Multidimensional Visual Analysis

A Survey of Multidimensional Visual Analysis Methods and Tools

Ožbej Golob

Graz University of Technology

27 May 2023

Abstract

With the rise of artificial intelligence and machine learning, there is an increasing amount of high-dimensional data available. Multidimensional Visual Analysis (MVA) uses visual representations to explore and analyze multi-dimensional datasets.

This survey reviews multiple popular MVA approaches, including scatterplots, scatterplot matrices, Star Coordinates, RadViz, Dust and Magnet, similarity maps, parallel coordinates, brushing, linking, grouping and labeling. The survey also reviews and compares some of the most popular MVA software tools, including XMDV, Parallax, GGobi, InfoScope, XDAT, High-D, TabuVis, Improvise, MyBrush, and mVis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multidimensional Visual Analysis (MVA) focuses on the use of visual representations to explore and analyze multidimensional datasets. This approach aims to provide a more intuitive and interactive way to understand and extract insights from large and complex multidimensional datasets. One of the main challenges in MVA is the effective representation of high-dimensional data in a way that is meaningful and intuitive for the user. To address this challenge, a wide range of visual encodings and interaction techniques have been developed.

One of the key benefits of MVA is its ability to reveal patterns and relationships in the data that may not be apparent through traditional statistical analysis methods. This is particularly useful in the exploratory phase of data analysis, where the aim is to gain a better understanding of the data, identify patterns, trends, and outliers, and determine potential areas of interest for further investigation. In addition to its use in exploratory data analysis, MVA also has a range of applications in areas such as data mining, machine learning, and business intelligence.

Multidimensional datasets are usually represented as a table or a spreadsheet, where each column represents one dimension (variable) and each row represents one record (data point, instance). Throughout this survey, three multidimensional datasets will often bes used for illustration:

1. *Iris*: The Iris dataset [Fisher 1936] is arguably the best known dataset in pattern recognition. The dataset contains 150 records belonging to three classes of Iris flower species (each class has 50 records). Each record has the following dimensions: sepal length in cm, sepal width in cm, petal length in cm, petal width in cm, and variety. Each record belongs to one of three classes (varieties): Setosa, Versicolour, or Virginica. Setosa is linearly separable from the other two; Versicolour and Virginica are not linearly separable from each other. Table 1.1 shows the first few rows from the Iris dataset.

2. *Palmer Penguins*: The Palmer Penguins dataset [Gorman et al. 2014; Horst et al. 2020] contains 342 records belonging to three classes of penguin species from Antarctica (from Torgersen, Biscoe, and Dream islands). Each record has the following dimensions: island, bill length in mm, bill depth in mm, flipper length in mm, body mass in g, sex, year, and species. Each record belongs to one of three classes (species): Adelie, Gentoo, or Chinstrap. Table 1.2 shows the first few rows from the Palmer Penguins dataset.

3. *Premier League Player Stats*: The Premier League Player Stats dataset [Samariya 2020] contains 540 records of Premier League football player statistics. Each record has the following dimensions: player (name and surname), team, games played (GP), games started (GS), minutes played (MIN), goals (G), assists (ASST), total shots (SHOTS), and shots on goal (SOG). Table 1.3 shows the first few rows from the Premier League Player Stats dataset.

This survey reviews a number of popular MVA approaches in Chapter 2 and several popular MVA software tools in Chapter 3.

	Sepal Length	Sepal Width	Petal Length	Petal Width	Variety
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
...					

Table 1.1: Snippet of the Iris dataset.

	Island	Bill Length	Bill Depth	Flipper Length	Body Mass	Sex	Year	Species
0	Torgersen	39.1	18.7	181.0	3750.0	male	2007	Adelie
1	Torgersen	39.5	17.4	186.0	3800.0	female	2007	Adelie
2	Torgersen	40.3	18.0	195.0	3250.0	female	2007	Adelie
3	Torgersen	36.7	19.3	193.0	3450.0	female	2007	Adelie
4	Torgersen	39.3	20.6	190.0	3650.0	male	2007	Adelie
5	Torgersen	38.9	17.8	181.0	3625.0	female	2007	Adelie
6	Torgersen	39.2	19.6	195.0	4675.0	male	2007	Adelie
...								

Table 1.2: Snippet of the Palmer Penguins dataset.

	Player	Team	GP	GS	MIN	G	ASST	SHOTS	SOG
0	Jamie Vardy	Leicester City	35	34	3034	23	5	71	43
1	Danny Ings	Southampton	38	32	2812	22	2	66	38
2	Pierre-Emerick Aubameyang	Arsenal	36	35	3138	22	3	70	42
3	Raheem Shaquille Sterling	Manchester City	33	30	2660	20	1	68	38
4	Mohamed Salah Ghaly	Liverpool	34	33	2884	19	10	95	59
5	Sadio Mané	Liverpool	35	31	2753	18	7	66	36
6	Harry Kane	Tottenham Hotspur	29	29	2589	18	2	62	37
...									

Table 1.3: Snippet of the Premier League Player Stats dataset.

Chapter 2

Multidimensional Visual Analysis (MVA)

Multidimensional Visual Analysis (MVA) refers to methods and techniques to analyze and understand complex datasets using visual representations. These approaches typically involve the use of specialized software or tools that allow analysts to create and manipulate graphical representations of the data in order to discover patterns, trends, and relationships. This chapter reviews some popular MVA approaches, following the discussions in Cao [2011] and Dzemyda et al. [2012, Chapter 2].

All graphics in the following section were generated using Python with the help of the following libraries: matplotlib [Hunter 2007], pandas [McKinney 2010], scikit-learn [Pedregosa et al. 2011], and umap [McInnes, Healy, Saul et al. 2018] .

2.1 Scatterplots

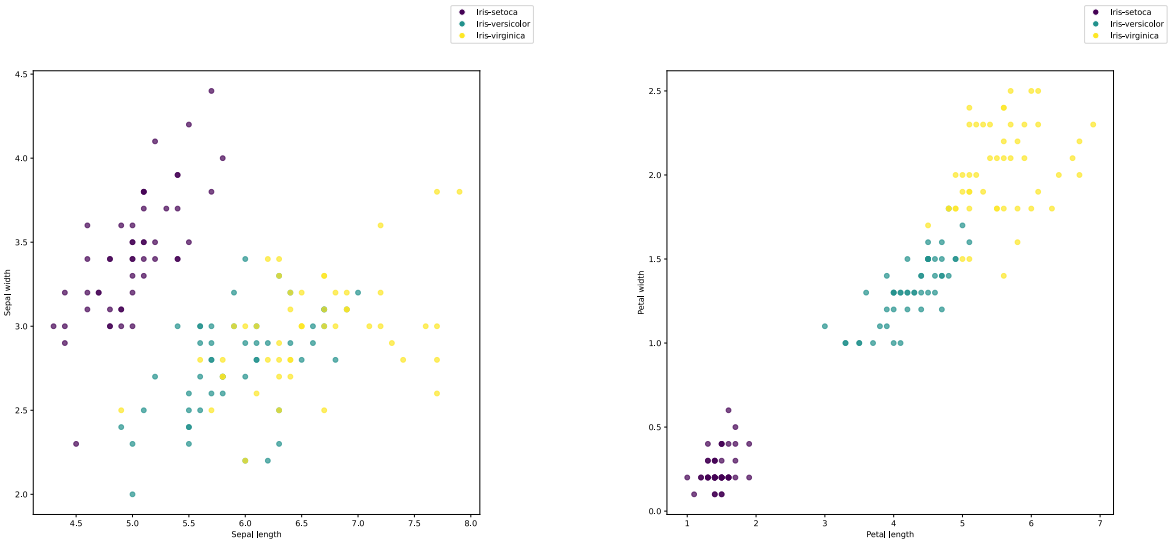
A scatterplot is a type of chart or graph used to display the relationship between two or three numerical dimensions [Friendly and Denis 2005]. It can help identify potential trends, patterns, or outliers in the data. It uses a dot or marker to represent each record, where the position of each dot on the graph indicates the values of the record in each dimension. As such, a scatterplot is a classic (x, y) or (x, y, z) plot.

The resulting graph displays a set of dots, where each dot represents a single record. Considering scatterplots in two dimensions, if the dots tend to form a diagonal line sloping upwards from left to right, that is an indication of a positive relationship (correlation) between them. If the dots tend to form a diagonal line sloping downwards from left to right, that is an indication of a negative relationship (correlation) between them. If the dots appear to be scattered randomly across the graph, there is no apparent linear relationship between the two dimensions. Figure 2.1 shows two scatterplots of the Iris dataset. Figure 2.1a shows the relationship between sepal width and sepal length, where there is no apparent relationship between the two dimensions. Figure 2.1b shows the relationship between petal width and petal length, where there is an apparent correlation between the two dimensions.

One of the key disadvantages of scatterplots is that they can only accommodate two (or three) dimensions at a time. Thus they are insufficient as a tool for MVA on their own.

2.2 Scatterplot Matrices

A scatterplot matrix (SPLOM) is used to explore the pairwise relationships between multiple dimensions [Carr et al. 1986]. It consists of a grid of scatterplots, with each individual plot displaying the relationship between two dimensions. The diagonal is sometimes used display the name and/or a histogram of the distribution of each dimension. The matrix is symmetric in the sense that the bottom left triangle is a reflection of the top right triangle. Figure 2.2 shows an example of a scatterplot matrix displaying the Iris dataset.



(a) Sepal width plotted against sepal length.

(b) Petal width plotted against petal length.

Figure 2.1: Scatterplots of the Iris dataset. [Drawn by Ožbej Golob using Python.]

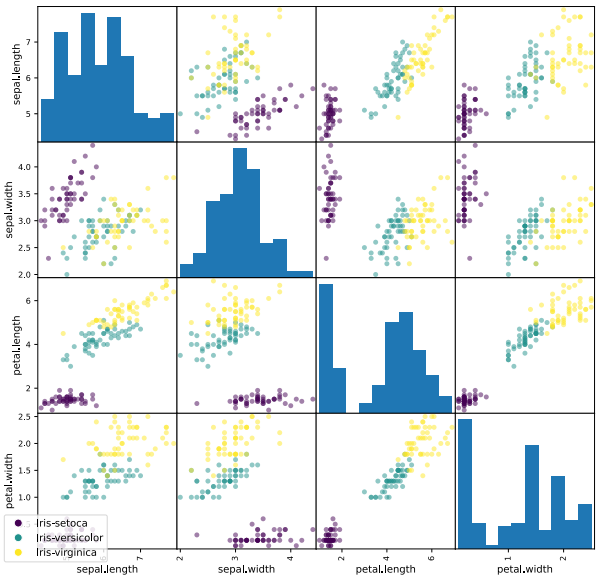
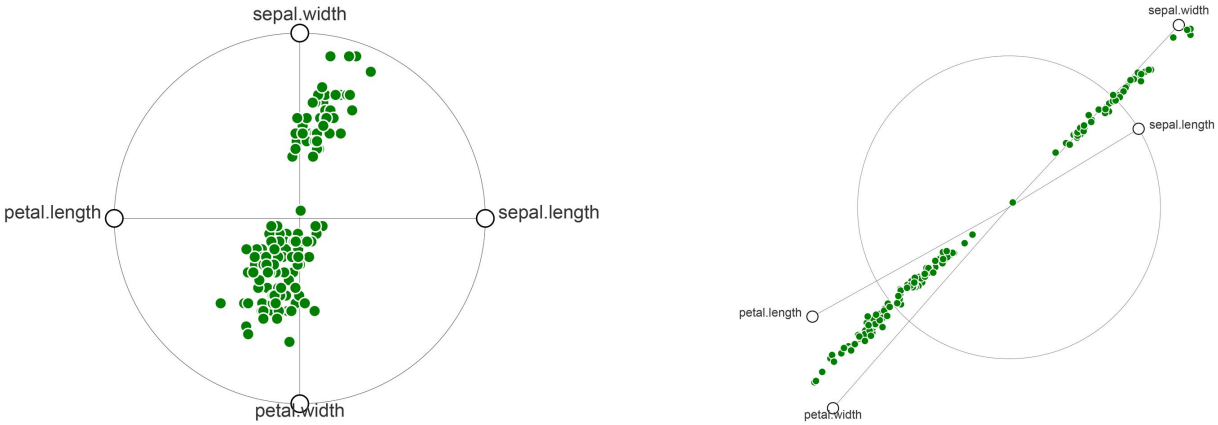


Figure 2.2: A scatterplot matrix displaying all pairwise scatterplots of the four dimensions in the Iris dataset. [Drawn by Ožbej Golob using Python.]



(a) Initial placement of records, with dimensions equally spaced around the perimeter.

(b) Placement of records after repositioning the dimension anchors.

Figure 2.3: Star Coordinates plots of the Iris dataset. [Drawn by Ožbej Golob using RPE [Neuhold et al. 2020].]

One of the primary advantages of using scatterplot matrices is the ability to explore multiple relationships simultaneously. Instead of creating multiple plots to visualize each relationship, a scatterplot matrix provide a comprehensive overview of all the pairwise relationships between dimensions in a single visualization. Typically, an individual scatterplot can be selected and enlarged for closer inspection. However, scatterplot matrices become unwieldly with a larger number (say 15 or more) of dimensions, as they can become cluttered and unreadable.

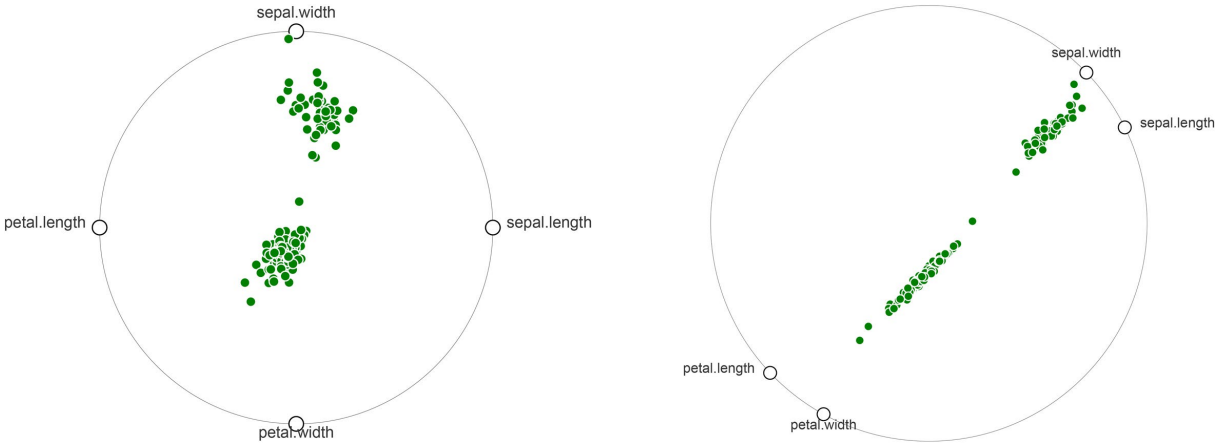
When interpreting scatterplot matrices, it is important to look for patterns and trends that emerge across the plots. For example, if the histograms on the diagonal show that one dimension has a highly skewed distribution, it may be necessary to transform the dimension to improve the performance of a statistical model. Similarly, if multiple plots show a strong linear relationship between two dimensions, it may be worth exploring the use of linear regression models to further investigate the relationship.

2.3 Star Coordinates

Star Coordinates is a visualization technique which maps multidimensional data onto a two-dimensional space using a radial layout [Kandogan 2001]. The basic idea behind Star Coordinates is to represent each record as a set of coordinates on a star-shaped plot. The plot consists of a series of radial axes radiating from a central point, with each axis representing a dimension in the data. Each record is mapped to its position using a weighted average of its values in each of the dimensions. Figure 2.3 shows an example of a star coordinates plot displaying the Iris dataset. In the plot, the data is separated by sepal and petal features (length and width).

The end of each axis is adorned with a circular marker (anchor), which can be interactively moved to relocate the corresponding axis in the plot. The records reposition themselves to follow the anchor's motion. In Star Coordinates, an anchor can be moved outside or inside the circle to emphasize or deemphasize its influence respectively. The records too can reside either inside or outside the circle.

Star Coordinates plot can comfortably accommodate a larger number (say 10 or 20) dimensions, but like other techniques it can become cluttered if the number of records becomes too large.



(a) Initial placement of records, with dimensions equally spaced around the perimeter.

(b) Placement of records after repositioning the dimension anchors.

Figure 2.4: RadViz plots of the Iris dataset. [Drawn by Ožbej Golob using RPE [Neuhold et al. 2020].]

2.4 RadViz

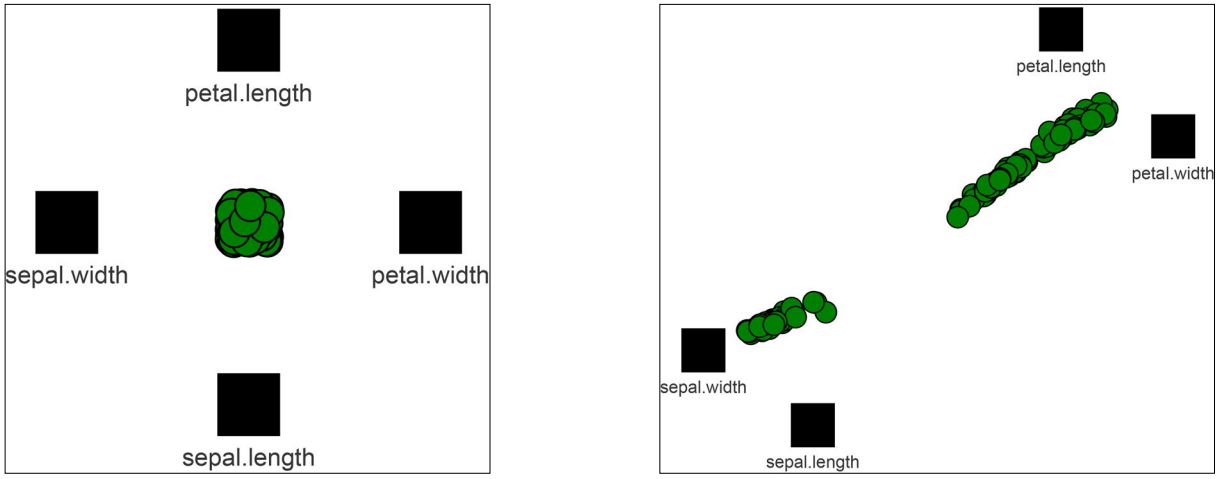
RadViz (Radial Visualization) is similar to Star Coordinates, in that it maps multidimensional data onto a two-dimensional space using a radial layout [Hoffman et al. 1997]. Dimensions are represented by anchor points placed around the perimeter of a circle. Each record is represented by a point inside the circle. The position of the record in the circle is determined by the weighted average of the dimension values associated with that point. The weight of each dimension is determined by the user, and can be used to emphasize or de-emphasize certain dimensions in the visualization. Figure 2.4 shows an example of a RadViz plot displaying the Iris dataset. In the plot, the data is separated by sepal and petal features (length and width).

In the case of RadViz, the anchors can only be moved around the perimeter of the circle, and the records always remain within the circle. A RadViz plot can comfortably accommodate a larger number (say 10 or 20) of dimensions, but like other techniques it can become cluttered if the number of records becomes too large.

2.5 Dust and Magnet (DnM)

Dust and magnet (DnM) is a visualization technique which maps multidimensional data onto a two-dimensional space [Yi et al. 2005]. The basic idea behind DnM is to represent each record as a particle of dust that is attracted to one or more magnets. The magnets represent different dimensions in the data, and their strength determines the influence of each dimension on the records. The stronger the magnet, the greater the influence of the corresponding dimension on the records. The user can increase the influence of a particular dimension by clicking and/or shaking the magnet of the corresponding dimension. The use of particles and magnets provides an intuitive and interactive way to explore and analyze the data. Figure 2.5 shows an example of a DnM plot displaying the Iris dataset. In the plot, the data is separated by sepal and petal features (length and width).

One of the advantages of DnM is that it can comfortably accommodate a larger number (say 10 or 20) of dimensions, but like other techniques it can become cluttered if the number of records becomes too large. Another advantage is its flexibility. Users can adjust the strength of the magnets to emphasize or de-emphasize different dimensions, and can manipulate the position and size of the magnets to highlight specific patterns in the data. However, a potential limitation of DnM is that the technique can be sensitive to the initial configuration of the particles and magnets. This means that users may need to experiment



(a) Initial placement of records in the center.

(b) Placement of records after using moving and shaking the magnets.

Figure 2.5: DnM plots of the Iris dataset. [Drawn by Ožbej Golob using RPE [Neuhold et al. 2020].]

with different settings to achieve an informative visualization.

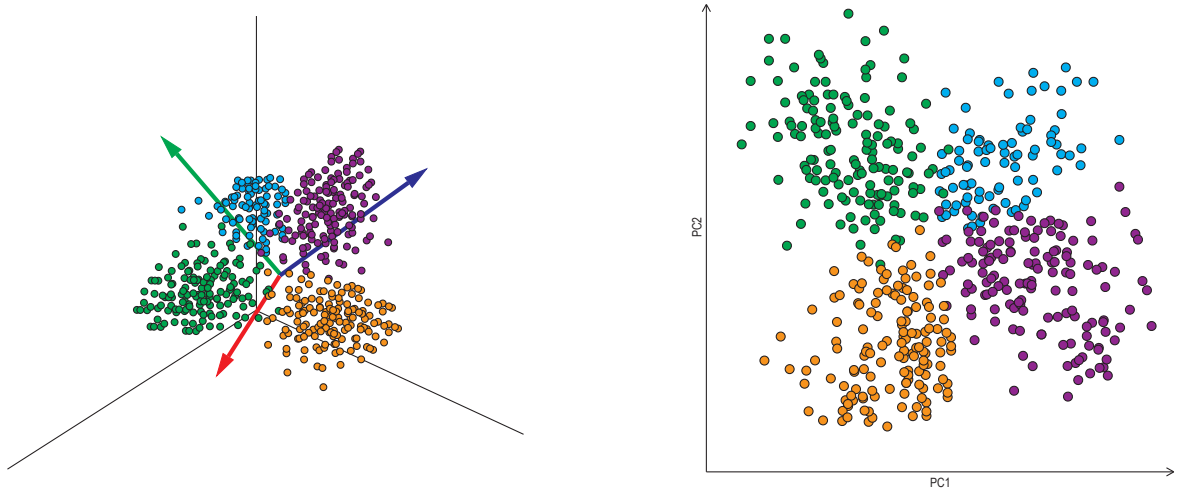
2.6 Similarity Maps

Similarity maps are projections of high-dimensional datasets to two (or sometimes three) dimensions. High-dimensional datasets, by definition, have a large number of dimensions (often hundreds or thousands), which presents many computational and mathematical challenges. Projection techniques are used to reduce the number of dimensions in the data, while attempting to preserve distances between items as much as possible. Items which are close in the high-dimensional space should also be close in the resulting two-dimensional projection space. One of the key advantages of similarity maps is that they can accommodate any number of dimensions, which are then mapped into two (or three) dimensions. Projections can further be split into *linear* and *non-linear* projections.

A linear projection is a transformation that can be represented by a linear function. This means that the output of a linear projection is a linear combination of the input dimensions, where each dimension is multiplied by a scaling factor and then added together. In other words, a linear projection involves scaling and rotating the data without distorting it, and its output could be used to label the projected axes. Linear projections are useful for reducing the dimensionality of data while preserving its structure and relationships between dimensions. Common examples of linear projections include Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Figure 2.6 shows an example similarity map where the number of dimensions were reduced from three to two using Principal Component Analysis (PCA).

A non-linear projection, on the other hand, involves more complex transformations that cannot be represented by a linear function. This means that the output of a non-linear projection is a complex combination of the input dimensions that may involve multiplication, exponentiation, or other non-linear operations. Non-linear projections can distort the data in order to reveal patterns and relationships that may not be apparent in the original high-dimensional space. Non-linear projections are particularly useful for data that exhibits complex and non-linear relationships between dimensions. Examples of non-linear projections include t-distributed Stochastic Neighbor Embedding (t-SNE) and Locally Linear Embedding (LLE).

Overall, the main difference between linear and non-linear projections is that linear projections preserve



(a) A 3d scatterplot of the original 3d dataset. The red, blue, and green arrows are the directions of the first, second, and third principal components, respectively.

(b) A 2d scatterplot of the first two principal components from PCA.

Figure 2.6: Scatterplots before and after dimensionality reduction with PCA. [Drawn by Ožbej Golob using Adobe Illustrator.]

the structure and relationships between dimensions, while non-linear projections may distort the data to reveal complex patterns and relationships.

2.6.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a linear projection [Abdi and Williams 2010]. PCA uses linear algebra to identify the underlying dimensions or factors in the data and project the data onto a lower-dimensional space. This is done by analyzing the covariance matrix of the original dataset, which describes the relationship between the different dimensions. By calculating the eigenvectors and eigenvalues of the covariance matrix, PCA identifies the directions in which the data varies the most, the *principal components*.

The principal components are ordered by the amount of variance they explain in the original data. The first principal component explains the largest amount of variance, followed by the second principal component, and so on. By keeping only the principal components that explain the majority of the variance in the data, the dimensionality of the dataset can be reduced, while still retaining the most important information .

2.6.2 Multi-Dimensional Scaling (MDS)

Multi-Dimensional Scaling (MDS) is a non-linear projection [Morrison et al. 2003]. MDS is used to visualize the similarity or dissimilarity between different objects or observations. The technique transforms a set of high-dimensional data into a lower-dimensional space, usually two or three dimensions, while preserving the relationships between the records.

MDS works by first calculating the pairwise distances or dissimilarities between all the objects in the dataset. These distances could be based on any metric, such as Euclidean distance, correlation distance, or other similarity measures. Once the distance matrix is constructed, MDS tries to find a configuration of points in a lower-dimensional space that best reproduces the distances or dissimilarities between the

original objects. This is done by minimizing a cost function, such as stress or error, which measures the discrepancy between the original pairwise distances and the distances between the projected points.

MDS can be classified into two main types: metric and non-metric. In metric MDS, the pairwise distances between the objects are preserved exactly, while in non-metric MDS, only the rank order of the distances is preserved. Non-metric MDS is often used when the underlying distance metric is unknown or not easily quantifiable.

2.6.3 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear projection [Van der Maaten and Hinton 2008; Wattenberg et al. 2016]. t-SNE uses a probabilistic model to preserve the local structure of the data in the high-dimensional space and project the data onto a lower-dimensional space, usually two or three dimensions.

The t-SNE algorithm works by first calculating the pairwise similarities between all the records in the high-dimensional space. This is typically done using a Gaussian kernel, which measures the similarity between two points as a function of their Euclidean distance. The similarities are then used to construct a probability distribution for each point that defines the likelihood of finding other points nearby.

In the low-dimensional space, t-SNE tries to find a configuration of points that preserves the pairwise similarities between the high-dimensional records as closely as possible. The algorithm does this by minimizing a cost function that measures the divergence between the probability distributions in the high-dimensional and low-dimensional spaces. The optimization is performed using gradient descent, which iteratively updates the position of each point in the low-dimensional space until the cost function is minimized.

One of the key features of t-SNE is that it uses a Student's t-distribution to model the probability distribution in the low-dimensional space. This allows t-SNE to emphasize the differences between nearby records and de-emphasize the differences between distant records, which helps to prevent crowding and distortion in the final visualization.

2.6.4 Uniform Manifold Approximation and Projection (UMAP)

Uniform Manifold Approximation and Projection (UMAP) is a non-linear dimension reduction algorithm based on manifold learning techniques and ideas from topological data analysis [McInnes, Healy and Melville 2018]. UMAP is based on three assumptions about the data: (i) The data has a uniform distribution of a Riemannian manifold, (ii) The Riemannian metric is locally constant, and (iii) The manifold is locally connected. UMAP uses these assumptions to model the manifold with a fuzzy topological structure. The embedding is determined by searching for a low-dimensional projection of the data that is the closest equivalent to the fuzzy topological structure.

UMAP works by creating a low-dimensional representation of the data while preserving the global structure of the data. This is achieved by modeling the data as a high-dimensional manifold, which is a mathematical object that represents the underlying structure of the data. UMAP then finds a low-dimensional embedding of the manifold that preserves the local structure of the data.

The UMAP algorithm consists of several steps. First, a nearest-neighbor graph is constructed by connecting each record to its nearest neighbors. This graph is then used to create a fuzzy simplicial set, which is a mathematical object that captures the local structure of the data. The fuzzy simplicial set is then transformed into a low-dimensional representation using a stochastic gradient descent algorithm.

UMAP has several advantages over other dimensionality reduction techniques. For example, UMAP is able to preserve both the global and local structure of the data, which means that it can be used for both exploratory data analysis and machine learning tasks.

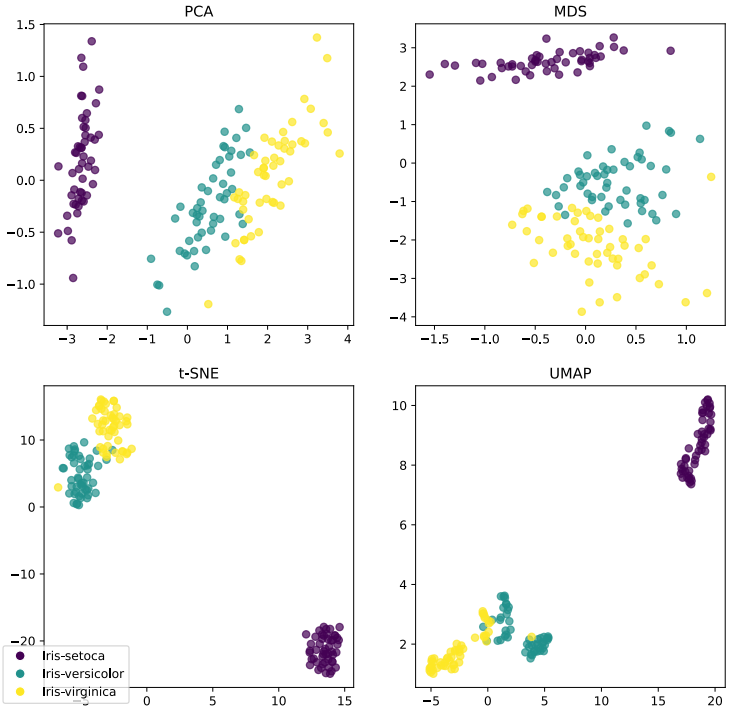


Figure 2.7: Comparison of four similarity mapping techniques on the Iris dataset. [Drawn by Ožbej Golob using Python.]

2.6.5 Comparison of Similarity Mapping Techniques

To illustrate the previously mentioned four similarity mapping techniques, they were applied to two well-known datasets. Figure 2.7 shows the four similarity mapping techniques applied to the Iris dataset. All four techniques separate the Iris Setoca class from the other two, while the other two cannot be clearly separated from each other. t-SNE and UMAP generate dense clusters, while PCA and MDS clusters are more sparse.

Figure 2.8 shows the four similarity mapping techniques applied to the Palmer Penguins dataset. All four techniques separate the Gentoo class from the other two, while the other two cannot be clearly separated from each other. Again, t-SNE and UMAP generate dense clusters, while PCA and MDS clusters are more sparse.

2.7 Parallel Coordinates

Parallel Coordinates are a type of chart used to visualize multi-dimensional data [Inselberg and Dimsdale 1990; Inselberg 2009]. In this type of chart, each dimesnion is represented by a vertical axis arranged in parallel. Each record is represented by a horizontal polyline. The position where a polyline touches an axis indicates the value of the record for that dimension. Parallel Coordinates allow multiple dimensions to be compared and analyzed simultaneously. Figure 2.9 shows an example of a parallel coordinates plot displaying records with multiple dimensions.

One of the key advantages of parallel coordinates plot is that it can comfortably accommodate a larger number (say 10 or 20) of dimensions, but like other techniques it can become cluttered if the number of records becomes too large.

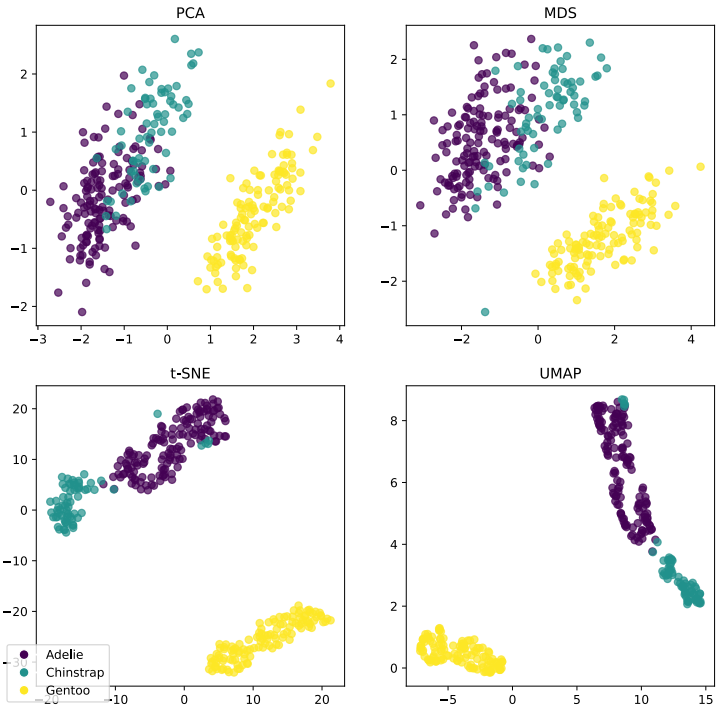


Figure 2.8: Comparison of four similarity mapping techniques on the Palmer Penguins dataset. [Drawn by Ožbej Golob using Python.]

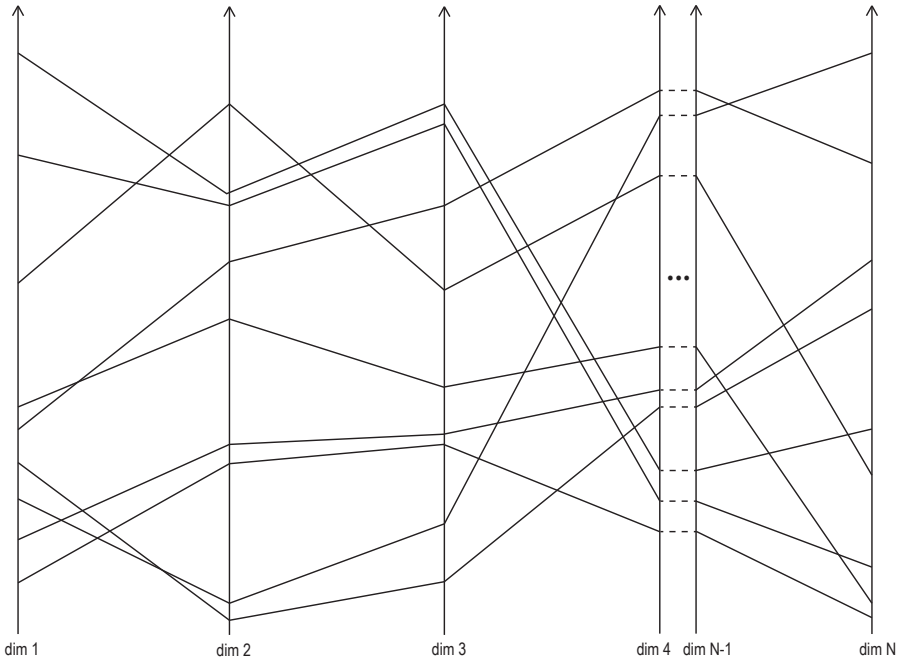


Figure 2.9: A parallel coordinates plot showing dimensions as vertical axes and records as horizontal polylines. [Drawn by Ožbej Golob using Adobe Illustrator.]

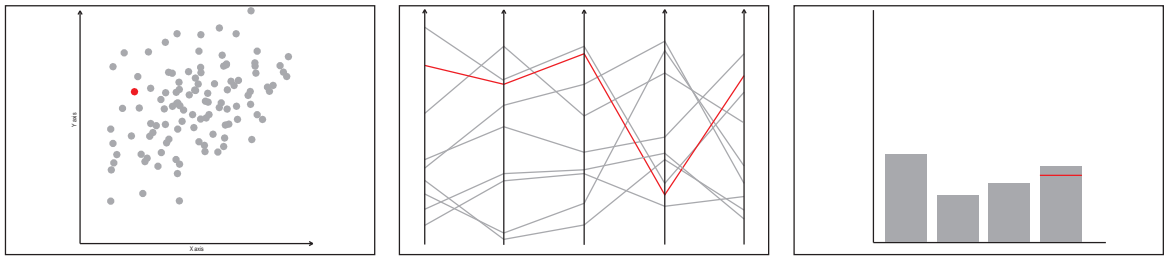


Figure 2.10: An example of brushing. The user selected a single record which is now highlighted in red in all views. [Drawn by Ožbej Golob using Adobe Illustrator.]

2.8 Brushing and Linking

Brushing and linking are techniques used in multidimensional visual analysis to allow the user to interact with a visualization and explore data in greater depth. *Brushing* refers to the process of selecting records or regions in one visualization and highlighting those records in other visualizations. This allows the user to see how the records or regions of interest are related to other dimensions in the dataset. Figure 2.10 shows an example of brushing. The user selected a single record which is now highlighted in red in all views.

Linking refers to the process of synchronizing the views of multiple visualizations, such that a change made to one visualization is reflected in the other visualizations. This allows the user to explore the data from different perspectives and understand how different dimensions are related to one another. Both brushing and linking are useful for helping users to identify patterns and relationships in the data and for facilitating the process of data exploration and analysis.

2.9 Grouping and Labeling

Grouping refers to the process of identifying and separating similar records, while *labeling* refers to the process of naming the groups. Grouping and labeling help organize and structure data records, allowing for more accurate and meaningful analysis of the data, as similar records can be grouped together and analyzed in relation to one another. It also allows a Machine Learning (ML) model to understand the context of the data and make more accurate predictions witzh greater interpretability.

Manual grouping and labeling refers to the process of organizing and providing relevant information about data manually, typically done by a human. The process of manual grouping and labeling can be time-consuming and labor-intensive.

Automated clustering uses mathematical methods used to identify which objects in a given data set are similar. Similar records are automatically grouped together into clusters. This allows analysts to identify common patterns and trends within the data, and to gain a better understanding of the relationships between the objects in the dataset [Romesburg 1984].

Chapter 3

MVA Tools

Multidimensional Visual Analysis (MVA) software is designed to help analysts explore and analyze complex datasets. These programs typically include a wide range of tools and features that allow users to create and manipulate graphical representations of the data, such as scatterplots, parallel coordinates, and similarity maps. Using these tools, analysts can quickly and easily explore the data and gain insights that might not be immediately apparent from looking at the raw data. MVA software is commonly used in fields such as business, finance, and marketing to help make data-driven decisions and uncover hidden trends in the data. In this section, some popular MVA software tools are reviewed.

3.1 XMDV

XMDV tool [Ward 1994] is a software package for the interactive visual exploration of multidimensional datasets. XMDV is written in Qt and Eclipse CDT and is available as free and open-source software. It was initially released in 1994 and last updated on 24 Sept 2021. XMDV is available for Windows, macOS, and Linux.

XMDV can load custom datasets in its own custom `.okc` format. XMDV displays the data in the following views: scatterplot matrix, parallel coordinates, star glyphs, dimensional stacking, tree maps, and pixel-oriented display. XDMV also supports many interaction modes and tools, including brushing and linking. Figure 3.1 shows a screenshot of the XMDV tool.

3.2 Parallax

Parallax [T. Avidan and S. Avidan 1999; Inselberg 2009, Chapter 10] is a tool for effectively analyzing multidimensional datasets and discovering patterns, properties, and relations in data. Parallax was developed by a small Israeli software company, MDG, in collaboration with Alfred Inselberg. It was initially released in 1994 and was available as commercial, closed-source software.

Parallax can load custom datasets in a custom format, based on a simple text file with `.dat` extension. The main part of Parallax is a powerful parallel coordinates view, which enables queries. The results of queries can be grouped and then shown seperately or in combination with other queries. Parallax provides the following views: scatterplot, parallel coordinates, and distribution (histogram). Parallax does not suport brushing and linking. Figure 3.2 shows a screenshot of the Parallax tool.

3.3 GGobi

GGobi [Cook et al. 2007] is a visualization program for exploring high-dimensional data. GGobi is written in C and is available as a free and open-source software. It was initially released in 1999 and last updated on 10 Jun 2012. GGobi is available for Windows, macOS, and Linux.

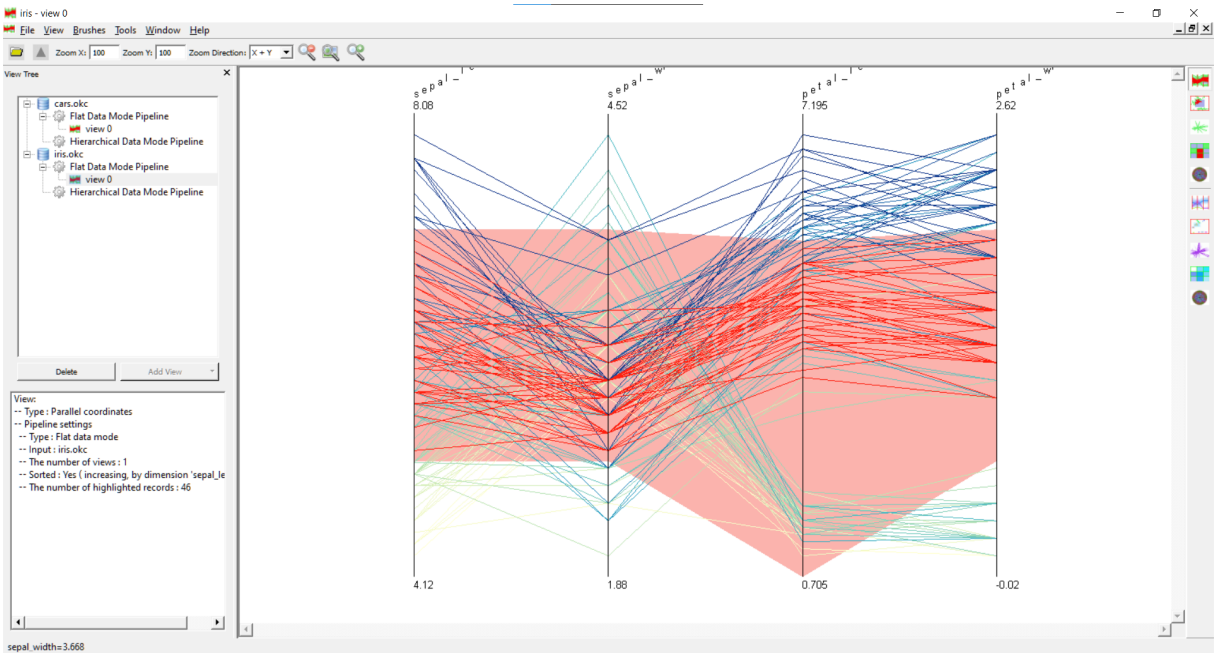


Figure 3.1: XMDV displaying the Iris dataset.

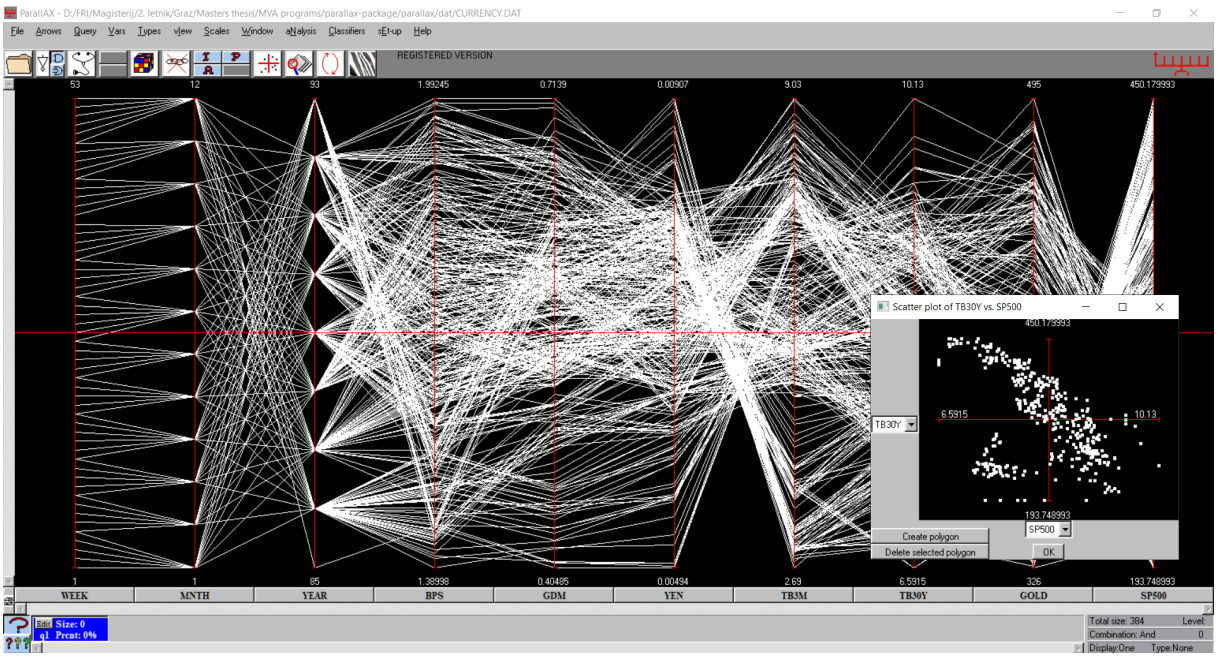


Figure 3.2: Parallax displaying a financial dataset.

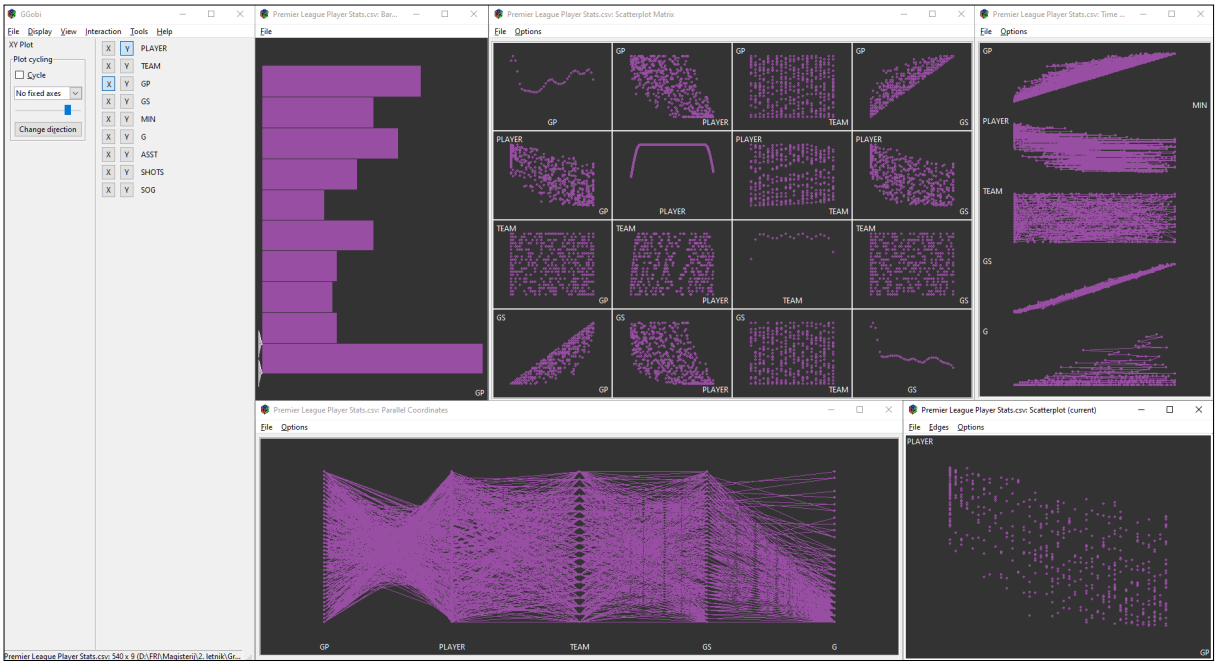


Figure 3.3: GGobi displaying the Premier League Player Stats dataset [Samariya 2020].

GGobi can load custom datasets and provides dynamic and interactive graphics as tours, where data is displayed in an animation. The data is also available in the following views: scatterplot, scatterplot matrix, parallel coordinates, time series, and distributions (histograms). GGobi supports limited brushing. The views offer limited interactivity and interpretability and are not closely connected. Figure 3.3 shows a screenshot of the GGobi tool.

3.4 InfoScope

InfoScope [Macrofocus 2015; Girardin and Brodbeck 2001; Brodbeck and Girardin 2003] is an interactive visualization tool to access, explore, and communicate large or complex datasets. InfoScope is available as free software. It was initially released in 2001 and last updated on 19 Aug 2015. InfoScope was available for Windows, macOS, and Linux.

InfoScope can load a selection of preprepared datasets, mainly from the finance sector. It is also possible to load a custom dataset using its own custom `.mis` format. InfoScope provides the following views: carto plot, similarity map, parallel coordinates, and table view. InfoScope supports brushing and linking, so all views are highly interactive and tightly linked. Users can obtain exact record values by inspecting the records, and select a subset of records with range sliders. InfoScope supports the manual grouping of records by color. Figure 3.4 shows a screenshot of InfoScope. It later evolved into the tool High-D.

3.5 XDAT

XDAT [XDAT 2020] is a multidimensional data analysis tool designed to help users quickly and easily extract valuable insights from large, complex datasets with many dimensions. XDAT is written in Java and is available as free software. It was initially released in May 2010 and last updated on 26 Aug 2020. XDAT is available for Windows, macOS, and Linux.

XDAT can load custom datasets and displays data in separate views. The following views are available: parallel coordinates, table view, and scatterplot. All of the visualizations are interconnected through

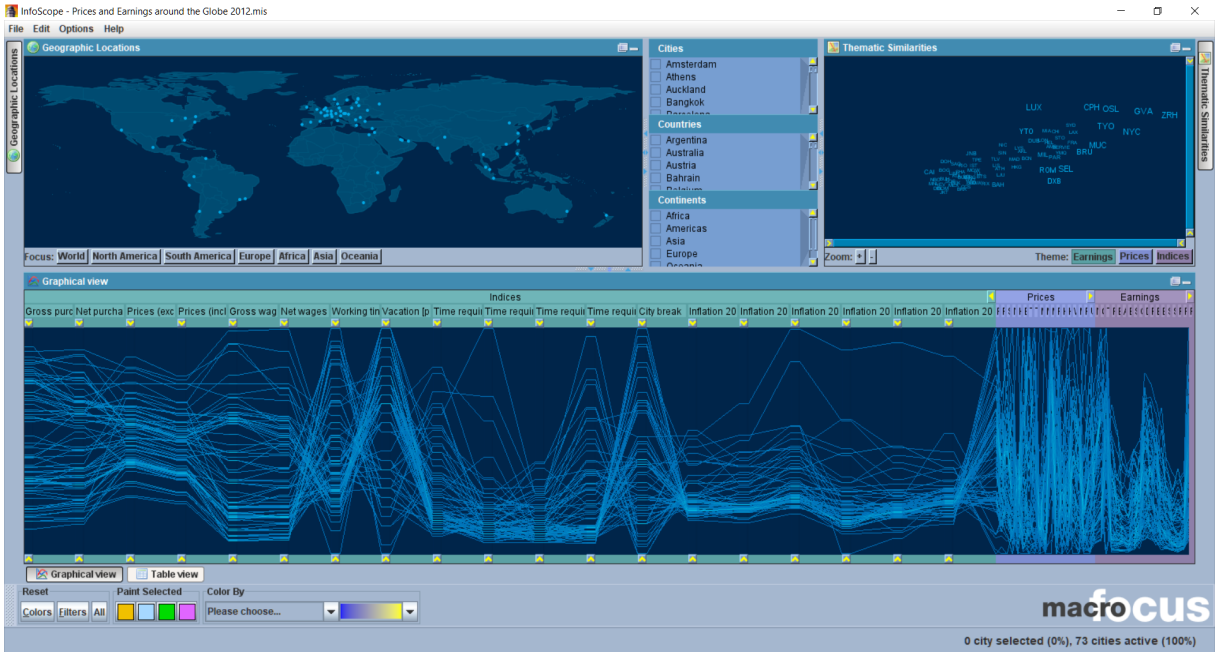


Figure 3.4: InfoScope displaying the *Prices and Earnings around the Globe 2012* dataset.

standard brushing and linking, so that any changes or selections made in one view are reflected in all of the other views. Figure 3.5 shows a screenshot of the XDAT tool.

3.6 High-D

High-D [Macrofocus 2022] is the successor to InfoScope. It offers similar functionality with some improvements and added views. High-D is a versatile tool for revealing hidden features, highlighting trends and relationships, and finding anomalies in datasets of any size. At its heart is a powerful interactive parallel coordinates plot. High-D is available as commercial software for US$ 199 and with 30-day free evaluation period. It was initially released in Sept 2013 and last updated on 05 Dec 2022. High-D is available for Windows, macOS, and Linux.

High-D can load a collection of selected publicly available datasets as well as custom datasets. High-D provides the following views: parallel coordinates, parallel coordinates matrix, table plot, distributions, scatterplot matrix, scatterplot, similarity map (Sammon, Spring, t-SNE, and PCA), tree map, and carto plot. High-D supports manual grouping of data by color and automated clustering with a k-means algorithm. High-D also supports brushing and linking, so all views are highly interactive and tightly linked. Users can obtain exact record values by inspecting the records, and select a subset of records with range sliders. Figure 3.6 shows a screenshot of High-D.

3.7 TabuVis

TabuVis [Nguyen et al. 2013] is a flexible and customizable visual analytics system for multidimensional data. Its visualizations can be customized by domain experts to suit the specific needs of the data being analyzed. TabuVis is written in Java and is available as free software. It was initially released in May 2013 and last updated on 19 Feb 2022. TabuVis is available for Windows, macOS, and Linux.

TabuVis can load custom datasets and displays data in separate views. TabuVis includes various features for analyzing data, such as the ability to process data, add automatic marks, create custom interactive visualizations, and filter the data. These features are designed to support the entire data analysis process. TabuVis displays the data in the following views: scatterplot, parallel coordinates, and

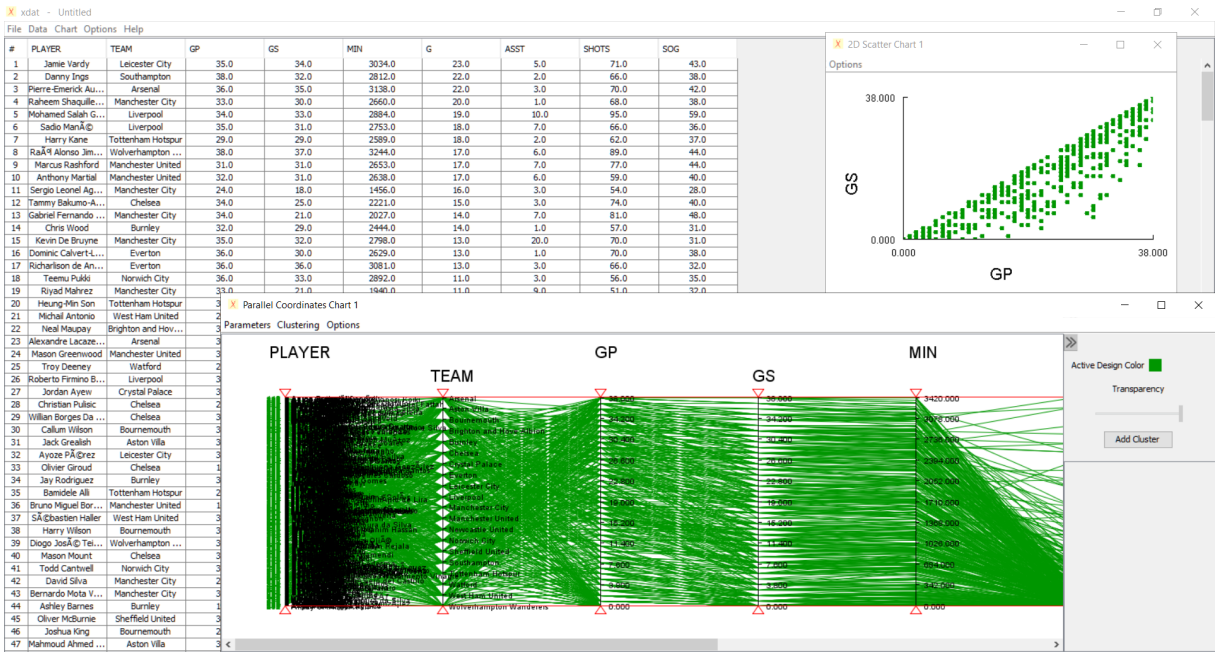


Figure 3.5: XDAT displaying the Premier League Player Stats Dataset [Samariya 2020].

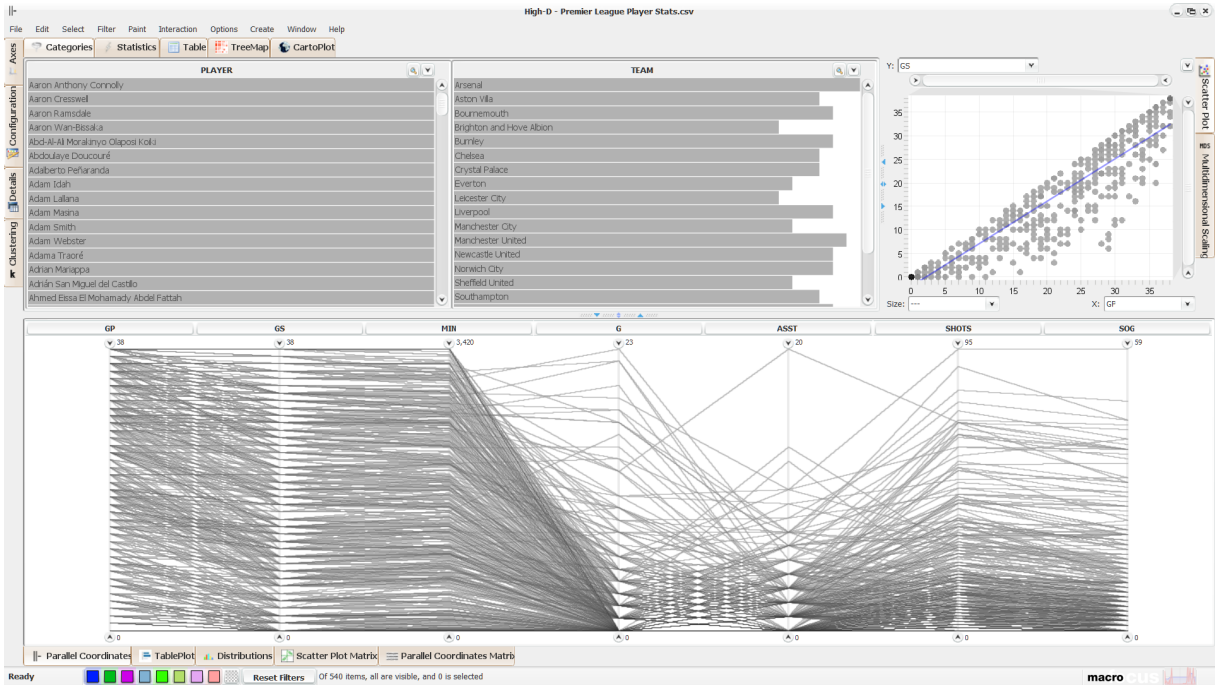


Figure 3.6: High-D displaying the Premier League Player Stats dataset [Samariya 2020].

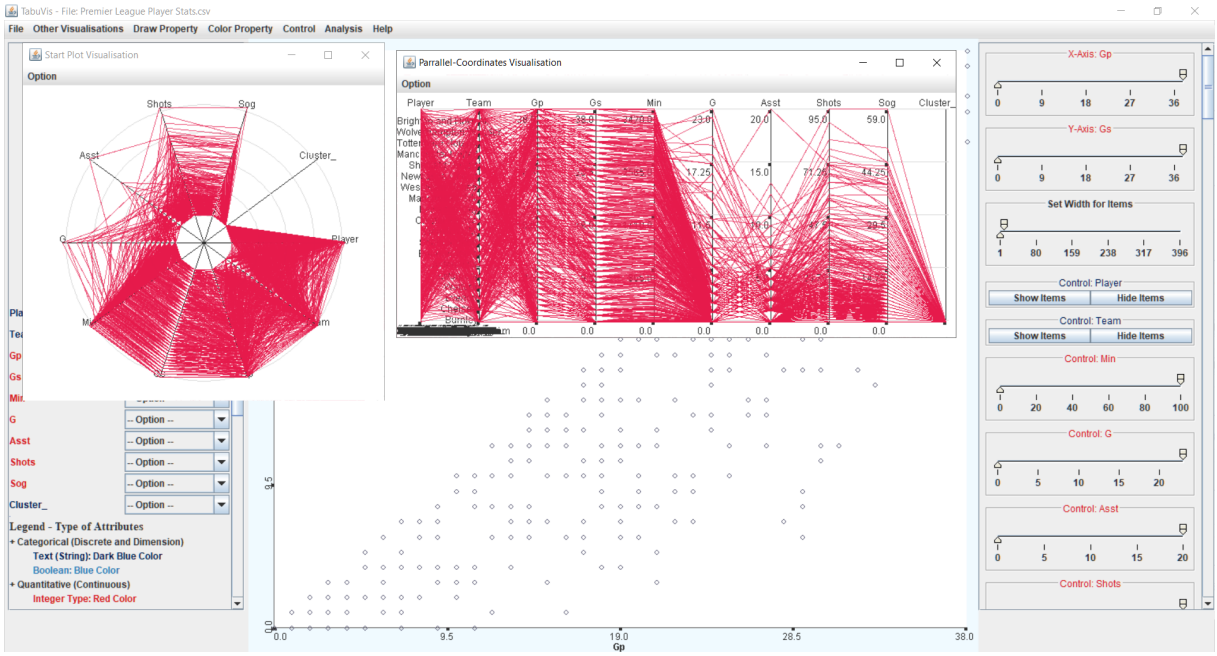


Figure 3.7: TabuVis displaying the Premier League Player Stats dataset [Samariya 2020].

star plot. TabuVis does not support brushing and linking. Figure 3.7 shows a screenshot of the TabuVis tool.

3.8 Improvise

Improvise [Weaver 2014] is a program that allows users to create and interact with visualizations that are linked together in various ways. Improvise is written in Java and is available as free, open-source software. It was initially released in 2014 and last updated on 28 Oct 2020. Improvise is available for Windows, macOS, and Linux.

The program uses a shared-object coordination model and a declarative visual query language to give users control over how data is displayed in multiple views. This allows users to create visualizations with a variety of coordination patterns, such as synchronized scrolling, overview and detail, drill-down, and semantic zoom. Improvise also has a user interface that allows users to build and explore visualizations in a live environment, making it easy to modify visualizations as needed. The goal of Improvise is to provide a high level of coordination flexibility while also being easy to use. Figure 3.8 shows a screenshot of the Improvise tool.

3.9 MyBrush

MyBrush [Koytek et al. 2017] is an experimental application that allows users to customize and control the brushing and linking process in their visualizations. It provides flexibility by allowing users to specify the source, link, and target of multiple brushes, and supports a variety of visualization types and multiple simultaneous brushes. Improvise is written in JavaScript and is available as a free, open-source web application. It was initially released in Jun 2016 with the last update issued on 22 Sept 2017.

MyBrush is experimental and offers limited functionality. Its purpose is to explore brushing and linking functionality. A user can explore a predetermined set of data with the following views: scatterplot, parallel coordinates, and bar plot. Any changes or selections made in one of the visualizations are reflected in all of the other views because they are all interconnected through standard brushing and linking. Figure 3.9 shows a screenshot of the MyBrush tool.

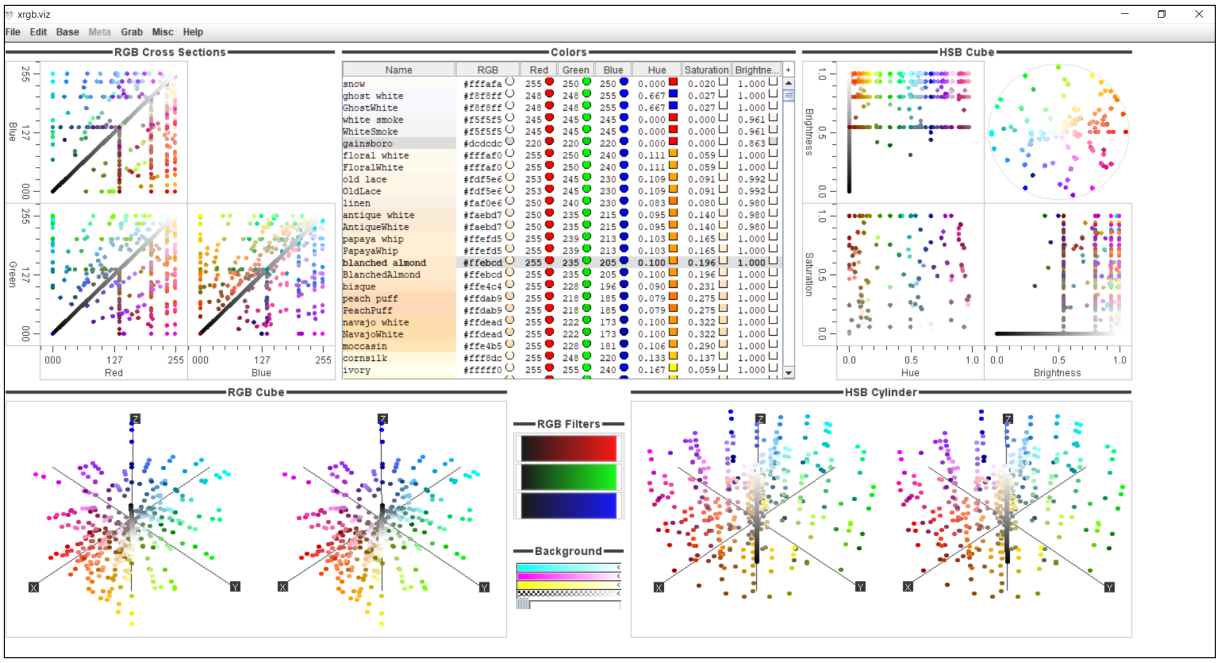


Figure 3.8: Improvise displaying the *Improvise Custom XRGB* dataset.

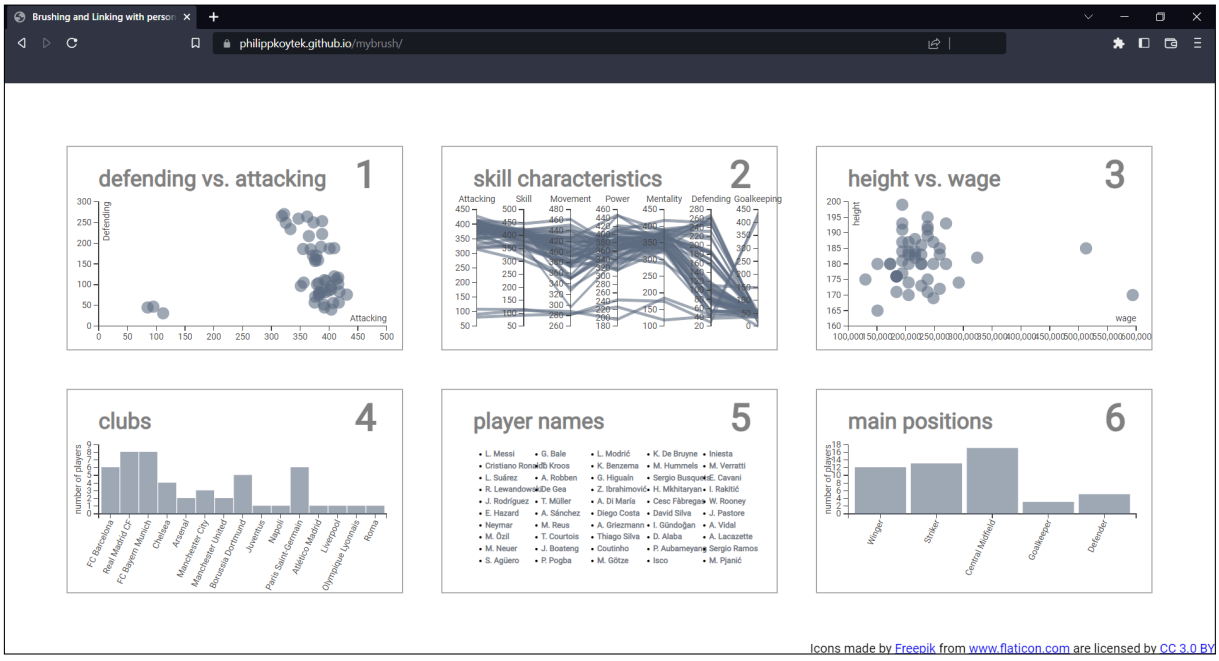


Figure 3.9: MyBrush displaying the *MyBrush Custom Football Players* dataset.

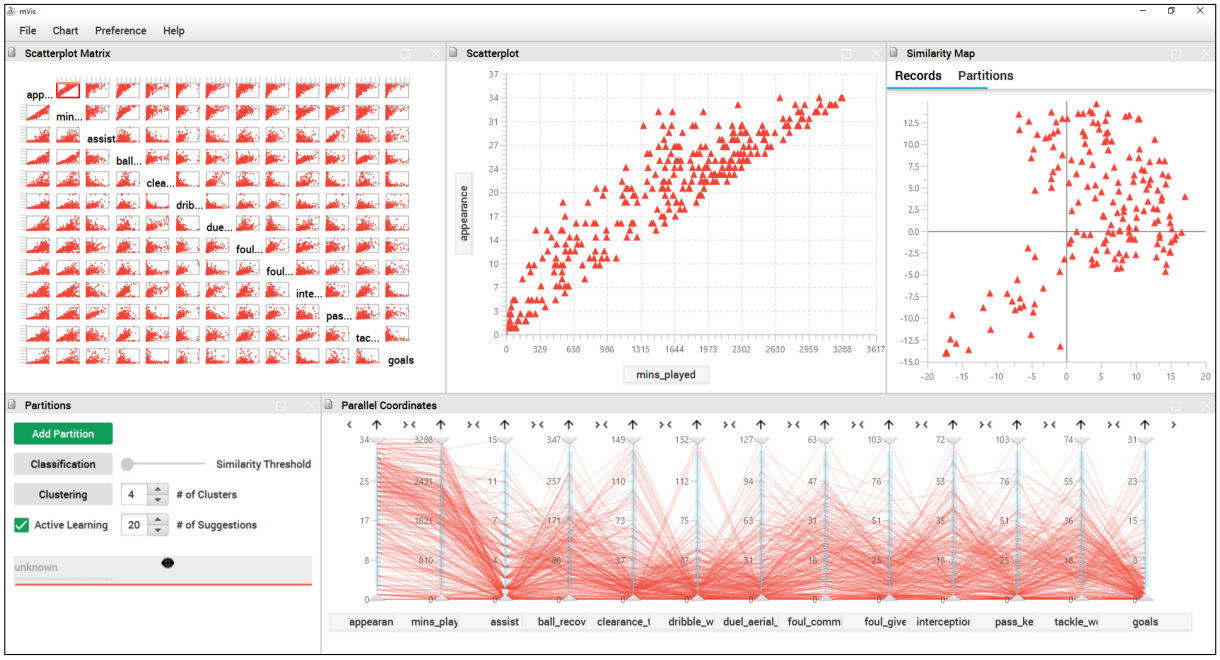


Figure 3.10: mVis displaying the *mVis Custom Football Players* dataset.

3.10 mVis

mVis [Chegini et al. 2019] is a visual analytics tool for visualizing multi-dimensional data. mVis is written in Java and is available as free, open-source software. It was initially released in Jul 2020 and last updated on 20 Jan 2021. mVis is available for Windows, macOS, and Linux.

mVis can visualize custom datasets and provides an overview of global relationships between objects by using multiple views to show different aspects of the data at the same time. mVis consists of four data visualization views: scatterplot matrix, scatterplot, similarity map (t-SNE, PCA, and MDS), and parallel coordinates. mVis also has a panel for controlling data partitions. All of the visualizations are interconnected through standard brushing and linking, so that any changes or selections made in one view are reflected in all of the other views. Additionally, the user has the ability to close, rearrange, or expand any view as needed.

mVis supports the interactive visual labeling of records in a dataset, both manually with the linked visualizations and by running built-in clustering and active learning methods. An analyst can take a dataset and efficiently partition it into classes by labelling it records. Such a labelled dataset can then be used as a training dataset for machine learning. Figure 3.10 shows a screenshot of the mVis tool.

Chapter 4

Concluding Remarks

This survey reviewed several popular MVA approaches and ten of the most popular MVA software tools. An overview of the ten tools can be seen in Table 4.1. The features provided by the ten tools are compared Table 4.2.

Nine of the MVA tools provide support for importing custom datasets. The tenth (MyBrush) is only intended as an experimental proof of concept. Most MVA tools support brushing and linking, which allow users to interact with a visualization, explore data in greater depth, and see how the records or regions of interest are related to other dimensions in the dataset. Some tools also provide support for manual grouping (which is useful for preparing and labeling data for ML models), and automated clustering. The most common views provided by the the reviewed tools are: scatterplot, scatterplot matrix, parallel coordinates, and similarity map. Other views are not as important and are not as widely used.

Of the reviewed tools, High-D offers the most functionality. However, it is available only as paid software. The best freely available software is mVis, which offers most the important views and features needed for an MVA tool.

	XMDV	**Parallax**	**GGobi**	**InfoScope**	**XDAT**	**High-D**	**TabuVis**	**Improvise**	**MyBrush**	**mVis**
Initial Release:	1994	Mar 1999	1999	2001	May 2010	Sep 2013	May 2013	2014	Jun 2016	Jul 2020
Last Update:	Sept 2021	?	Jun 2012	Aug 2015	Aug 2020	Dec 2022	Feb 2022	Oct 2020	Sept 2017	Jan 2021
License:	Free, open-source	Commer-cial	Free, open-source	Free demo	Free	Commer-cial	Free	Free, open-source	Free, open-source	Free, open-source
Systems:	Win, MacOS, Linux	Win, MacOS, Linux	Win, MacOS, Linux	Win, MacOS, Linux	Win, MacOS, Linux	Win, MacOS, Linux	Win, MacOS, Linux	Win, MacOS, Linux	Web Browser	Win, MacOS, Linux
Language:	Qt	?	C	Java	Java	Java	Java	Java	JavaScript	Java
Installation:	Local	Local	Local	Local	Local	Local	Local	Local	Online	Local

Table 4.1: Overview of MVA tools.

Feature	XMDV	Parallax	GGobi	InfoScope	XDAT	High-D	TabuVis	Improvise	MyBrush	mVis
Custom Datasets:	✓	✓	✓	✓	✓	✓	✓	✓		✓
Brushing:	✓		✓	✓	✓	✓			✓	✓
Linking:	✓		✓	✓	✓	✓			✓	✓
Manual Grouping:	✓	✓		✓	✓	✓	✓			✓
Automated Clustering:		✓				✓	✓			✓
Table View:				✓	✓	✓		✓		
Scatterplot:	✓	✓	✓		✓	✓	✓	✓	✓	✓
Scatterplot Matrix:	✓		✓			✓		✓		✓
Parallel Coordinates:	✓	✓	✓	✓	✓	✓	✓		✓	✓
ParCoord Matrix:						✓				
Similarity Map:				✓		✓	✓	✓		✓
Time Series:			✓					✓		
Distributions:		✓	✓			✓		✓	✓	
Table Plot:						✓		✓		
Tree Map:	✓					✓		✓		
Carto Plot:				✓		✓		✓		

Table 4.2: Comparison of MVA tool features.

Bibliography

Abdi, Hervé and Lynne J. Williams [2010]. *Principal Component Analysis*. Wiley Interdisciplinary Reviews: Computational Statistics 2.4 (15 Jul 2010), pages 433–459. ISSN 1939-0068. doi:10.1002/wics.101. https://personal.utdallas.edu/~herve/abdi-awPCA2010.pdf (cited on page 8).

Avidan, Tova and Shlomo Avidan [1999]. *ParallAX – A Data Mining Tool Based on Parallel Coordinates*. Computational Statistics 14.1 (Mar 1999), pages 79–89. ISSN 0943-4062. doi:10.1007/PL00022707 (cited on page 13).

Brodbeck, Dominique and Luc Girardin [2003]. *Using Multiple Coordinated Views to Analyze Geo-Referenced High-Dimensional Datasets*. Proc. International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2003). 15 Jul 2003, pages 104–111. doi:10.1109/CMV.2003.1215008. http://download.macrofocus.com/publications/cmv2003.pdf (cited on page 15).

Cao, Nan [2011]. *A Survey on Multidimensional Visual Analysis Techniques*. Slide deck. Sep 2011. https://nancao.org/pubs/cao_pqe_chart.pdf (cited on page 3).

Carr, Daniel B., Richard J. Littlefield, W. L. Nicholson, and J. S. Littlefield [1986]. *Scatterplot Matrix Techniques for Large N*. Journal of the American Statistical Association 82.398 (01 Jul 1986), pages 424–436. doi:10.1080/01621459.1987.10478445 (cited on page 3).

Chegini, Mohammad, Jürgen Bernard, Philip Berger, Alexei Sourin, Keith Andrews, and Tobias Schreck [2019]. *Interactive Labelling of a Multivariate Dataset for Supervised Machine Learning using Linked Visualisations, Clustering, and Active Learning*. Visual Informatics 3.1 (Mar 2019). Proc. PacificVAST 2019, pages 9–17. doi:10.1016/j.visinf.2019.03.002. https://ftp.isds.tugraz.at/pub/papers/chegini-pvast2019-ix-labelling.pdf (cited on page 20).

Cook, Dianne, Deborah F. Swayne, and Andreas Buja [2007]. *Interactive and Dynamic Graphics for Data Analysis with R and GGobi*. Springer, 05 Sep 2007. ISBN 0387717617 (cited on page 13).

Dzemyda, Gintautas, Olga Kurasova, and Julius Žilinskas [2012]. *Multidimensional Data Visualization: Methods and Applications*. Springer, 09 Nov 2012. ISBN 144190235X (cited on page 3).

Fisher, Ronald A [1936]. *The Use of Multiple Measurements in Taxonomic Problems*. Annals of Eugenics 7.2 (1936), pages 179–188. doi:10.1111/j.1469-1809.1936.tb02137.x (cited on page 1).

Friendly, Michael and Daniel Denis [2005]. *The Early Origins and Development of the Scatterplot*. Journal of the History of the Behavioral Sciences 41.2 (2005), pages 103–130. doi:10.1002/jhbs.20078. https://datavis.ca/papers/friendly-scat.pdf (cited on page 3).

Girardin, Luc and Dominique Brodbeck [2001]. *Interactive Visualization of Prices and Earnings Around the Globe*. Proc. IEEE Symposium on Information Visualization (InfoVis 2001) (San Diego, California, USA). 21 Oct 2001. http://download.macrofocus.com/publications/infovis2001.pdf (cited on page 15).

Gorman, Kristen B, Tony D Williams, and William R Fraser [2014]. *Ecological Sexual Dimorphism and Environmental Variability Within a Community of Antarctic Penguins (Genus Pygoscelis)*. PloS one 9.3 (05 Mar 2014), e90081. doi:10.1371/journal.pone.0090081 (cited on page 1).

Hoffman, Patrick, Georges Grinstein, Kenneth Marx, Ivo Grosse, and Eugene Stanley [1997]. *DNA Visual and Analytic Data Mining*. Proc. 8th IEEE Conference on Visualization (Vis '97) (Phoenix, Arizona, USA). 24 Oct 1997, pages 437–441. doi:10.1109/VISUAL.1997.663916 (cited on page 6).

Horst, Allison Marie, Alison Presmanes Hill, and Kristen B. Gorman [2020]. *palmerpenguins: Palmer Archipelago (Antarctica) Penguin Data*. R package version 0.1.0. 25 Jul 2020. doi:10.5281/zenodo.3960218. https://allisonhorst.github.io/palmerpenguins/ (cited on page 1).

Hunter, John D. [2007]. *Matplotlib: A 2D graphics environment*. Computing in Science & Engineering 9.3 (2007), pages 90–95. doi:10.1109/MCSE.2007.55 (cited on page 3).

Inselberg, Alfred [2009]. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 01 Sep 2009. 554 pages. ISBN 0387215077 (cited on pages 10, 13).

Inselberg, Alfred and Bernard Dimsdale [1990]. *Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry*. Proc. 1st IEEE Conference on Visualization (Vis '90) (San Francisco, California, USA). 23 Oct 1990, pages 361–378. doi:10.1109/VISUAL.1990.146402. https://ifs.tuwien.ac.at/~mlanzenberger/teaching/ps/ws07/stuff/00146402.pdf (cited on page 10).

Kandogan, Eser [2001]. *Visualizing Multi-Dimensional Clusters, Trends, and Outliers Using Star Coordinates*. Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001) (San Francisco, California, USA). 26 Aug 2001, pages 107–116. doi:10.1145/502512.502530 (cited on page 5).

Koytek, Philipp, Charles Perin, Jo Vermeulen, Elisabeth André, and Sheelagh Carpendale [2017]. *MyBrush: Brushing and Linking with Personal Agency*. IEEE Transactions on Visualization and Computer Graphics 24.1 (29 Aug 2017), pages 605–615. ISSN 1077-2626. doi:10.1109/TVCG.2017.2743859 (cited on page 18).

Macrofocus [2015]. *InfoScope*. 19 Aug 2015. https://web.archive.org/web/20160429015130/http://www.macrofocus.com/public/products/infoscope/ (cited on page 15).

Macrofocus [2022]. *High-D*. 05 Dec 2022. https://www.high-d.com/ (cited on page 16).

McInnes, Leland, John Healy, and James Melville [2018]. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. arXiv (09 Feb 2018). ISSN 2331-8422. doi:10.48550/arXiv.1802.03426 (cited on page 9).

McInnes, Leland, John Healy, Nathaniel Saul, and Lukas Grossberger [2018]. *UMAP: Uniform Manifold Approximation and Projection*. The Journal of Open Source Software 3.29 (2018), page 861. https://umap-learn.readthedocs.io/ (cited on page 3).

McKinney, Wes [2010]. *Data Structures for Statistical Computing in Python*. Pro. 9th Python in Science Conference (SciPy 2010) (Austin, Texas, USA). Volume 445. 28 Jun 2010, pages 51–56. https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf (cited on page 3).

Morrison, Alistair, Greg Ross, and Matthew Chalmers [2003]. *Fast Multidimensional Scaling Through Sampling, Springs and Interpolation*. Information Visualization 2.1 (01 Mar 2003), pages 68–77. doi:10.1057/palgrave.ivs.9500040. https://dcs.gla.ac.uk/~matthew/papers/JInfoVis.pdf (cited on page 8).

Neuhold, Lukas, Ridvan Aydin, and Georg Regitnig [2020]. *The Radial Projection Explorer*. 29 Jun 2020. https://courses.isds.tugraz.at/ivis/projects/ss2020/ivis-ss2020-g4-project-radial-projection-explorer.pdf (cited on pages 5–7).

Nguyen, Quang Vinh, Yu Qian, MaoLin Huang, and JiaWan Zhang [2013]. *TabuVis: A Tool for Visual Analytics Multidimensional Datasets*. Science China Information Sciences 56.5 (24 May 2013),

pages 1–12. ISSN 1869-1919. doi:10.1007/s11432-013-4870-1. https://researchgate.net/publication/257686743_TabuVis_A_tool_for_visual_analytics_multidimensional_datasets (cited on page 16).

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay [2011]. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research 12 (2011), pages 2825–2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html (cited on page 3).

Romesburg, H. Charles [1984]. *Cluster Analysis for Researchers*. Lifetime Learning Publications, 01 Jan 1984. ISBN 0534032486 (cited on page 12).

Samariya, Durgesh [2020]. *Premier League Player Stats Data*. 27 Jul 2020. https://kaggle.com/datasets/themlphdstudent/premier-league-player-stats-data (cited on pages 1, 15, 17–18).

Van der Maaten, Laurens and Geoffrey Hinton [2008]. *Visualizing Data using t-SNE*. Journal of Machine Learning Research 9.11 (Nov 2008), pages 2579–2605. ISSN 1532-4435. https://jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf (cited on page 9).

Ward, Matthew O. [1994]. *Xmdvtool: Integrating Multiple Methods for Visualizing Multivariate Data*. Proc. 5th IEEE Conference on Visualization (Vis '94) (Washington, DC, USA). 1994, pages 326–333. doi:10.1109/VISUAL.1994.346302 (cited on page 13).

Wattenberg, Martin, Fernanda Viégas, and Ian Johnson [2016]. *How to Use t-SNE Effectively*. Distill (18 Oct 2016). doi:10.23915/distill.00002 (cited on page 9).

Weaver, Chris [2014]. *Improvise*. 21 Sep 2014. https://cs.ou.edu/~weaver/improvise/ (cited on page 18).

XDAT [2020]. *XDAT*. 26 Aug 2020. https://xdat.org/ (cited on page 15).

Yi, Ji Soo, Rachel Melton, John Stasko, and Julie A. Jacko [2005]. *Dust & Magnet: Multivariate Information Visualization Using a Magnet Metaphor*. Information Visualization 4.4 (11 Apr 2005), pages 239–256. doi:10.1057/palgrave.ivs.9500099. https://faculty.cc.gatech.edu/~stasko/papers/iv05-dnm.pdf (cited on page 6).