS

A *cluster* is a collection of data points with similar properties. Cluster analysis helps you find such groupings in your data. A classic use case comes from the field of marketing, where clustering is often used to define different customer segments.

The clustering tool in Tableau uses the widely used k-means algorithm, a type of vector quantization developed originally in the field of signal processing. Simply put, the method works by assigning *n* number of observations to *k* number of clusters, so that each observation is part of the cluster with the nearest cluster mean.

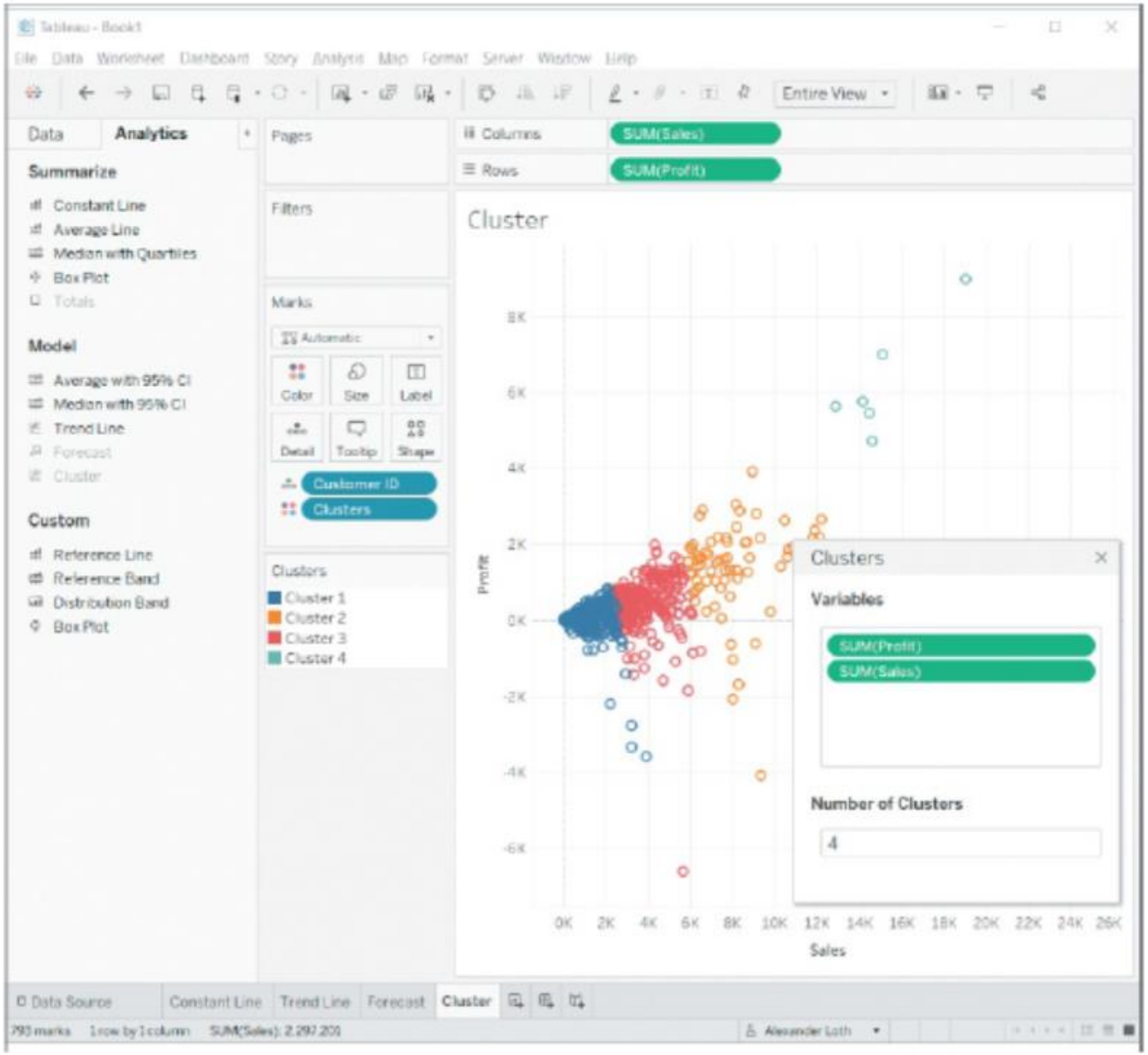


Figure 7.11 Customer segments based on sales and profits.

PYTHON, R, AND MATLAB INTEGRATION

As you have seen, the Analytics pane offers quick access to a handful of useful analytics tools. However, if you require deeper statistical analysis of your data to complement Tableau's visual analytics approach, it is often necessary to use one of the popular data science tools: Python, R, or MATLAB. Beginning with the 2013 version, it has been possible to use Tableau together with the programming language R, meaning you can add calculated fields with R scripts to the view to visualize the results of calculations run in R. In subsequent years, Python and MATLAB integrations were added to provide similar functionality with these two tools. As you can imagine, this opens up a world of advanced statistical analyses that can be performed with your data.

Tableau supports the integration of these three services:

R R is an open source programming language widely used in the scientific community because of its many packages for statistical analysis and the creation of statistical charts.

MATLAB MATLAB is a software package with a mathematics-focused syntax. It is often used in signal processing, in testing and measurement processes, in financial modelling, and in the field of bioinformatics.

Python Python is a popular general-purpose programming language used both in academia and in many business applications. Python includes a number of statistical and machine learning tools out of the box and, like R, can be extended by adding modules created by the Python community.

Trellis Chart with Python Script

Let's use this field to create the trellis chart shown in Figure 7.18. Start by creating a scatterplot with `Profit` and `Sales` on the two axes and with `Customer ID` on Detail. Then add the dimensions `Region` and `Sub-Category` to Rows and Columns, respectively. This gives you a grid with several smaller scatter plots showing how profits and sales by customer are related to each other.

Figure 7.18 Trellis chart, with trend lines and colors according to the correlation coefficient.