Descriptive Multivariate Analysis

In real-life data sets, the number of attributes often exceeds two, ranging from tens to hundreds or even more. Multivariate analysis is used to analyze data sets with more than two attributes. Similar to univariate and bivariate analysis, multivariate analysis can utilize frequency tables, statistical measures, and plots, which can be adapted for use with an arbitrary number of attributes.

Methods for Multivariate Analysis

1. **Frequency Tables:**
   - Frequency tables can be extended to include multiple attributes. For example, a cross-tabulation (contingency table) can be used to show the relationship between two categorical attributes.
   - For numerical attributes, histograms or frequency distributions can be used to visualize the distribution of values across multiple attributes.
2. **Statistical Measures:**
   - Central tendency measures such as mean, median, and mode can be calculated for each attribute individually or for combinations of attributes.
   - Measures of dispersion like variance and standard deviation can help understand the spread of values across multiple attributes.
3. **Visualization:**
   - Scatter plots can be extended to visualize relationships between more than two numerical attributes. For example, a 3D scatter plot can show the relationship between three numerical attributes.
   - Parallel coordinate plots can be used to visualize relationships between multiple numerical attributes simultaneously.
4. **Correlation and Covariance:**
   - Correlation and covariance matrices can provide insights into the relationships between pairs of numerical attributes.
   - Positive values indicate a positive relationship, negative values indicate a negative relationship, and values close to zero indicate no relationship.
5. **Cluster Analysis:**
   - Cluster analysis can be used to group similar observations based on multiple attributes.
   - This technique helps in identifying natural groupings or patterns in the data.
6. **Factor Analysis:**
   - Factor analysis can be used to identify underlying factors or latent variables that explain the patterns in the data.
   - It helps in reducing the dimensionality of the data by identifying a smaller set of factors that capture most of the variation in the data.

Example: Contact Data Set

- Consider a data set with attributes such as contact name, maximum temperature, weight, height, duration of acquaintance, gender, and rating of company.
- Multivariate analysis can be applied to this data set to understand relationships and patterns among these attributes.
- For example, a scatter plot matrix can be used to visualize relationships between pairs of numerical attributes, while a cross-tabulation can show relationships between categorical attributes.

In summary, multivariate analysis extends the principles of univariate and bivariate analysis to datasets with more than two attributes, providing insights into complex relationships and patterns in the data.

3.1 Multivariate Frequencies

Multivariate frequencies can be computed independently for each attribute in a dataset. These frequencies can be represented in a matrix form, where the number of rows corresponds to the number of unique values assumed by each attribute, and the columns represent the frequency values for each attribute. For example, consider the dataset of private contacts with weight and height shown below:

Contact	Maxtemp	Weight	Height	Years	Gender	Company
Andrew	25	77	175	10	M	Good
Bernhard	31	110	195	12	M	Good
Carolina	15	70	172	2	F	Bad
Dennis	20	85	180	16	M	Good
Eve	10	65	168	0	F	Bad
Fred	12	75	173	6	M	Good
Gwyneth	16	75	180	3	F	Bad
Hayden	26	63	165	2	F	Bad
Irene	15	55	158	5	F	Bad
James	21	66	163	14	M	Good
Kevin	30	95	190	1	M	Bad
Lea	13	72	172	11	F	Good
Marcus	8	83	185	3	F	Bad
Nigel	12	115	192	15	M	Good

For each attribute, the following frequency measures can be computed:

- **Absolute Frequency:** The number of occurrences of each value in the dataset.

- **Relative Frequency:** The proportion of occurrences of each value relative to the total number of observations.
- **Absolute Cumulative Frequency:** The cumulative sum of the absolute frequencies from the smallest to the largest value.
- **Relative Cumulative Frequency:** The cumulative sum of the relative frequencies from the smallest to the largest value.

3.2 Multivariate Data Visualization

In multivariate analysis, visualization plays a crucial role in understanding complex relationships among multiple attributes. While many plots from univariate and bivariate analysis can be extended to visualize three or more attributes, new visualization techniques are continuously being developed to handle the complexities of multivariate data.

Extension of Bivariate Plots:

- **Adding a Third Attribute:** When visualizing three attributes, one can still use a bivariate plot by associating the values of the third attribute with the size, color, or shape of the plotted objects. For example, in a scatter plot, the size of each point can represent the value of a third quantitative attribute, as shown in Figure 3.1.
- **Qualitative Third Attribute:** If the third attribute is qualitative, its values can be represented using colors or shapes. Each unique value of the attribute can be mapped to a different color or shape, as shown in Figure 3.2.

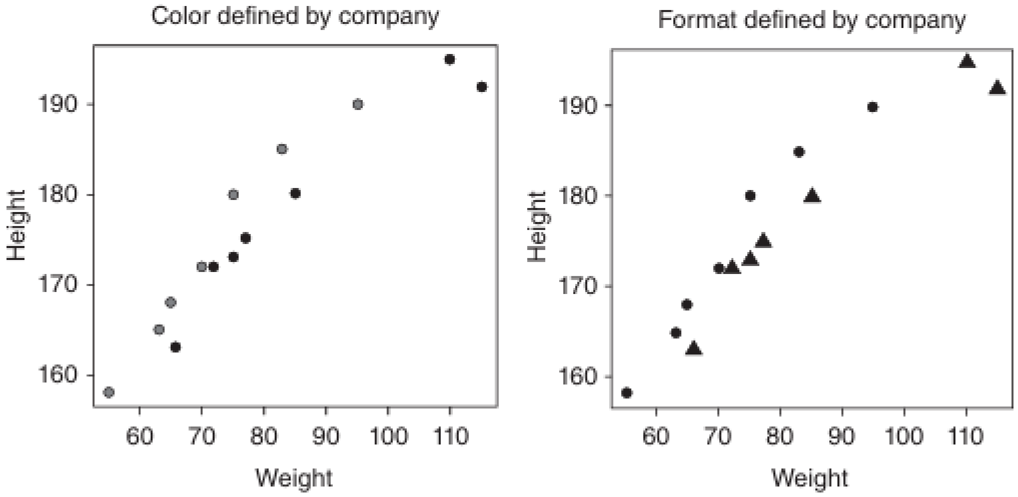


Figure 3.2 Two alternatives for a plot of three attributes, where the the third attribute is qualitative.

Three-Dimensional Plots:

- **Three-Dimensional Scatter Plot:** A direct way to visualize three quantitative attributes is to use a three-dimensional scatter plot, where each axis represents one of the attributes. This allows for the visualization of relationships among the three attributes, as shown in Figure 3.3.

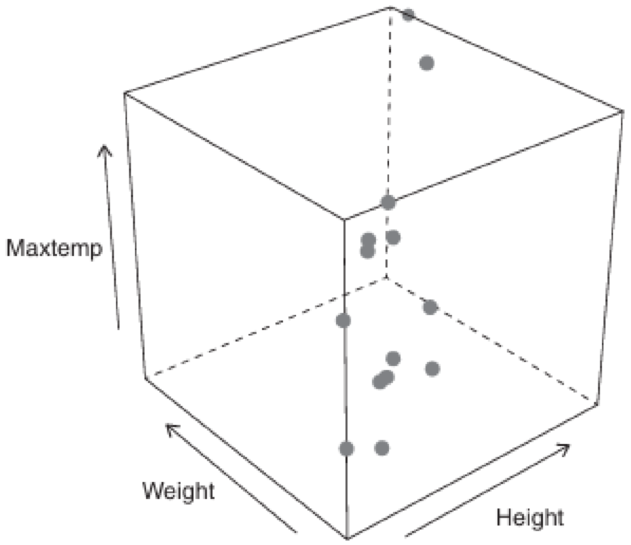
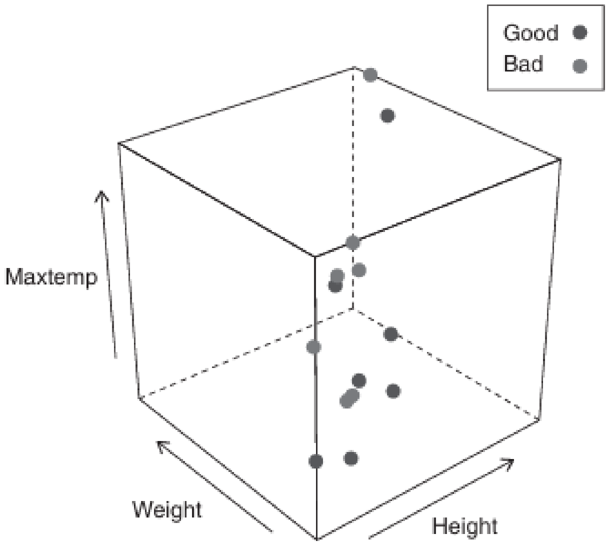


Figure 3.3 Plot for three attributes from the contacts data set.

- **Adding a Fourth Attribute:** To visualize four attributes, you can use color to represent the values of a fourth qualitative attribute, as shown in Figure 3.4.

Figure 3.4 Plot for four attributes of the friends data set using color for the forth attribute.



Beyond Three Attributes:

- **Parallel Coordinates Plot:** For more than three quantitative attributes, a parallel coordinates plot can be used. Each object is represented by a line that connects values on parallel vertical axes, one for each attribute. This allows for the visualization of patterns and relationships among multiple attributes, as shown in Figure 3.5 and Figure 3.6.

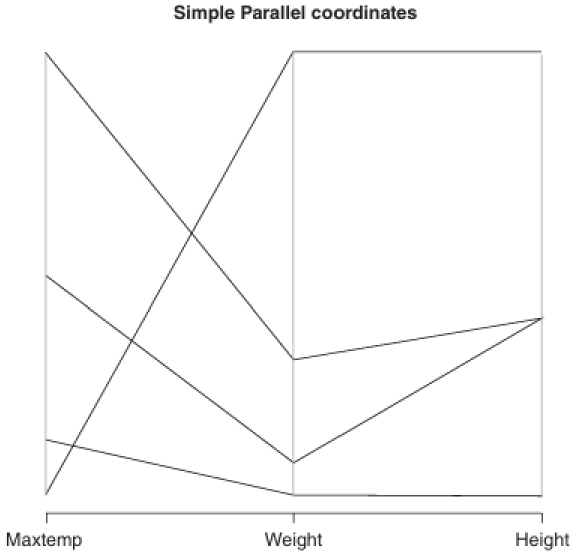


Figure 3.5 Parallel coordinate plot for three attributes.

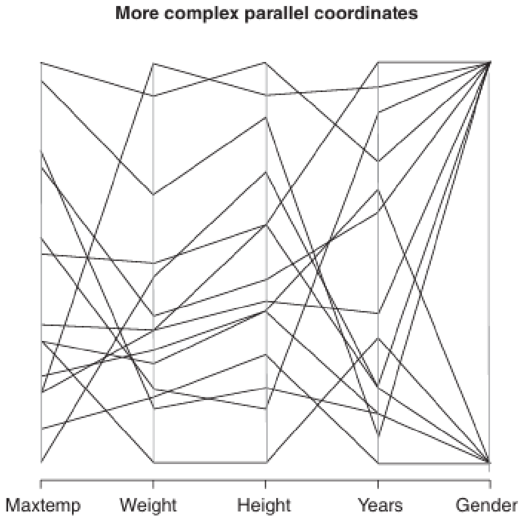


Figure 3.6 Parallel coordinate plot for five attributes.

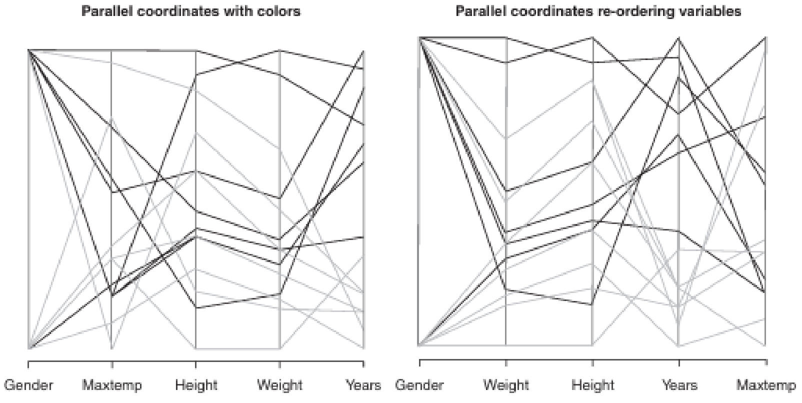


Figure 3.7 Parallel coordinate plots for multiple attributes: left, using a different style of line for contacts who are good and bad company; right, with the order of the attributes changed as well.

- **Star Plot:** Another approach is the star plot, where each attribute is represented by a spoke, and the length of the spoke corresponds to the attribute's value. This allows for the comparison of attribute values across multiple objects, as shown in Figure 3.8 and Figure 3.9.

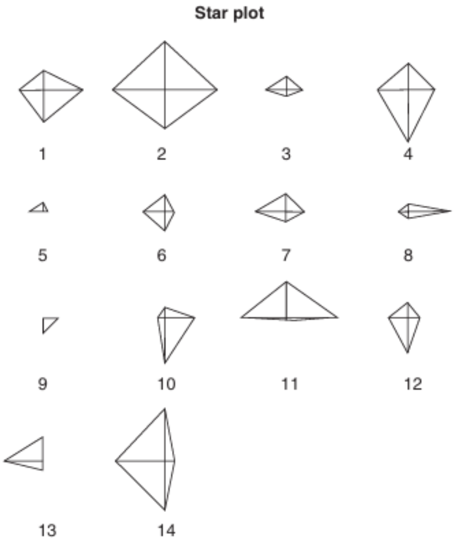


Figure 3.8 Star plot with the value of each attribute for each object in our contacts data set.

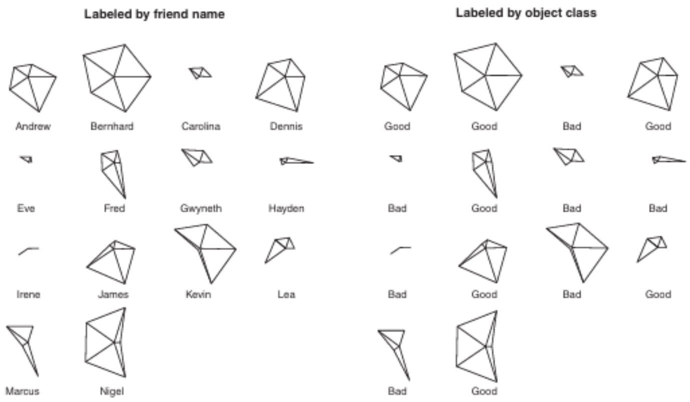


Figure 3.9 Star plot with the value of each attribute for each object in contacts data set.

- **Chernoff Faces:** Chernoff faces are a unique way to represent multivariate data, where different features of a human face represent different attributes. Each face represents an object in the dataset, as shown in Figure 3.10.

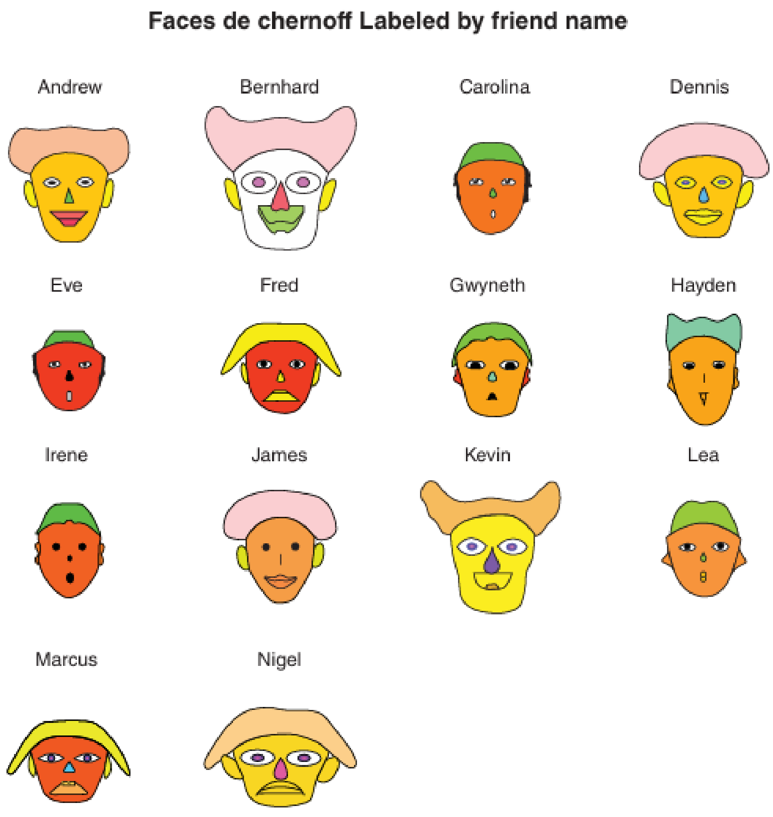


Figure 3.10 Visualization of the objects in our contacts data set using Chernoff faces.

Interactive Visualization:

- Interactive visualization allows users to manipulate plots to explore data more effectively. Users can change viewing angles, zoom in on specific areas, or filter data interactively to gain deeper insights.

Continued Development:

- Data visualization is a rapidly evolving field, with new techniques and approaches being developed to handle the complexities of modern datasets. Interactive and dynamic visualization tools are becoming increasingly popular for exploring and presenting multivariate data.

In multivariate analysis, location statistics are computed independently for each attribute. This means that for each attribute, you calculate its location statistic. These values are typically represented by a numeric vector, where each element corresponds to one attribute.

For example, let's consider the main location statistics for the attributes "maxtemp," "height," "weight," and "years" from a dataset. These statistics can be presented in a table format, where each row represents a statistical measure for the four attributes. Here's an illustration of such a table:

Location statistics for quantitative attributes
Maxtemp

8.00
31.00
18.14
15.00
12.25
15.50
24.00

Additionally, box plots can be used to visualize the distribution of each attribute in a multivariate dataset. For example, you can create a set of box plots, one for each attribute, to show how the values of these attributes vary. Here's an example of a set of box plots for the quantitative attributes from the contacts dataset:

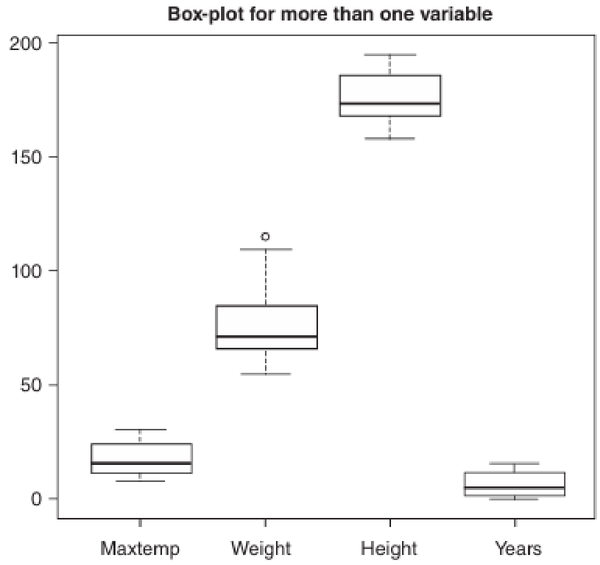


Figure 3.11 Set of box plots, one for each attribute.

In this plot, each box plot represents one attribute, showing the distribution of values, including the median, quartiles, and outliers. It provides a visual summary of the data distribution for each attribute.

3.3.2 Dispersion Multivariate Statistics

In multivariate statistics, dispersion measures like amplitude, interquartile range, mean absolute deviation (MAD), and standard deviation can be independently defined for each attribute.

For example, Table 3.3 shows multivariate dispersion statistics for the attributes "maxtemp," "height," "weight," and "years" from the data set in Table 3.1. Each row in the table represents a statistical measure for each of the four attributes.

These statistics measure the dispersion of each attribute independently. However, we can also measure how the values of one attribute vary with those of another attribute. This relationship can be assessed using covariance or correlation measures. Covariance measures for all pairs of attributes can be represented using a covariance matrix, where the attributes are listed in both the rows and columns in the same order.

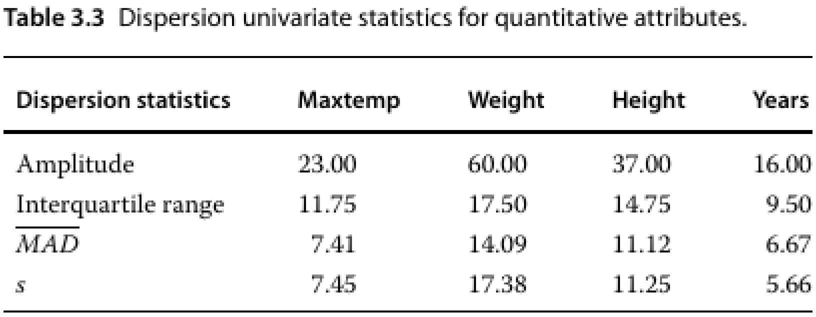


Table 3.3 Dispersion univariate statistics for quantitative attributes.

Dispersion statistics	Maxtemp	Weight	Height	Years
Amplitude	23.00	60.00	37.00	16.00
Interquartile range	11.75	17.50	14.75	9.50
$\overline{MAD}$	7.41	14.09	11.12	6.67
s	7.45	17.38	11.25	5.66

Data Quality and Preprocessing

Introduction

Data quality and preprocessing are fundamental aspects of data analytics that significantly influence the outcomes of any analysis. The reliability of models, charts, and studies hinges on the quality of the data being used. Issues related to data quality can arise from various sources, such as human error, integration of disparate data sets, and the methodologies employed during data collection. Addressing these issues through effective preprocessing is essential to ensure accurate and meaningful results.

Data Quality

The quality of data used in analytics is paramount, as it directly impacts the validity of the findings. Poor-quality data can lead to flawed models and incorrect conclusions, ultimately affecting decision-making processes. Data quality issues are broadly categorized into internal and external factors. Internal factors relate to the measurement process and the collection of information through chosen attributes. External factors encompass errors in the data collection process, including missing values, and the introduction of errors, whether voluntary or involuntary.

Key Data Quality Issues:

1. **Missing Values**:
   - Missing data points can occur due to incomplete data collection or recording errors.
   - Techniques to handle missing values include data imputation, where missing values are estimated and filled, or removing the affected data points entirely.
2. **Inconsistency**:
   - Data inconsistency arises when different sources provide conflicting information, often due to variations in data entry or mismatched formats.
   - Ensuring uniform data formats and implementing validation checks can help reduce inconsistencies.
3. **Redundancy**:
   - Redundant data involves duplicate records that can lead to inefficiencies in storage and processing.
   - Data deduplication processes help identify and eliminate duplicate entries, ensuring a more streamlined dataset.
4. **Noise**:
   - Noise refers to irrelevant or meaningless data that can obscure true patterns in the dataset.
   - Techniques such as smoothing, filtering, and outlier detection are employed to reduce noise and enhance data clarity.
5. **Outliers**:
   - Outliers are data points that significantly deviate from the rest of the dataset, potentially skewing analysis results.
   - Handling outliers involves identifying and possibly removing or transforming these data points to prevent distortion of the analysis.

Ensuring high data quality is crucial for effective data analysis. High-quality data leads to more accurate models and insightful findings. Preprocessing steps such as data cleaning, normalization, and transformation are essential to prepare the data for analysis.

Data Preprocessing Techniques:

1. **Data Cleaning**:
   - Involves identifying and correcting errors in the data, handling missing values, rectifying inconsistencies, and removing duplicate records.
2. **Data Transformation**:

   - Converts data into a suitable format for analysis. This includes normalization (scaling data to a standard range), standardization (transforming data to have a mean of zero and standard deviation of one), and encoding categorical variables.
3. **Dimensionality Reduction**:
   - Reduces the number of variables in the dataset to simplify the analysis while retaining essential information. Methods include Principal Component Analysis (PCA) and feature selection techniques.

By addressing data quality issues and employing robust preprocessing techniques, the reliability and efficiency of data analytics can be significantly enhanced, leading to more accurate and actionable insights.

4.1.1 Missing Values

In practical scenarios, it's common to encounter datasets with missing values in some predictive attributes. Several factors can cause missing values, including:

Delayed Recording: Attribute values recorded only after data collection has started, leaving early records incomplete.

Unknown Values: Attributes unknown at the time of collection.

Human Factors: Distraction, misunderstanding, or refusal during data collection.

Irrelevance: Attributes not required for specific objects.

Non-existence: Absence of a value altogether.

Device Faults: Errors due to faulty data collection devices.

Cost and Difficulty: Challenges in assigning class labels in classification problems.

Since many data analysis techniques cannot handle missing values effectively, preprocessing is essential. Approaches to manage missing values include:

Ignoring Missing Values:

Use only available attribute values for each object, ignoring missing ones.

Modify learning algorithms to accept and process missing values.

Removing Objects:

Discard objects with missing values in all attributes.

Ensure important data is not lost by careful removal.

Estimating Missing Values:

Location Value: Fill missing values with mean, median, or mode based on attribute type.

Mean or median for quantitative and ordinal attributes.

Mode for nominal values.

Class-Specific Estimation: Use only instances from the same class to calculate location statistics.

Predictive Modeling: Use learning algorithms to predict and fill missing values based on other attributes.

Examples:

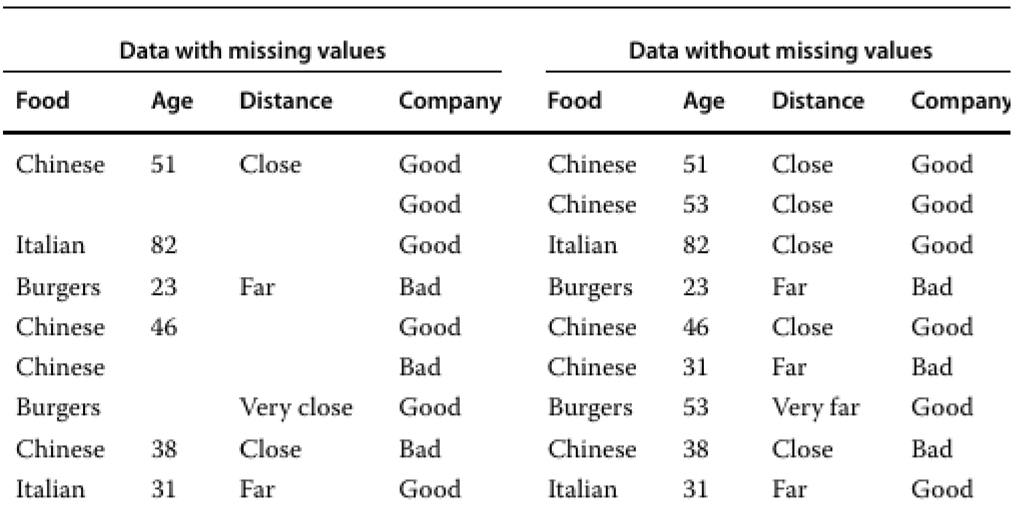
Simple Estimation: Remove objects with many missing values.

Boolean Attribute: Create a new attribute indicating the presence of missing values.

Qualitative Attributes: Use new values indicating missingness.

Quantitative Attributes: Estimate missing values using mean, median, or mode.

Table 4.1 Filling of missing values.



Data with missing values				Data without missing values			
Food	**Age**	**Distance**	**Company**	**Food**	**Age**	**Distance**	**Company**
Chinese	51	Close	Good	Chinese	51	Close	Good
			Good	Chinese	53	Close	Good
Italian	82		Good	Italian	82	Close	Good
Burgers	23	Far	Bad	Burgers	23	Far	Bad
Chinese	46		Good	Chinese	46	Close	Good
Chinese			Bad	Chinese	31	Far	Bad
Burgers		Very close	Good	Burgers	53	Very far	Good
Chinese	38	Close	Bad	Chinese	38	Close	Bad
Italian	31	Far	Good	Italian	31	Far	Good

4.1.2 Redundant Data

Redundant data represents excess information that doesn't add value, often due to very similar or duplicate objects. Redundancy can result from:

Minor Mistakes or Noise: Similar addresses with slight variations in names.

Duplicate Data: Identical data entries repeated in the dataset.

Preprocessing techniques aim to identify and remove redundant data to prevent skewed analyses. For instance:

Deduplication: Remove duplicate entries.

Attribute Redundancy: Detect and handle predictive attributes that can be derived from others.

Example:

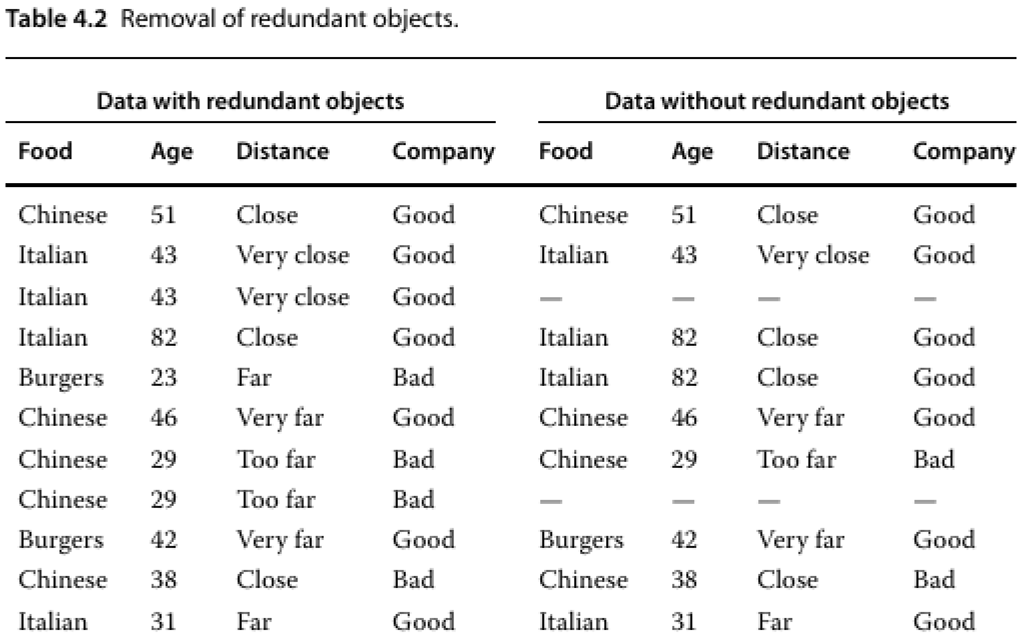


Table 4.2 Removal of redundant objects.

Data with redundant objects				Data without redundant objects			
Food	**Age**	**Distance**	**Company**	**Food**	**Age**	**Distance**	**Company**
Chinese	51	Close	Good	Chinese	51	Close	Good
Italian	43	Very close	Good	Italian	43	Very close	Good
Italian	43	Very close	Good	—	—	—	—
Italian	82	Close	Good	Italian	82	Close	Good
Burgers	23	Far	Bad	Italian	82	Close	Good
Chinese	46	Very far	Good	Chinese	46	Very far	Good
Chinese	29	Too far	Bad	Chinese	29	Too far	Bad
Chinese	29	Too far	Bad	—	—	—	—
Burgers	42	Very far	Good	Burgers	42	Very far	Good
Chinese	38	Close	Bad	Chinese	38	Close	Bad
Italian	31	Far	Good	Italian	31	Far	Good

Table 4.2: Illustrates data with and without redundant objects, highlighting the impact of removing duplicates.

4.1.3 Inconsistent Data

Inconsistent values in a dataset reduce the quality of models derived from machine learning algorithms. Inconsistencies can appear in both predictive and target attributes and may result from:

Data Entry Errors: Mistakes or fraud during data entry, such as mismatched zip codes and city names.

Ambiguities: Different target values for objects with the same predictive attributes due to labeling errors.

Strategies to manage inconsistent values include:

Detection: Identify inconsistencies through known attribute relationships (e.g., one attribute must be larger than another).

Treatment: Treat inconsistencies as missing values to simplify handling.

Example:

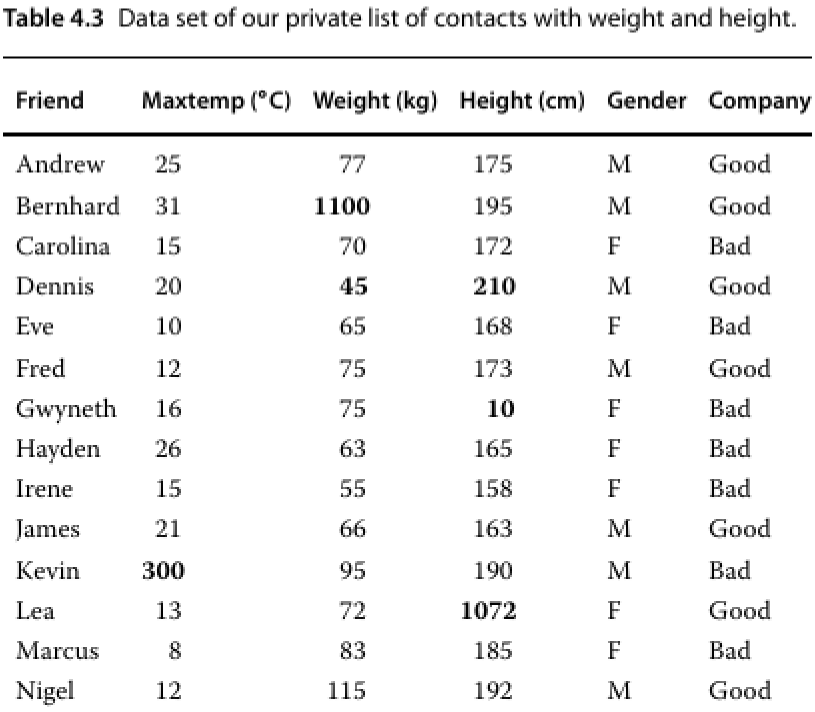


Table 4.3 Data set of our private list of contacts with weight and height.

Friend	Maxtemp (°C)	Weight (kg)	Height (cm)	Gender	Company
Andrew	25	77	175	M	Good
Bernhard	31	**1100**	195	M	Good
Carolina	15	70	172	F	Bad
Dennis	20	**45**	**210**	M	Good
Eve	10	65	168	F	Bad
Fred	12	75	173	M	Good
Gwyneth	16	75	**10**	F	Bad
Hayden	26	63	165	F	Bad
Irene	15	55	158	F	Bad
James	21	66	163	M	Good
Kevin	**300**	95	190	M	Bad
Lea	13	72	**1072**	F	Good
Marcus	8	83	185	F	Bad
Nigel	12	115	192	M	Good

Table 4.3: Demonstrates a dataset with and without inconsistent values, showing the correction of data inconsistencies.

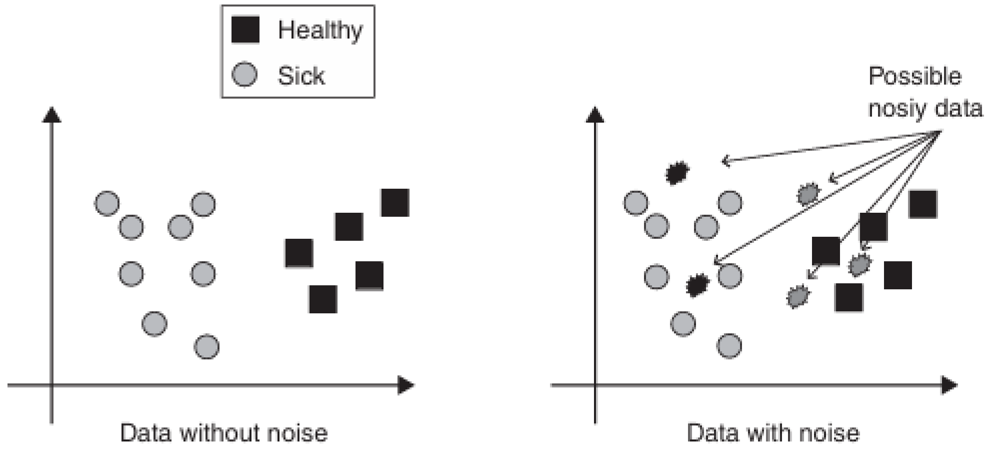


Figure 4.1 Data set with and without noise.

Summary

Addressing data quality issues through effective preprocessing is essential for accurate data analysis. Techniques to handle missing values, redundant data, and inconsistencies include estimation, deduplication, and error detection. Ensuring high-quality data enhances the reliability of analytical models and insights.

Noisy Data

Definition: Data that doesn't meet expected standards, caused by errors or contamination.

Detection: Use of classification algorithms or noise filters in preprocessing.

Filter Focus: Mainly developed for target attributes due to complexity in predictive attributes.

Example: k-NN algorithm-based filters identify noisy objects based on similarity to others.

Outliers

Definition: Values or objects significantly different from others in a dataset.

Identification: Important in anomaly detection; outliers can be valid but unconventional.

Detection Method: Use of interquartile range (IQR) for quantitative attributes.

IQR Calculation: IQR = Q3 - Q1; outliers are values beyond Q1 - 1.5 × IQR or Q3 + 1.5 × IQR.

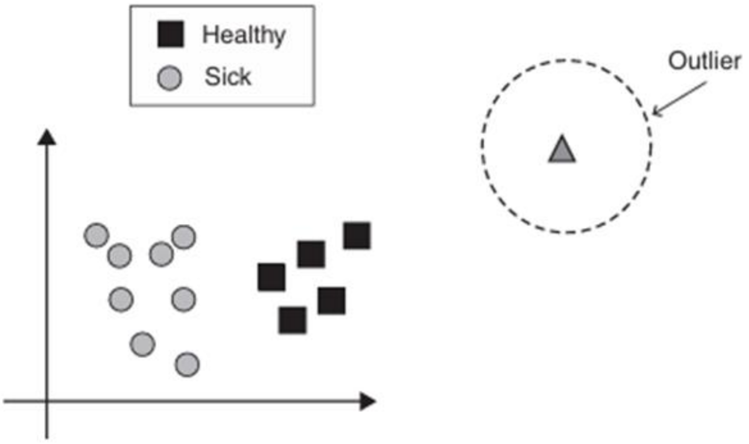


Figure 4.2 Data set with outliers.

Converting to a Different Scale Type

Scale Types: Some ML algorithms require specific scale types (qualitative or quantitative).

Conversion: Possible to convert data between qualitative and quantitative scales.

Example Data Set:

Attributes: Favorite food, age, distance, and company quality.

Example Entries:

Favorite Food: Chinese, Italian, Burgers

Age: Numerical values

Distance: Qualitative (e.g., Close, Far, Very Far)

Company: Qualitative (e.g., Good, Bad)

Conversion:

Qualitative to Quantitative:

Example: Converting distance from qualitative (Close, Far) to quantitative (e.g., Close = 1, Far = 2).

Quantitative to Qualitative:

Example: Grouping ages into categories (e.g., 0-20 = Young, 21-40 = Middle-aged, 41+ = Senior).

Importance:

Ensuring Compatibility: Convert data to match algorithm requirements.

Enhancing Analysis: Allows for more effective data analysis and model training.

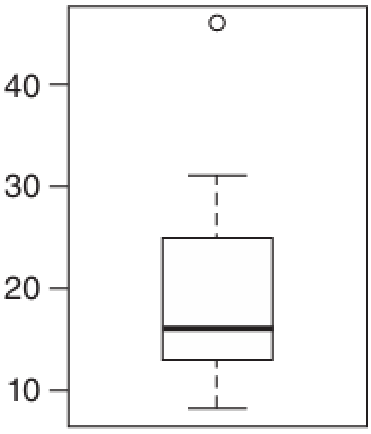
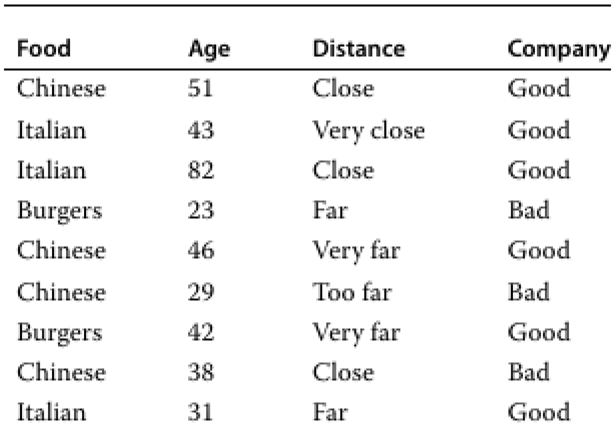


Figure 4.3 Outlier detection based on the interquartile range distance.

Table 4.4 Food preferences of our colleagues.



Food	Age	Distance	Company
Chinese	51	Close	Good
Italian	43	Very close	Good
Italian	82	Close	Good
Burgers	23	Far	Bad
Chinese	46	Very far	Good
Chinese	29	Too far	Bad
Burgers	42	Very far	Good
Chinese	38	Close	Bad
Italian	31	Far	Good

Converting Nominal to Relative

4.2.1 Converting Nominal to Relative

Nominal scale does not assume an order between its values.

Conversion to relative or binary values preserves this information.

"1-of-n" Conversion

Also known as canonical or one-attribute-per-value conversion.

Transforms n values of a nominal attribute into n binary attributes.

Each binary attribute has only two values, 0 or 1.

Example:

Original Attribute: Favorite Food (Nominal)

Values: Chinese, Italian, Burgers

Conversion:

Chinese -> [1, 0, 0]

Italian -> [0, 1, 0]

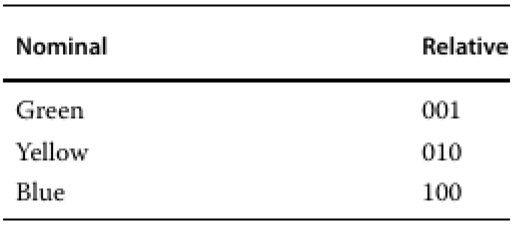
Burgers -> [0, 0, 1]

Significance:

Maintaining Information: Preserves the distinction between different nominal values.

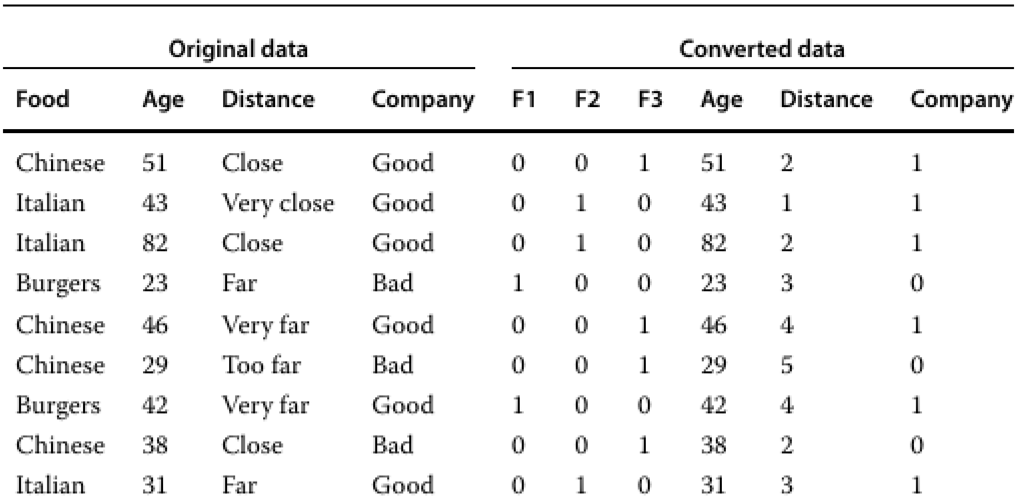
Enhancing Analysis: Enables algorithms to process nominal data effectively.

Table 4.5 Conversion from nominal scale to relative scale.



Nominal	Relative
Green	001
Yellow	010
Blue	100

Table 4.6 Conversion from the nominal scale to binary values.



Original data				Converted data					
Food	Age	Distance	Company	F1	F2	F3	Age	Distance	Company
Chinese	51	Close	Good	0	0	1	51	2	1
Italian	43	Very close	Good	0	1	0	43	1	1
Italian	82	Close	Good	0	1	0	82	2	1
Burgers	23	Far	Bad	1	0	0	23	3	0
Chinese	46	Very far	Good	0	0	1	46	4	1
Chinese	29	Too far	Bad	0	0	1	29	5	0
Burgers	42	Very far	Good	1	0	0	42	4	1
Chinese	38	Close	Bad	0	0	1	38	2	0
Italian	31	Far	Good	0	1	0	31	3	1

Converting Ordinal to Relative or Absolute

4.2.2 Converting Ordinal to Relative or Absolute

When converting ordinal values, the process is more intuitive compared to nominal values, as ordinal values inherently possess an order. To convert ordinal values to a quantitative scale, we can map them to natural numbers. Starting with 0 for the smallest ordinal value, each subsequent ordinal value is assigned the next natural number in sequence. This conversion allows us to maintain the ordinal relationship between values, ensuring that larger values correspond to larger numbers.

Converting to Binary Values

In certain scenarios, algorithms may require binary values. To convert ordinal values to binary, we can use coding techniques such as Gray code or thermometer code. Gray code ensures that each two consecutive ordinal values differ by only one binary digit. On the other hand, the thermometer code starts with a binary vector containing only 0 values. As the ordinal value increases, one 0 value is substituted by 1 from right to left in the binary vector.

Example:

Consider an ordinal attribute representing sizes: Small, Medium, Large, and Very Large. Converting these ordinal values to natural numbers would result in Small->0, Medium->1, Large->2, and Very Large->3.

Absolute vs. Relative Scale

In an absolute scale, the smallest ordinal value is mapped to 0, starting the scale from this point. However, if we want to use the values in a relative manner, we can start the natural numbers from values larger than 0.

Significance:

Maintaining Order: The conversion preserves the ordinal order, ensuring that larger ordinal values correspond to larger natural numbers.

Binary Conversion: Enables algorithms that require binary values to process ordinal data, expanding the applicability of ordinal attributes in machine learning tasks.

Converting Relative or Absolute to Ordinal or Nominal

4.2.3 Converting Relative or Absolute to Ordinal or Nominal

Quantitative values can be converted to nominal or ordinal values through a process known as "discretization." This process is essential when a learning algorithm can only handle qualitative values or when there is a need to reduce the number of quantitative values.

Steps in Discretization

Definition of Bins: The number of qualitative values (bins) is determined by the data analyst. Each bin corresponds to an interval of quantitative values.

Association with Intervals: The intervals of values are associated with each bin using an algorithm. This association can be based on width or frequency.

Association Methods

By Width: Intervals have the same range, ensuring an equal difference between the largest and smallest values in each interval.

By Frequency: Intervals contain an equal number of values, ensuring a similar distribution of values in each interval.

Example:

Consider the conversion of nine quantitative values (2, 3, 5, 7, 10, 15, 16, 19, 20) into three bins, denoted as A, B, and C. The conversion is done using both association by width and association by frequency.

Significance:

Data Preparation: Discretization prepares data for algorithms that can only process qualitative values, expanding the range of algorithms that can be applied to the dataset.

Dimensionality Reduction: Reducing the number of quantitative values simplifies the dataset, making it easier to analyze and interpret.

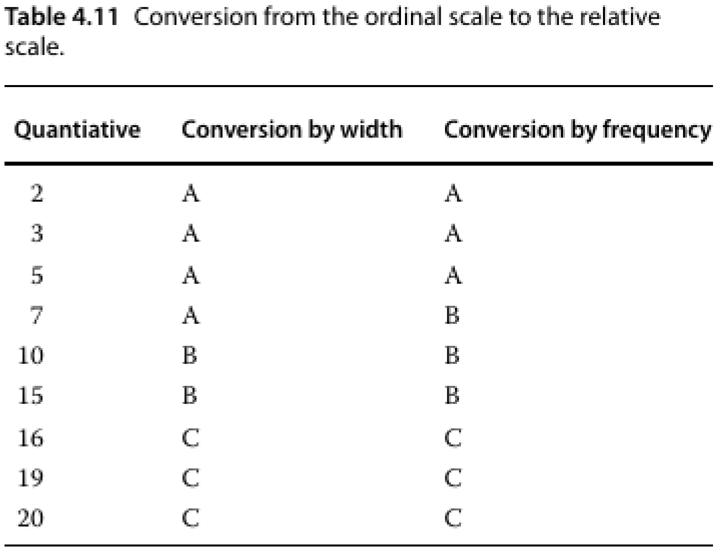


Table 4.11 Conversion from the ordinal scale to the relative scale.

Quantiative	Conversion by width	Conversion by frequency
2	A	A
3	A	A
5	A	A
7	A	B
10	B	B
15	B	B
16	C	C
19	C	C
20	C	C

e concept of converting data from one scale to another, particularly in the context of normalization for distance measures. Here's a summary of the key points:

- **Purpose**: Converting data to a different scale is necessary in various situations, such as when using distance measures like Euclidean distance. Normalization is done to have different attributes expressed on the same scale.
- **Example**: Three friends have age and education values. When calculating the Euclidean distance between them, the results vary based on whether ages are

expressed in years or decades, highlighting the influence of scale on similarity calculations.

- **Normalization Methods**:
    1. **Min-Max Rescaling**: Converts numerical values to a given interval (e.g., [0.0, 1.0]) by subtracting the smallest value from all values and dividing by the amplitude (difference between max and min values).
    2. **Standardization**: Subtracts the average of attribute values and then divides by the standard deviation, resulting in a new average of 0.0 and standard deviation of 1.0.
- **Implementation**: Normalization should be done for each attribute individually. It's a typical preprocessing task in the data preparation phase and helps avoid issues like the dominance of certain attributes in distance calculations.

Here's an example showing the application of min-max scaling and standardization to age and education attributes:

Friend	Age	Education	Rescaled Age	Rescaled Education
Bernhard	43	2.0	1.0	0.0
Gwyneth	38	4.2	0.0	1.0
James	42	4.0	0.8	0.91

Friend	Age	Education	Rescaled Age	Rescaled Education
Bernhard	43	2.0	0.76	-1.15
Gwyneth	38	4.2	-1.13	0.66
James	42	4.0	0.38	0.49

Data transformation

Data Transformation

Introduction

- Data transformation is a critical step in data analysis.
- It involves converting data into a more suitable form for analysis or modeling.

Purpose

- Simplify analysis.
- Enable the use of specific modeling techniques.
- Improve data summarization.

Common Transformations

1. **Logarithmic Transformation**
    - Used for skewed distributions.
    - Reduces skewness.

    - Makes interpretation of highly skewed data easier.

2. **Conversion to Absolute Values**
    - Useful when the magnitude of a value is more important than its sign.
    - Helps focus on the size of the value, irrespective of its positive or negative nature.

Example

- Consider a dataset with income and dinner expenses of friends.
- If the income distribution is highly skewed, a log transformation can make the data more interpretable.
- After transformation, the data become more spread out, aiding visualization and interpretation.

Table: Income and Dinner Expenses of Friends

Friend	Income	Dinner Expenses
Andrew	17,000	2,200
Bernhard	53,500	4,500
Carolina	69,000	6,000
Dennis	72,000	7,100
Eve	125,400	10,800
Fred	89,400	7,100
Gwyneth	58,750	6,000
Hayden	108,800	9,000
Irene	97,200	9,600
James	81,000	7,400
Kevin	21,300	2,500
Lea	138,400	13,500
Marcus	830,000	92,000
Nigel	1,000,000	120,500

Conclusion

- Data transformation is a powerful tool in data analysis.
- It improves data summarization and enables more effective analysis and modeling.

This example table illustrates the income and dinner expenses of friends, showcasing how data transformation techniques can be applied to make the data more interpretable and suitable for analysis.

Summary:

Data transformation is a crucial step in data analysis, helping to simplify complex data and make it more suitable for analysis or modeling. Two common transformations are logarithmic functions and conversion to absolute values. Logarithmic functions are used for skewed distributions to reduce skewness and make data interpretation easier. Conversion to absolute values is useful when the magnitude of a value is more important than its sign. These transformations improve data summarization and enable more effective analysis and modeling.

4.5 Dimensionality Reduction

Dimensionality reduction techniques like Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Multidimensional Scaling (MDS) are essential for handling datasets with a large number of attributes. These techniques offer several benefits, such as reducing training time, improving performance of machine learning algorithms, simplifying model interpretation, and enabling easier data visualization.

PCA, for instance, is widely used to reduce the dimensionality of datasets by combining original attributes into new, fewer components that retain most of the information present in the original data. It works by linearly projecting the data onto a new set of attributes, called principal components, which are ranked based on their variance. The components with the highest variance are selected, resulting in a reduced-dimensional representation of the data.

ICA, on the other hand, assumes that the original attributes are statistically independent and tries to decompose the data into independent components. It is particularly useful for noisy datasets where reducing higher-order statistics, like kurtosis, can lead to better results compared to PCA.

MDS, while also involving a linear projection of the data, differs from PCA and ICA in that it uses the distances between pairs of objects in the dataset instead of the attribute values. This makes it suitable for cases where extracting relevant features to represent objects is challenging, as it only requires information on how similar pairs of objects are.

Overall, these dimensionality reduction techniques play a crucial role in simplifying complex datasets, making them more manageable and facilitating better analysis and decision-making processes.

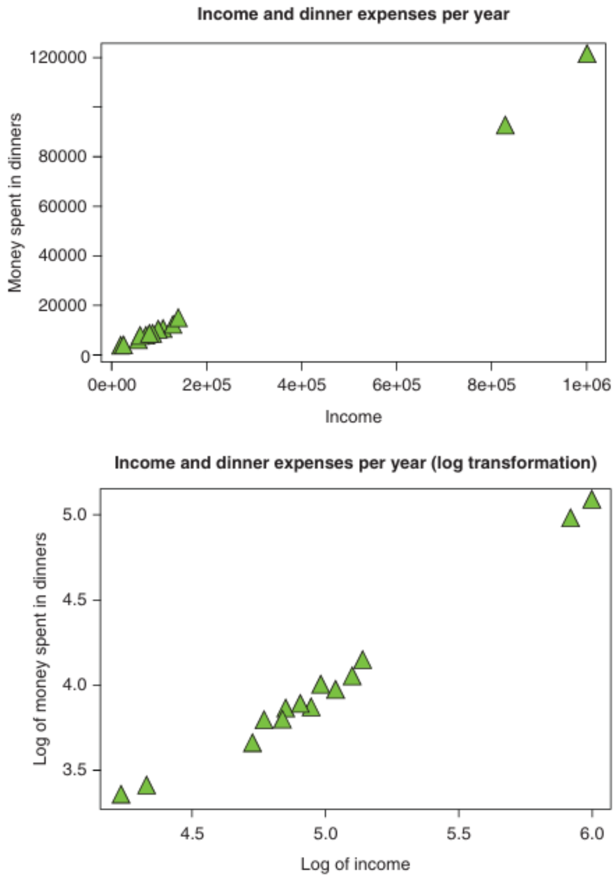


Figure 4.4 Two alternatives for a plot for three attributes the last of which is qualitative.

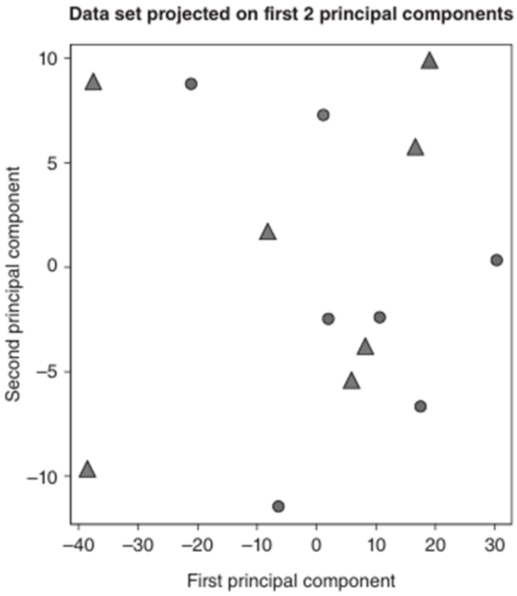


Figure 4.5 Principal components obtained by PCA for the short version of the contacts data set.

Attribute selection

Attribute selection is another approach to reduce the dimensionality of a dataset, offering benefits such as speeding up the learning process and improving predictive performance. This approach involves selecting a subset of attributes from the original set, and it can be categorized into filters, wrappers, and embedded methods.

Filters focus on finding simple, individual relationships between the predictive attributes and the target attribute, ranking the attributes based on this relationship. They are computationally efficient but may overlook complex interactions between attributes.

Wrappers, on the other hand, use a classifier to guide the attribute selection process. They select the subset of attributes that provides the highest predictive performance for the classifier, which can capture complex attribute interactions but are computationally more expensive than filters.

Embedded methods perform attribute selection as an internal procedure of a predictive algorithm. For example, decision tree induction algorithms can perform embedded attribute selection as part of their model-building process.

Search strategies play a crucial role in attribute selection. Exhaustive search evaluates all possible attribute subsets, which can be computationally expensive for large datasets. Greedy sequential techniques, such as forward selection and backward selection, are more efficient alternatives. Forward selection starts with an empty set of attributes and adds one attribute at a time, while backward selection starts with all attributes and removes them one by one. These strategies iteratively improve the attribute subset until a stopping criterion is met.

In conclusion, attribute selection is a valuable technique for reducing the dimensionality of datasets, improving the efficiency of machine learning algorithms, and enhancing predictive performance. Different methods and search strategies can be employed based on the specific characteristics of the dataset and the goals of the analysis.

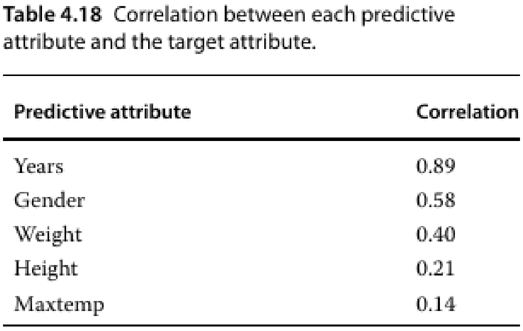


Table 4.18 Correlation between each predictive attribute and the target attribute.

Predictive attribute	Correlation
Years	0.89
Gender	0.58
Weight	0.40
Height	0.21
Maxtemp	0.14

Table 4.19 Predictive performance of a classifier for each predictive attribute.

Predictive attribute	Predictive performance
Years	0.78
Height	0.46
Gender	0.42
Weight	0.38
Maxtemp	0.14