Hochschule Merseburg

Faculty

Wirtschaftswissenschaften und Informationswissenschaften

Masterthesis

Multivariate Control Charts

By:

Muhammad Akhiruddin El Islami

Matrikel Number: 28631

Master Industrial Engineering

Supervised by:

Prof. Dr. rer. nat. habil. Eckhard Liebscher

Dr. Benjamin Wacker

Merseburg, 28.05.2024

Abstract

In modern industry, maintaining and improving quality standards is a must to increase competitiveness and also achieve customer satisfaction. Statistical Process Control (SPC), a methodology for both monitoring and controlling a process through control charts, is one important option for achieving quality improvement goals. Univariate control charts provide access to monitoring a single quality characteristic, whereas for monitoring from multiple variables, multivariate control charts are the answer. The application of multivariate control charts makes it possible to monitor several interrelated quality characteristics, to find out a comprehensive picture of the process behavior.

This thesis explores the theoretical basics relating to multivariate control charts, focusing on Hotelling's T^2, MCUSUM (Cumulative Sum) and MEWMA (Exponentially Weighted Moving Average), as well as their application with practice data examples. These graphs are very powerful for detecting small shifts in the process mean vector and changes in process variability, which are often missed by univariate graphs.

The research begins with an exploration of the theoretical foundations, including the principles of statistical quality control, the multivariate normal distribution, and the statistical process control framework. A thorough examination of various control charts, such as Shewhart, CUSUM, and EWMA control charts, forms the basis for understanding the complexity and necessity of the multivariate approach.

The implementation section provides comprehensive guidance on the software and programming languages used for the analysis. In particular, the statistical capabilities of Statistica to visualize univariate control charts, and the flexibility of the R programming language to illustrate multivariate control charts are discussed. Detailed explanation illustrate how these tools can be used to create and interpret control charts.

In the applications section, the application of Statistica to visualize several examples of univariate charts will be demonstrated. However, but the main focus is on illustrating multivariate control charts using the R programming language. Hotelling's T^2, MCUSUM as well as MEWMA will be shown in several cases, namely normal, jump and shift.

Finally, the thesis concludes with a conclusion that summarizes all the findings of the thesis and recommendations for future research directions. The ability of multivariate control charts to detect subtle changes in process behavior further highlight the importance of control charts in quality control.

Zusammenfassung

In der modernen Industrie ist die Aufrechterhaltung und Verbesserung von Qualitätsstandards ein Muss, um die Wettbewerbsfähigkeit zu steigern und die Kundenzufriedenheit zu erreichen. Die statistische Prozesskontrolle (SPC), eine Methode zur Überwachung und Steuerung eines Prozesses mittels Kontrollkarten, ist eine wichtige Option zur Erreichung von Qualitätsverbesserungszielen. Univariate Kontrollkarten bieten Zugang zur Überwachung eines einzelnen Qualitätsmerkmals, während für die Überwachung von mehreren Variablen multivariate Kontrollkarten die Lösung sind. Die Anwendung von multivariaten Kontrollkarten ermöglicht es, mehrere miteinander verbundene Qualitätsmerkmale zu überwachen, um ein umfassendes Bild des Prozessverhaltens zu erhalten.

In dieser Arbeit werden die theoretischen Grundlagen zu multivariaten Kontrollkarten mit den Schwerpunkten Hotelling‘s T^2, MCUSUM (Cumulative Sum) und MEWMA (Exponentially Weighted Moving Average) sowie deren Anwendung anhand von Praxisdatenbeispielen untersucht. Diese Diagramme sind sehr leistungsfähig, um kleine Verschiebungen im Mittelwertvektor des Prozesses und Änderungen der Prozessvariabilität zu erkennen, die von univariaten Diagrammen oft übersehen werden.

Die Untersuchung beginnt mit einer Erforschung der theoretischen Grundlagen, einschließlich der Prinzipien der statistischen Qualitätskontrolle, der multivariaten Normalverteilung und des Rahmens für die statistische Prozesskontrolle. Eine gründliche Untersuchung verschiedener Kontrollkarten wie Shewhart-, CUSUM- und EWMA-Kontrollkarten bildet die Grundlage für das Verständnis der Komplexität und Notwendigkeit des multivariaten Ansatzes.

Der Abschnitt über die Implementierung bietet eine umfassende Anleitung zu der für die Analyse verwendeten Software und den Programmiersprachen. Insbesondere werden die statistischen Möglichkeiten von Statistica zur Visualisierung von univariaten Regelkarten und die Flexibilität der Programmiersprache R zur Darstellung von multivariaten Kontrollkarten diskutiert. Detaillierte Erläuterungen veranschaulichen, wie diese Werkzeuge zur Erstellung und Interpretation von Kontrollkarten verwendet werden können.

Im Abschnitt Anwendungen wird die Anwendung von Statistica zur Visualisierung verschiedener Beispiele von univariaten Karten gezeigt. Der Schwerpunkt liegt jedoch auf der Veranschaulichung multivariater Kontrollkarten mit der Programmiersprache R. Hotelling‘s T^2, MCUSUM sowie MEWMA werden in mehreren Fällen, nämlich Normal, Jump und Shift, gezeigt.

Schließlich schließt die Arbeit mit einer Schlussfolgerung, die alle Ergebnisse der Arbeit und Empfehlungen für zukünftige Forschungsrichtungen zusammenfasst. Die Fähigkeit der multivariaten Kontrollkarten, subtile Veränderungen im Prozessverhalten zu erkennen, unterstreicht die Bedeutung von Kontrollkarten in der Qualitätskontrolle.

Table of Contents

List of Abbreviations

BPR	: Business Process Reengineering
CPD	: Change-Point Detection
CUSUM	: Cumulative Sum
EWMA	: Exponentially Weighted Moving Average
IC	: In-Control
LCL	: Lower Control Limit
LWL	: Lower Warning Limit
MCUSUM	: Multivariate Cumulative Sum
MEWMA	: Multivariate Exponentially Weighted Moving Average
OC	: Out-of-Control
SPC	: Statistical Process Control
SQC	: Statistical Quality Control
UCL	: Upper Control Limit
UWL	: Upper Warning Limit

1 Introduction

Statistical Process Control (SPC) drives continuous improvement by consistently monitoring and controlling manufacturing processes, ensuring they operate at peak efficiency for high-quality results. Introduced by William A. Shewart in 1924, SPC rose to prominence during World War II to maintain product quality and safety in munitions and weapons facilities. Post-war, Japan adopted and perfected SPC in manufacturing, leading to its revival in the US during the 1970s to compete with Japanese products. Today, SPC is essential in managing quality amid fluctuating raw material costs and intense competition, shifting the focus from defect detection to trend prevention and process optimization [1].

Another example is in polymer manufacturing, variations in the production process can significantly affect product quality. Implementing SPC is essential to address this issue. SPC involves continuously monitoring and analysing data from the production process to ensure it stays within quality parameters. This real-time monitoring enables immediate corrective action, improves product consistency, reduces costs, and increases customer satisfaction. SPC also helps in regulatory compliance and supports data-driven decision-making. Quality engineers and managers play a critical role in defining quality parameters, analysing data, and driving continuous process improvement, thus ensuring high-quality polymer products [2].

SPC offers numerous benefits for manufacturing processes, including reduced wastage and warranty claims, maximized productivity, and increased operational efficiency. It minimizes the need for manual inspections, leading to enhanced customer satisfaction and controlled costs. Additionally, SPC improves analytics and reporting capabilities, providing better insights into the manufacturing process and facilitating continuous improvement [1].

In today's highly competitive industrial world, maintaining quality standards is a must. To maintain, monitor and control the quality of a process used to achieve these standards, the industrial world uses a tool called Statistical Process Control or often shortened to SPC. Among the various techniques in SPC, control charts are a very useful tool for identifying shifts and trends in process data.

Multivariate control charts extend the principles of univariate charts to handle the complexity of multiple variables. These charts are especially important in environments where quality cannot be captured by a single measurement, such as in manufacturing, healthcare, and finance. By considering the relationships between variables, multivariate control charts provide a more comprehensive view of process stability and variability. This thesis explores the application and effectiveness of multivariate control charts, specifically Hotelling's T2, Multivariate Cumulative Sum (MCUSUM), and Multivariate Exponentially Weighted Moving Average (MEWMA) charts. Each of these graphs offers unique advantages and is suitable for different types of shifts and data characteristics.

2 Theoretical Basics

This section studies the basic concepts of statistics and quality control, which are essential for understanding and applying advanced statistical methods in practice. We start with descriptive statistics, which provide a comprehensive summary of data through measures such as mean, median, mode, variance and standard deviation. These statistics are essential for summarizing and understanding the basic features of a data set, serving as the first step before performing more complex analyses.

After the introduction to descriptive statistics, the focus shifts to Statistical Quality Control (SQC). SQC encompasses a variety of techniques used to monitor and control processes to ensure they are operating at their maximum potential. This includes the application of control charts, which are tools used to determine whether a manufacturing or business process is in under control. One of the most important fundamentals is the basics of the multivariate normal distribution, which is essential for understanding the behavior of multivariate data in the quality control process.

The principles of statistical process control (SPC) will be explored, a method used to monitor and control processes through data-driven techniques, ensuring that processes operate efficiently, producing more products that conform to specifications with less waste. Finally there will be a brief explanation of univariate Shewhart, CUSUM, EWMA along with multivariate Hotelling's T^2, MCUSUM and MEWMA which is the core of this thesis.

2.1 Descriptive Statistics

Statistics involves the collection, organization, analysis, interpretation and presentation of data. In data analysis, this translates to statistical analysis of data sets. There are two main categories of statistical techniques: descriptive and inferential statistics. Descriptive statistics focus on summarizing and describing the visible characteristics of a data set, such as mean, variance, and distribution [3].

These statistics organize and present data in a factual manner, often using tables, charts, and graphs. In contrast, inferential statistics use sample data to make generalizations about larger populations, predict future outcomes, and draw conclusions beyond the available data. In inferential statistics, there are 3 commonly used techniques: hypothesis testing, confidence intervals, and regression analysis. In this section, the focus is on descriptive statistics for multivariate control charts [3].

Statistics includes three fundamental tasks in data analysis, each corresponding to a different subfield:

- Description

  Description involves depicting the frequencies of various manifestations of observed characteristics through methods such as graphical data processing, including diagrams and frequency tables, which are particularly useful for presenting large data sets. These methods also help in forming initial impressions or ideas for further analysis. In addition, descriptive statistical methods help in validating data by identifying errors, such as inaccuracies stemming from questionnaire transcriptions. However, it is important to note that descriptive analysis does not involve stochastic processes and does not allow for making conclusions beyond the survey data collected [4].
- Exploration

  Exploration involves uncovering the structure in the data without using stochastic methods. This includes the formulation of hypotheses for the stochastic model underlying the data, which is essential for induction. This process is often computerized, requiring significant computational resources for thorough analysis and hypothesis generation [4].
- Induction

  Induction involves providing a method to draw generalized conclusions about the population. These inferences are based on probabilistic models, often implied by exploratory methods. The process usually starts with asking a question, followed by formulating a stochastic model (not based on data), then followed by point estimates, confidence intervals, and hypothesis testing. This approach makes it possible to draw broader conclusions beyond the specific data set being examined, using statistical models to generalize the findings to a larger population [4].

Descriptive statistics can be used to describe multivariate data, which the simple descriptive are used to summarize each variables individually and it is important to describe the relationship between the variables. There are two possibilities to describe multivariate data i.e. by using Pearson's or Spearman's Correlation coefficients [5].

Karl Pearson, a British mathematician and biostatistician, pioneered the concept of correlation and was instrumental in applying statistical methods to biological issues such as heredity and evolution. Along with Francis Galton, Pearson was a proponent of eugenics [6]. The Pearson's Correlation formula is written below:

$$r = \frac{cov\ (x,y)}{SD\ (x).\ SD\ (y)} = \frac{cov\ (x,y)}{\sqrt{var\ (x)}.\sqrt{var\ (y)}} \quad [7]$$

Pearson's formula for the correlation coefficient (r) quantifies the linear relationship between two variables, x and y. Here, cov(x,y) represents the covariance, measuring how much x and y vary together. A positive covariance signifies that the variables usually increase together, while a negative covariance implies that one variable increase while the other decreases. Standard deviation (SD) measures the dispersion or variation in a set of values, and is the square root of the variance (var(x)), which measures the spread of the data. By normalizing the covariance by the product of the standard deviations of x and y, the Pearson correlation coefficient provides a dimensionless measure, scaled between -1 and 1, that indicates the strength and direction of the linear relationship between two variables.

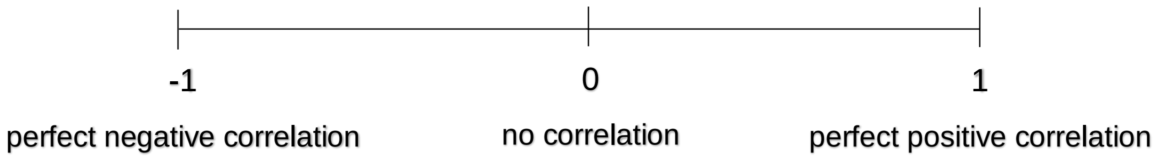

Figure 1: Correlation's value.

Charles Edward Spearman (1863-1945) was a British psychologist best known for his contributions to statistics, particularly in pioneering factor analysis and developing the Spearman rank correlation coefficient. Elected to the Royal Society in 1924, his citation highlights his innovative application of mathematical methods to psychology and his study of correlation. Influenced by Francis Galton, Spearman's statistical advances, including rank correlation and correction for attenuation, were not fully appreciated by his colleague Karl Pearson, leading to a long-standing feud. Although celebrated for his statistical work, Spearman viewed it as secondary to his pursuit of fundamental psychological laws [8]. His formula is written below:

$$\rho = 1 - \frac{6 \sum d_i^2}{n\,(n^2-1)} \qquad [7]$$

Spearman's rank correlation coefficient (ρ) measures the strength and direction of the association between two ranked variables. The formula involves calculating the rank differences (d_i) between corresponding values of the two variables, squaring these differences, and summing them up. This sum of squared rank differences is then scaled by a factor of 6 and divided by the product of the number of observations (n) and (n^2-1), which normalizes the value. The result is the same dimensionless coefficient that ranges from -1 to 1 with the same meaning as Pearson's correlation.

Table 1: Strength of the correlation.

Value of r	Strength of the correlation
0 < 0,1	No correlation
0,1 < 0.3	Low correlation
0,3 < 0,5	Medium correlation
0,5 < 0,7	High correlation
0,7 < 1,0	Very high correlation

The value of correlation indicates the strength of the linear relationship between two variables. An r value between 0 and 0.1 indicates no correlation, which means there is no linear relationship. Values between 0.1 and 0.3 indicate a low correlation, which suggests a weak linear relationship. An r value between 0.3 and 0.5 reflects a moderate correlation, which indicates a moderate linear relationship. A high correlation is indicated by a value between 0.5 and 0.7, which indicates a strong linear relationship. Finally, an r value between 0.7 and 1.0 indicates a very high correlation, which indicates a very strong linear relationship between the two variables.

In this section, the focus will shift to the main multivariate descriptive statistics: the mean vector, covariance matrix, correlation matrix, and how to derive the covariance matrix from the correlation matrix. These metrics are essential for understanding the relationships and variability between variables in complex data sets, thus aiding informed decision-making during multivariate analysis.

Mean vector:

$$\bar{x} = \frac{1}{n} X^T X \qquad [9]$$

The mean vector, denoted as $\bar{x}$, is a key concept in multivariate analysis, especially in multivariate control charts. This vector represents the average value across multiple variables. The formula for calculating it involves multiplying the transposition of the data matrix X^T by the original data matrix X, summing the values across observations, and dividing by the total number of observations n. This average serves as a reference point to detect shifts or deviations in the process being monitored.

Covariance matrix

$$S = \frac{1}{n-1} X^{*T} X^{*} \qquad [9]$$

The formula calculates the sample covariance matrix, denoted as S, where X* represents the centered data matrix. This matrix is obtained by subtracting the mean vector of each observation in the original data matrix X. The sample covariance matrix provides insight into the relationship and variability between variables in the data set. By normalizing by n - 1, where n is the number of observations, this matrix adjusts the degrees of freedom and estimates the covariance structure based on the sample data. This statistic is very important in multivariate analysis, allowing assessment of how variables vary with each other and providing important information for various statistical analyses and modeling techniques.

Correlation matrix

$$R = D^{-\frac{1}{2}} S D^{-\frac{1}{2}} \qquad [9]$$

This formula calculates the correlation matrix R using the sample covariance matrix S and the diagonal matrix D. Here, D contains the square root of the variance of each variable along its diagonal, which serves to standardize the data. The correlation matrix R provides a measure of the linear relationship between pairs of variables while adjusting for their respective variances. Dividing the covariance between variables by the product of their standard deviations yields a correlation coefficient ranging from -1 to 1. This formula is essential in multivariate analysis to understand the interdependence between variables and identify patterns in complex data sets.

Building covariance matrix from correlation matrix

$$S = D^{-\frac{1}{2}} R D^{-\frac{1}{2}} \qquad [9]$$

This formula calculates the covariance matrix S by using the correlation matrix R and the diagonal matrix D. Here, D contains the variance of each variable along its diagonal, and R represents the correlation coefficient between pairs of variables. By taking the square root of the variance and plugging it into the formula, the resulting covariance matrix S is scaled to account for the variability of each variable. This process standardizes the data and converts the correlation matrix R into a covariance matrix, which provides insight into the relationship and variability between variables. Used in multivariate analysis, this formula plays an important role in understanding the covariance structure of complex data sets and in various statistical modeling techniques.

2.2 Statistical Quality Control

Statistical quality control (SQC) is the use of statistical methods aimed at maintaining and improving product quality. The use of analysis and statistics is fundamental to the area of quality control and is an essential task in the world of production due to the uncontrollable factors and variations in the production process. Even with the best plant management and handling processes, there will be factors that affect and reduce the product's optimality, such as reduced quality of the raw products used, machine conditions that deteriorate over time or decreased operator performance. That's why SPC is so vital in the industrial world [10]. The control chart technology developed by Walter Shewhart at Bell Labs in 1924 was used in US companies in the 1930s and was further refined and distributed worldwide after 1950 [11].

Another prospect for using statistical methods to improve product quality can be seen at the product design stage and process. It is no secret that two-thirds of all malfunctions are usually located in the design (Bagchi, 2000). Basically, to achieve maximum production results, good basic materials, optimal settings in product design and also settings for the production process are needed. To be able to find out where the optimal settings are so that the production results achieve the expected results (maximum results), experiments can be carried out by adjusting the input parameters and paying attention to the resulting output and analysing it diagrammatically [10].

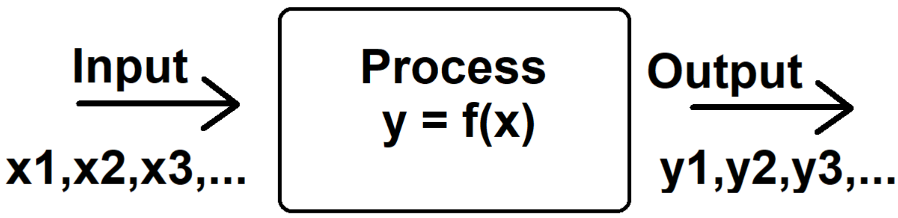

Figure 2: Idealisation process.

In practice, control sometimes needs to be carried out even live (while the process is in progress) with the aim of maintaining product quality. The main methodologies of SQC are divided into 3, namely:

1. Acceptance sampling:
   Another name for this method is sampling inspection. This is because at this stage, if 100% inspection cannot be carried out, then some samples of the product will be taken and inspected and draw conclusions using the inspection results. This technique is usually applied to a population, where the sample will be used and used as a representative of the population can be general and can also be specific,

depending on the characteristics or nature of the sample used. To determine whether the sample taken is feasible and acceptable, there must be rules used to determine its validity [10].

2. Statistical process quality:

   In practice, even the most stable production output is still subject to random variation and therefore, the SPC method is the answer. SPC aims to control the variability of process output by using a control chart where certain characteristics of the product are plotted. Under normal circumstances, the movement of the points on the plot will move according to the expected pattern ("normal") and should there be any unusual movement, it can be concluded that there has been a change in the process parameters and an investigation should be carried out immediately to find the unusual condition. This unusual condition may be due to equipment failure, incorrect use of raw materials, poor operator performance and so on, causing the point to move out of its normal pattern. Once the reason for the abnormality has been discovered, corrective action is required to get things back to normal. It can be concluded that SPC has the goal of maintaining a stable, capable and predictable process [10].

3. Design of experiments:

   Design of Experiments, commonly abbreviated as DOE, is a systematic and scientific statistical methodology used to conduct an experiment to find the optimal solution for parameter settings so that the best quality production results are achieved. Another advantage of applying this method is that it minimises the total effort required in product or process development experiments while achieving an increased level of accuracy in the results [10].

2.3 Multivariate Normal Distribution

<u>Univariate Normal Distribution</u>

Before delving into the multivariate normal distribution, it's important to understand the univariate normal distribution. A random variable X is said to be normally distributed with a mean μ and variance σ^2 if it follows the probability density function:

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{(-\frac{(x-\mu)^2}{2\sigma^2})} \qquad [12]$$

This function results in the familiar bell-shaped curve prevalent in statistics. The PDF shows the squared difference between the variable x and its mean μ. This value is minimized when x equals μ, meaning the density function reaches its maximum at x = μ. The variance σ^2

determines the spread of the distribution around this maximum. A larger σ^2 implies a wider spread, whereas a smaller σ^2 results in a narrower spread. In short form notation, the normal distribution of X is often denoted as:

$$X \sim N\,(\mu, \sigma^2) \quad [12]$$

This indicates that X follows a normal distribution (denoted by N), with mean μ and variance σ^2. This concise notation is widely used in statistical analysis to represent normally distributed variables.

<u>Multivariate Normal Distribution</u>

The multivariate normal distribution is a fundamental concept in statistics and probability theory. It extends the idea of a univariate normal distribution to multiple dimensions, where instead of dealing with a single random variable, multiple random variables that follow a joint normal distribution are dealt with. At the core of this distribution are two important parameters: the mean vector and the covariance matrix.

The mean vector (μ) summarizes the average value of the variable under consideration. It consists of a vector where each element corresponds to the average of the variable, so that the central tendency across dimensions can be known. Complementing the mean vector is the covariance matrix (Σ), a symmetric matrix that reveals the relationship between variables. Its diagonal elements represent the variance, which shows the spread of each variable, while the off-diagonal elements convey the covariance, which describes the extent to which the variables vary together.

A p×1 random vector X is said to follow a multivariate normal distribution if it has a population mean vector μ and a population variance-covariance matrix Σ. The joint density function for this random vector is given by:

$$f_X(x) = \frac{1}{(2\pi)^{\frac{p}{2}}\, \Sigma^{\frac{1}{2}}}\, e^{\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)} \quad [12]$$

Here, |Σ| denotes the determinant of the variance-covariance matrix, and Σ^{-1} is its inverse. The distribution reaches its maximum when the vector x equals the mean vector μ, decreasing around that maximum. When p = 2, it means bivariate normal distribution, which forms a three-dimensional bell-shaped curve. The short notation for the multivariate normal distribution is:

$$X \sim N_p(\mu, \Sigma) \quad [12]$$

This indicates that the vector X is distributed as a multivariate normal with mean vector μ and variance-covariance matrix Σ. Key aspects of the multivariate normal distribution include the quadratic form inside the exponent, known as the squared Mahalanobis distance between the random vector x and the mean vector μ. If the variables are uncorrelated, the variance-covariance matrix is diagonal, with the variances of the individual variables on the main diagonal and zeros elsewhere. This simplifies the multivariate normal density function to a product of univariate normal densities.

Linear combinations of multivariate normal random variables can also be considered. Suppose X is multivariate normal with mean μ and variance-covariance matrix Σ. Then, for a linear combination $Y = c^{T}X$, where c is a vector of coefficients, Y is normally distributed with mean $c^{T}\mu$ and variance $c^{T}\Sigma c$. These quantities can be estimated using sample estimates of the population parameters.

In addition, the multivariate normal distribution exhibits symmetry around the mean vector, displaying a bell-shaped curve that extends along each dimension. This symmetry underscores the attractiveness and usefulness of this distribution in modeling diverse situations in real-world practice.

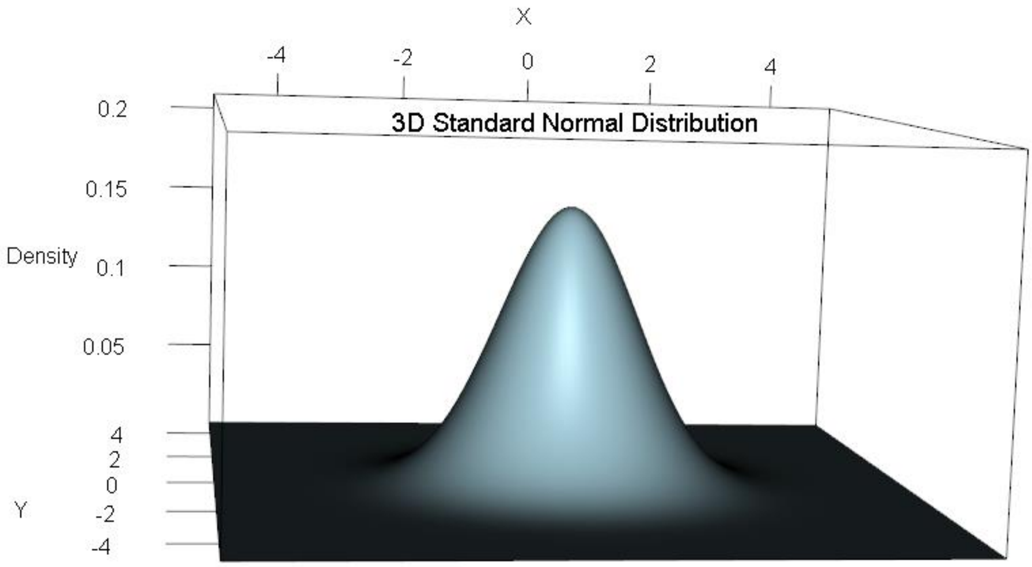

Figure 3: Multivariate normal distribution: Bell-shaped curve.

The figure 2 illustrates a three-dimensional representation of the standard bivariate normal distribution, a specific case of the multivariate normal distribution. This bell-shaped surface visually represents the joint probability density of two normally distributed variables, X and Y, both with a mean of 0 and standard deviation of 1. The peak of the bell curve corresponds to the point where both X and Y are equal to their means, indicating the highest density and thus the highest probability of occurrence.

In this plot, the height (or density) of the surface at any point represents the probability of observing a particular combination of X and Y. The density function is highest at the center of the distribution (where both variables are at their mean values) and decreases symmetrically as one moves away from the center on both axes. This decrease follows the properties of the multivariate normal distribution, where the probability density decreases exponentially as the distance from the mean vector increases.

The bell-shaped curve shows the main characteristics of the multivariate normal distribution: it is unimodal and symmetrical around its mean vector. The smooth and continuous surface indicates that any small change in X or Y results in a predictable and gradual change in the density, reflecting the inherent correlation between the variables. The spread and shape of the curve is affected by the variance-covariance matrix, which, in this standard case, is the identity matrix, indicating that the variables are uncorrelated with equal variances.

In the field of statistical quality control and process monitoring, the application of multivariate control charts relies on a fundamental understanding of the multivariate normal distribution. This distribution serves as the basis for modeling the joint behavior of multiple variables, which is critical for detecting deviations from the expected pattern. Setting control limits and accurately identifying out-of-control conditions relies heavily on an understanding of the properties of this distribution. By integrating the principles of the multivariate normal distribution into multivariate control charts, practitioners can improve their ability to maintain process stability and quickly address anomalies. This underscores the important role of statistical theory in real-world applications, to ensure a robust quality management system.

2.4 Statistical Process Control

Statistical Process Control (SPC) is a very important tool for ensuring that manufactured products meet their designed requirements. This tool helps distinguish between common cause variation, which is inherent to the production process, and special cause variation, which is caused by factors such as defective raw materials or improper operations. The main objective of SPC is to detect special cause variation and signal when it occurs [13].

SPC is usually divided into two phases: Phase I involves setting up and stabilizing the production process, while Phase II focuses on monitoring the stabilized process online. In Phase I, statistical analysis is performed exploratively, and the relationships between quality characteristics and controllable input variables are studied. The process is iteratively adjusted until it runs stably. In Phase II, the stabilized process is monitored using SPC charts, and any significant deviations trigger signals to investigate and eliminate special causes [13].

There are seven main tools that are the most important tools for SPC, namely:

1. Flowchart

   Flowcharts serve as a visual representation of the sequence of activities in a project or process, highlighting their dependencies. Standardized symbols are used to convey detailed process knowledge: ovals represent the beginning or end of the process, boxes indicate action items, and diamonds signify decision points. This makes it easy to identify steps that affect quality and potential control points. In addition, these diagrams can compare the ideal process with the actual process, pointing out areas that need improvement. Flowcharts often kick-start Business Process Reengineering (BPR) efforts [10].

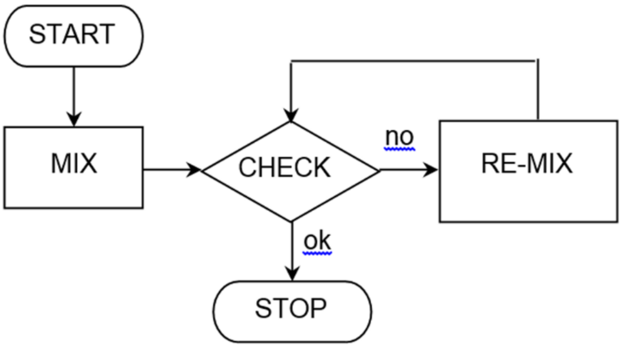

Figure 4: Example of flowchart[10].

2. Histogram

   A histogram is a bar chart that displays the distribution of the number or characteristics of a variable. For example, a lineup of students by height shows the tallest, shortest, and most numerous students clustered around the average height. In manufacturing, histograms quickly identify the nature and extent of quality problems by examining the shape and width of the distribution, thus informally establishing process capability. In addition, histograms can be used to compare multiple distributions [10].

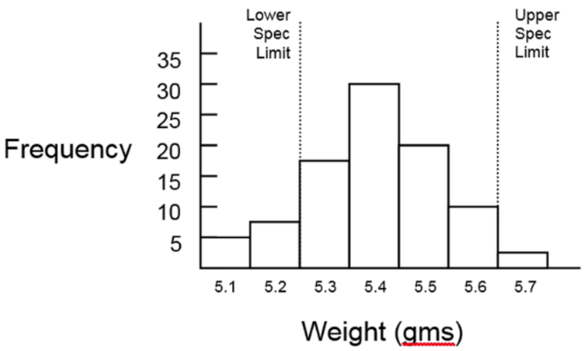

Figure 5: Example of Histogram [10].

3. Pareto Chart

The Pareto chart displays the distribution of the effects of various causes, organized from most frequent to least frequent, highlighting the most common problems. Named after economist Wilfredo Pareto, who noted that wealth is not evenly distributed, this chart visually represents the relative frequency of quality problems. The chart helps identify some of the critical issues that cause most problems, following the principle that 80% of problems are usually caused by 20% of factors. This makes it a valuable tool for prioritizing improvement efforts [10].

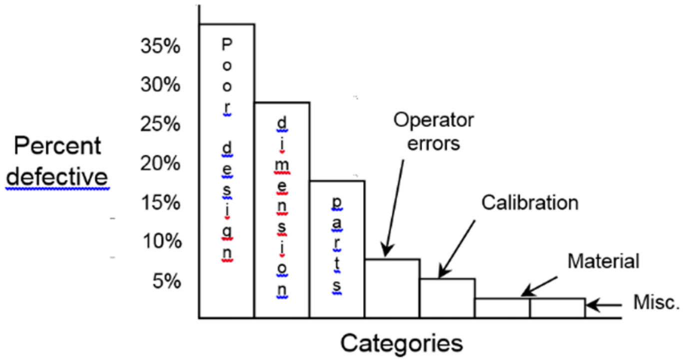

Figure 6: Example of Pareto Chart [10].

4. Cause and Effect (Ishikawa Diagram)

A cause and effect diagram, also known as a fishbone diagram or Ishikawa diagram, is mainly used to list the causes of a particular quality problem or defect. The main causes branch out from a central horizontal line, with sub-causes originating from those branches. This diagram helps identify areas for data collection and analysis, develop reaction plans for out-of-control points on control charts, and serves as the first step in planning design of experiments (DOE) studies and applying Taguchi methods to improve product and process design.

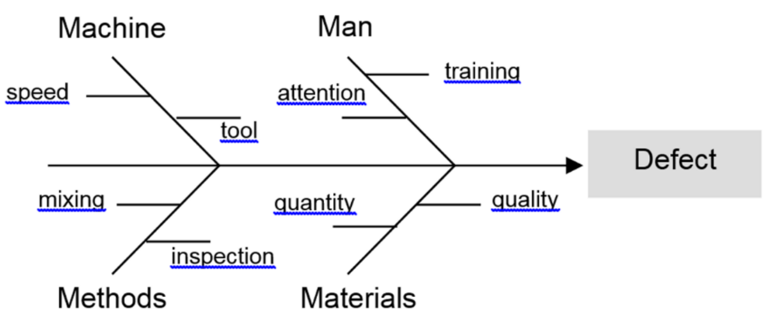

Figure 7: Example of Ishikawa Diagram [10].

5. Scatter Diagram

A scatter diagram displays the relationship between two variables to identify existing patterns. For example, it can show whether there is a relationship between outdoor temperature and the incidence of the common cold. The closer the scatter points are to the diagonal line, the stronger the one-to-one relationship between the variables. This diagram helps develop an informal model to predict future outcomes based on past correlations. However, it is important to note that correlation does not imply causation, and it is not a tool for controlling processes [10].

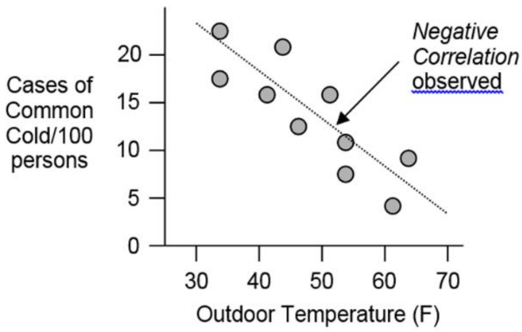

Figure 8: Example of Scatter Diagram [10].

6. Run Chart

A running chart displays the history and patterns of variation of data points over time, connected by a line. This tool is primarily used to identify trends over time, with analysts indicating whether the upward or downward trend is favorable. This tool is used at the beginning of a change process to identify existing problems and at the end to verify whether the implemented changes have resulted in permanent process improvements [10].

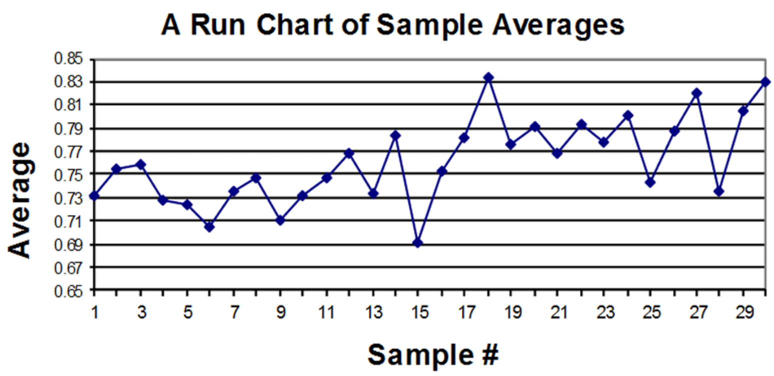

Figure 9: Example of Run Chart [10].

7. Control Chart

   A control chart is a graphical display of a sample over a specified duration of time that shows the measured or calculated quality of that sample. The control chart has one center line that has the function of indicating the average value of the quality characteristic that is desired or corresponds to the condition under control. In addition, there are also two other horizontal lines that function as controls, namely UCL and LCL, UCL is the Upper Control Limit and LCL is the Lower Control Limit. These control limits are determined according to the user's wishes and are useful for showing the user that if there is a point that is outside the control point, it means that there is an error in the process or it can also be interpreted that the process is out of control, so it is necessary to investigate and take corrective action to find and eliminate the cause and return the process to control again [14].

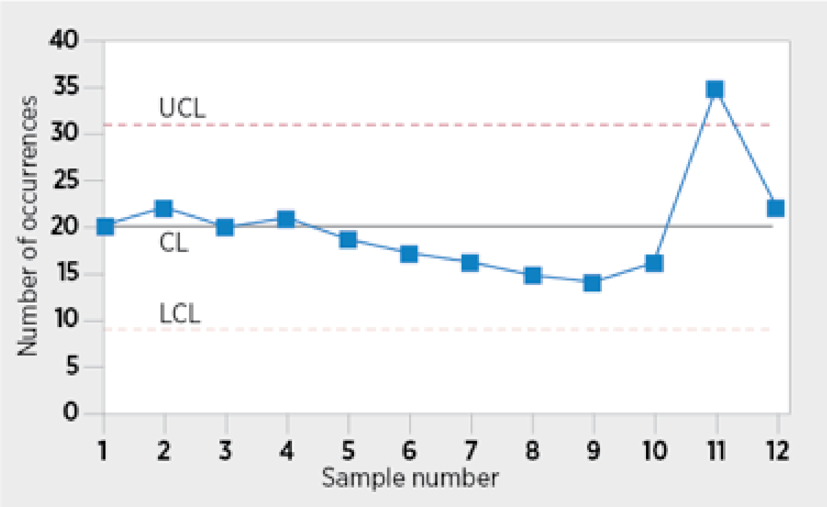

Figure 10: Example of Control Chart [15].

Important advances in SPC include the development of control charts such as cumulative sum (CUSUM) and exponentially weighted moving average (EWMA) charts, which use all available data to detect variations caused by special causes. Recent advances in SPC include methodologies for change point detection, nonparametric control charts, and monitoring profiles [13].

Overall, SPC plays an important role in maintaining product quality and improving process efficiency by identifying and addressing sources of variation in the manufacturing process.. According to Montgomery, SPC is one of the most powerful technological advances of the 20th century, because it is based on sound basic principles, is easy to use, has a significant impact, and can be applied to any process [14].

2.5 Control Charts

In 1931 Walter Andrew Shewhart, an American physicist, engineer and statistician, promoted a systematic approach to quality management in the industrial world using control charts, which also explains the foundation of the idea of control charts. Therefore he is known as the "father of statistical quality control" and now his charts are known as Shewharts charts or Shewhart control charts (introducing a visual representation of variation by plotting the sample data on the control limits) [16].

The Shewhart chart became a cornerstone in the evolution of quality management, due to its practical way of detecting variation, while introducing the concept of assignable causes and root cause analysis, thus enabling advances in statistical process recognition. The term "Shewhart chart" is often used interchangeably with "univariate control chart". Basically, all Shewhart charts are part of univariate control charts, but not all univariate control charts are part of Shewhart control charts [13].

Statistical process control (SPC) is a comprehensive approach to managing production processes, encompassing two distinct phases. Phase I is dedicated to stabilizing the production process, involving the collection and analysis of data on the quality characteristics of a set of products. Iterative adjustments are made until stability is achieved, with a focus on addressing root causes. Once stability is confirmed, an in-control (IC) dataset is obtained under stable conditions, and the IC distribution of quality characteristics is estimated. Phase II introduces a control chart designed based on the IC distribution for online monitoring. In the event of a significant shift, signaling an out-of-control (OC) performance, production is immediately halted for root cause identification [13].

Both phases employ various statistical tools such as histograms, stem-and-leaf plots, regression, and design of experiments. Notably, control charts, especially Shewhart charts, play a crucial role in detecting OC performance. Originating in 1931, Shewhart charts have evolved over 80 years, with different versions proposed for diverse purposes. The chapter delves into representative Shewhart charts for univariate quality characteristics, while subsequent chapters cover multivariate quality characteristics and nonparametric distributions. The second part of the chapter briefly touches on process capability analysis. Overall, SPC aims to ensure the stable and controlled operation of production processes through systematic analysis and adjustment, leveraging a variety of statistical tools, with a particular emphasis on the efficiency of control charts in detecting any deviations from expected performance [13].

At control charting, the statistics determined by the observed data should be chosen, and it should contain as much information as possible about the distribution of the quality characteristics and be sensitive to any distribution shifts. There are several types of control charts that have been developed, namely Shewhart control charts, Cumulative Sum (CUSUM) control charts, exponentially weighted moving average (EWMA) control charts and control charts based on change-point detection (CPD) [13]. In the next subchapters, some examples of control charts will be described more specifically along with its types, namely Shewhart Control Charts, CUSUM Control Charts and EWMA Control Charts (CPD will not be included).

2.5.1 Univariate Shewhart Control Chart

In the application of univariate Shewhart chart and process capability analysis in statistical process control (SPC) to the production process, there are two main phases of SPC: phase I, in which the production process is adjusted to ensure stability, and phase II, in which online monitoring is performed using control charts designed based on the in-control (IC) distribution of quality characteristics. During phase I, adjustments are made iteratively until the process stabilizes, and IC data sets are collected to design phase II control charts. These control charts, including Shewhart charts, CUSUM charts, EWMA charts, and others, are specifically designed to detect out-of-control (OC) performance [17].

This chapter focuses on Shewhart charts for univariate quality characteristics, with a discussion of their variations and applications over the years. In addition, this chapter also provides insight into process capability analysis, which forms the basis for further exploration in subsequent chapters. Overall, this chapter provides a comprehensive overview of the SPC methodology, emphasizing the importance of control charts in maintaining process stability and quality control in manufacturing processes. In general, Shewhart charts can be said to be the traditional types of charts, such as X-Bar charts, R charts, P charts, C charts, U charts, and Individual charts. These charts are based on statistical process control principles and are designed to monitor the central tendency and variability of a process [17].

Univariate Shewhart control charts consist of X-bar ($\bar{X};\ mean$) and Range (R) charts, X-bar ($\bar{X}$) and Standard Deviation (S) charts, and Proportion (P) and Count (C) charts. X-bar charts monitor the process average, while R or S charts assess the variability within each sample. Alternatively, X-bar charts and S charts provide another option for variability tracking. For attribute data, there are P charts, which monitor the proportion of non-conforming items in each sample, and C charts, which track the number of defects per constant unit. X bar charts are essential for detecting shifts in process averages, while R, S, P, and C charts help identify changes in variability and attribute conformance. The selection of a particular graph depends on the nature of the data and the purpose of process monitoring [17].

Other standard traditional Shewhart control charts commonly used include the Defective Items (N) in Sample (P) chart and the Unit (U) chart. NP charts are an additional type of attribute control chart, similar to P charts but applied when the subgroup size remains constant. In contrast, U charts are used to track the number of defects or nonconformities per unit of measurement in varying subgroup sizes, so they can be used for scenarios where subgroup sizes may fluctuate.

2.5.2 CUSUM Control Chart

The CUSUM control chart, introduced by Page (1954), quickly detects small to moderate process shifts by using information from a long sequence of samples, though it struggles with handling outliers and deviations from normality. To address these issues, the CUSUM chart has been proposed by Yang et al. (2010) and further evaluated for its robustness against non-normality and outliers in various studies, including those by Nazir et al. (2013a). [18].

The fundamental principle of CUSUM is to continuously compare the values of a variable against a predetermined target, known as the reference value. By plotting or tabulating the cumulative sum of deviations from this target, the method effectively tracks the process performance over time. When the cumulative sum exceeds a pre-set decision interval, it indicates a significant shift in the process mean, signaling potential issues in quality [19].

The goal of the CUSUM method is to detect mid-term changes in a process by using historical data. This involves simultaneously plotting two cumulative sum curves, Sj+ and Sj-, which track upward and downward deviations, respectively. The reference values are calculated as

$$k = d * (\frac{\sigma}{\sqrt{n}}), \text{ and } H = B * (\frac{\sigma}{\sqrt{n}}), \text{ where } d = \frac{m}{2} \qquad [20]$$

is optimal for detecting a shift of $m\sigma$. Standard values for these parameters are

$$d = \frac{1}{2}; B = 4 \; or \; 5 \qquad [20]$$

with k being the lower reference value and H being the upper reference value.

The following algorithm is used for CUSUM control chart:

Default values: μ_0, σ, d, B [20]

Algorithm based on [20]:

1. $S_0^+ := 0,\ S_0^- := 0,\ k = d * \left(\frac{\sigma}{\sqrt{n}}\right), H = B * (\frac{\sigma}{\sqrt{n}})$
2. Calculation of the block mean value $\bar{X}_j$ (or use individual values)
3. $S_j^+ := S_{j-1}^+ + (\bar{X}_j - \mu_0 - k)$

   If S_j^+ < 0, the chart is ended or not started. (i.e. S_j^+ = 0)
4. $S_j^- := S_{j-1}^- + (\bar{X}_j - \mu_0 + k)$

   If S_j^- > 0, the chart is ended or not started. (i.e. S_j^- = 0)
5. If $S_j^+ \leq H$ and $S_j^- \geq H$ → process for subgroup j is under control, otherwise out of control (j ← j+1)

2.5.3 EWMA Control Chart

The EWMA chart was originally called the geometric moving average chart introduced by Roberts (Technometrics 1959). Due to the fact that exponential smoothing serves as its basis, the name of this chart was changed to the exponentially weighted moving average (EWMA) chart [21]. It is also an alternative to Shewhart as well as x chart. The EWMA chart incorporates information from all previously collected data so it provides a faster response to shifts in the process mean compared to individual charts or x-charts. EWMA charts are particularly useful for use in monitoring phase 2 processes against predefined control limits, as they have a shorter average length in detecting small process drifts [22].

The following procedure is important for decision making in EWMA control chart:

Standard values: $\mu_0, \sigma, \lambda, w, k$ with $w = 1{,}96;\ , k = 3$ [23]

Decisions limits based on [23]:

1. Warning limits:
   - $Lower\ Warning\ Limit\ LWL = W_L = \mu_0 - w\sigma\sqrt{\frac{\lambda}{2-\lambda} * (1-(1-\lambda)^{2t})}$
   - $Upper\ Warning\ Limit\ LWL = W_U = \mu_0 + w\sigma\sqrt{\frac{\lambda}{2-\lambda} * (1-(1-\lambda)^{2t})}$
2. Control limits:
   - $Lower\ Control\ Limit\ LCL = C_L = \mu_0 - k\sigma\sqrt{\frac{\lambda}{2-\lambda} * (1-(1-\lambda)^{2t})}$
   - $Upper\ Control\ Limit\ LCL = C_U = \mu_0 + k\sigma\sqrt{\frac{\lambda}{2-\lambda} * (1-(1-\lambda)^{2t})}$

Decision Rule: If $W_L \leq \tilde{X}_t < C_L$ or $W_U < \tilde{X}_t \leq C_U$ then a warning is issued for the value t. And if $C_L \leq \tilde{X}_t \leq C_U$ than the process is under control for the value t, otherwise it is out of control [23].

2.6 Multivariate Control Charts

Multivariate control charts are used to monitor multiple variables simultaneously to detect any patterns, trends, or anomalies that may suggest a process is not operating within acceptable limits. These control charts are particularly useful in situations where multiple factors can affect the quality or performance of a product or process. In this section, 3 multivariate control charts will be introduced as well as their key parameters and most important formulas, namely Hotelling's T^2 control chart, MCUSUM control chart and MEWMA control chart.

2.6.1 Multivariate Hotelling's T^2 Control Chart

In 1931, Harold Hotelling's extended the univariate t test with one dependent variable to a multivariate t test with two or more dependent variables, hence the name Hotelling's T^2 [24]. In the multivariate $\bar{X}$ control chart, a d-dimensional vector of quality characteristics is observed instead of a one-dimensional quality characteristic and each subgroup contains the same number n of measured values. For the subgroup j, the data vectors $\vec{X}_{ji1}, \ldots, \vec{X}_{jin}$ with $\vec{X}_{ji} = (X_{ji1}, \ldots, X_{jid})^T$ are present. Assuming that the vector $\vec{X}_{ji}$ has a multivariate normal distribution with expected value vector $\vec{\mu} \epsilon \mathbb{R}^d$ and covariance matrix $\sum \in \mathbb{R}^{d,d}$, the control chart procedure to be discussed here corresponds to hypothesis testing:

$$H_0\colon \vec{\mu} = \vec{\mu}_0, H_1\colon \vec{\mu} \neq \vec{\mu}_0 \qquad [25]$$

The vector $\vec{\mu}_0$ contains in its components the nominal values of the quality characteristics, which if the process runs in the normal path, then the hypothesis H_0 is satisfied. The estimator for vector $\vec{\mu}_0$ using data from subgroup j is:

$$\bar{X}_j = \frac{1}{n}\sum_{i=1}^{n} \vec{X}_{ji} \qquad [25]$$

The multivariate mean map is used to plot the time evolution of the Hotelling's Statistic T. The formula is:

$$T_j = n\,(\bar{X}_j - \vec{\mu}_0)^T\, \Sigma^{-1}\, (\bar{X}_j - \vec{\mu}_0) \qquad [25]$$

The following procedure is to be used for decision making in multivariate X-Chart:

Default values: $\vec{\mu}_0, \Sigma, \alpha_W, \alpha_K$

Decision boundaries: $W = X^2_{d;1-\alpha_W},\ K = X^2_{d;1-\alpha_K}$ [25]

Boundaries rules based on [25]: If Datas T is between 0 and K, the process is under control, if it is higher than K, the process is out of control. If Datas is between W and K, a warning is issued.

$$0 \leq T_j \leq K = under\ control$$

$$T_j > K = out\ of\ control$$

$$K < T_j < W = warning$$

Standard values of α_W and α_K based on [25]

$$\alpha_W = 0.05$$

$$\alpha_K = 0.0027$$

A very important problem in the context of multivariate control charts is the interpretation of the "out of control" condition [25].

2.6.2 Multivariate CUSUM Control Chart

No single person is credited for the invention of the MCUSUM chart, as it came about through the work of various researchers who adapted and extended Page's original CUSUM methodology to handle multivariate data. In the context of a Multivariate Cumulative Sum (MCUSUM) control chart, there are some key components and terminology that are crucial to understanding the function. The reference values, k and h, play significant roles. The k value, also known as slack or reference value, is a pre-defined threshold that helps distinguish between in-control and out-of-control process conditions. Basically, k determines the sensitivity of the graph; a smaller value makes the graph more sensitive to small changes. The value of h, or the decision interval, acts as the control limit. If the monitoring statistics exceed this value, it indicates a potential shift in the process, thus triggering an alarm.

Reference values → k, h [26]

The monitoring statistic is calculated using the observed vector X_t at time t, the target mean vector μ_0 (in-control mean), and the cumulative sum vector S_{t-1} from the previous time period t−1. The current cumulative sum vector S_t is updated based on the current observation X_t and the previous sum S_{t-1}. The monitoring statistic C_T is defined as:

$$C_T = \left\| \vec{S}_{t-1} + \vec{X}_t - \vec{\mu}_0 \right\| \qquad [26]$$

→ ‖·‖ denotes the Euclidean norm, representing the length of the vector.

Updating the cumulative sum vector depends on the value of Ct. If Ct ≤ k, the process is considered under control, and the cumulative sum vector St is reset to zero. However, if C_t > k, the cumulative sum vector is updated to $\left(\vec{S}_{t-1} + \vec{X}_t - \vec{\mu}_0\right)\left(1 - \frac{k}{C_t}\right)$. This adjustment ensures that only significant deviations from the target average accumulate over time.

$$\vec{S}_t = \begin{cases} 0 \, for \, C_t \leq k \\ \left(\vec{S}_{t-1} + \vec{X}_t - \vec{\mu}_0\right)\left(1 - \frac{k}{C_t}\right) for \, C_T > k \end{cases} \qquad [26]$$

Finally, the process is considered out of control if C_T – k > h. This condition indicates that the monitoring statistic C_T exceeds the reference value k by more than the decision interval h, indicating a significant shift in the process mean. This comprehensive approach enables the MCUSUM control chart to effectively monitor and detect shifts in multivariate processes, ensuring timely intervention and maintaining process stability.

Out of control if: $C_T - k > h$ [26]

2.6.3 Multivariate EWMA Control Chart

The concept of multivariate EWMA charts generalized from univariate to multivariate by Cynthia A. Lowry, William H. Woodall, Charles W. Champ and Steven E. Rigdon in 1992 [27]. The MEWMA formula was provided by the first supervisor in a PDF format during the research phase. The three-page PDF containing this formula will be included in the appendix as a source.

The Multivariate Exponentially Weighted Moving Average (MEWMA) control chart is a powerful tool for monitoring the stability of processes involving multiple quality characteristics that are correlated over time. In the MEWMA chart, the smoothed statistic $\tilde{X}_t$ is calculated using this formula:

$$\tilde{X}_t = \lambda X_t + (1 - \lambda)\tilde{X}_{t-1} \qquad [26]$$

X_t is the current observation and $\tilde{X}_{t-1}$ is the previous smoothed value. Here, λ is the smoothing parameter, determining the weight given to the most recent observation compared to the past observations. Alternatively, the smoothed statistic can be expressed as:

$$\tilde{X}_t = \Lambda X_t + \left(I - \Lambda\right)\tilde{X}_{t-1} \qquad [26]$$

Λ is a diagonal matrix that allows different smoothing parameters for each quality characteristic. An out-of-control signal is generated when the statistic T^2 exceeds a pre-defined reference value h. The T^2 statistic is calculated as:

$$\text{Out of control if: } T^2 = \left(\tilde{X}_t - \mu_0\right)^T \Sigma^{-1} \left(\tilde{X}_t - \mu_0\right) > h \qquad [26]$$

$\tilde{X}_t$ is the smoothed vector, μ_0 is the target mean vector (in-control mean), and Σ is the covariance matrix of the in-control process. This formula measures the squared deviation of the smoothed vector from the target mean, weighted by the inverse of the covariance matrix. If T^2 exceeds the control limit h, it indicates a significant deviation from the in-control condition, indicating that the process may be out of control. This method effectively detects small, gradual shifts in the process mean, thus making the MEWMA chart a powerful tool for quality control in multivariate processes.

3 Implementation: Software and tools

In this thesis, Statistica is used to visualize each univariate control chart, and R is used to perform multivariate control chart analysis. Statistica, with its easy-to-use interface, facilitates exploratory data analysis and visualization, which is crucial for control chart implementation. Its intuitive functionality accelerates data manipulation and enhances the ability to interpret complex patterns. R, a powerful programming environment, complements this approach with extensive statistical functions and advanced modeling tools that provide flexibility in advanced multivariate control charting. The scripting nature of R ensures careful customization and reproducibility, which increases the reliability of the analysis.

Other notable software options for multivariate control charts include MATLAB, SAS, and Python with libraries such as NumPy and SciPy. MATLAB excels in numerical calculations, SAS offers a complete package for analysis and Python is versatile with powerful libraries.

3.1 Statistica

StatSoft and its software, Statistica, share a long and dynamic history, beginning with the release of the Complete Statistica System (CSS) in 1986. This was followed by the first DOS version under the name STATISTICA in 1991, alongside early Mac releases which were later discontinued. After various versions and intervals, the first 32-bit version of Statistica (version 6) was launched in 2001, leading to regular updates with new features. In 2014, StatSoft Incorporated, the developer based in Tulsa, Oklahoma, was acquired by Dell, and StatSoft (Europe) GmbH in Hamburg became independent. Statistica then moved to Quest through Dell and later joined TIBCO's portfolio in 2017, aligning closely with TIBCO's Spotfire. By 2024, Statistica became part of the newly formed Spotfire Business Unit, branded as Spotfire Statistica, enhancing its role in complex data analyses and integration with Spotfire's dashboarding capabilities [28].

TIBCO Statistica® seamlessly combines various functionalities including data blending, data discovery, predictive analytics, forecasting, text mining, and an embedded business rules engine. Its primary goal is to provide actionable insights and recommendations to frontline workers and decision-makers. By translating predictions into tangible business actions, Statistica plays a crucial role in guiding decision-making across the organization. It facilitates the integration of these recommendations directly into operational systems, thereby streamlining processes and enhancing efficiency. Additionally, Statistica promotes collective intelligence by seamlessly integrating with app marketplaces such as Algorithmia, Apervita, and AzureML. Furthermore, it empowers data scientists by offering support for standard scripting languages like Python and R, enabling them to leverage their expertise and contribute to the analytical process [29].

Statistica is not the main focus of this thesis, but it is used to demonstrate its capabilities in univariate control chart analysis. This chapter will highlight two specific features of Statistica: the descriptive statistics function and the ability to create Hotelling T² control charts. However, the main emphasis of this thesis is on using the R programming language to implement and visualize the three multivariate control charts. Therefore, although Statistica will be used in the application chapter, its role will be limited to univariate Shewhart, CUSUM, and EWMA control charts. Although Statistica is also capable of performing multivariate control charts, as will be shown in this section.

3.1.1 Elementary Statistics

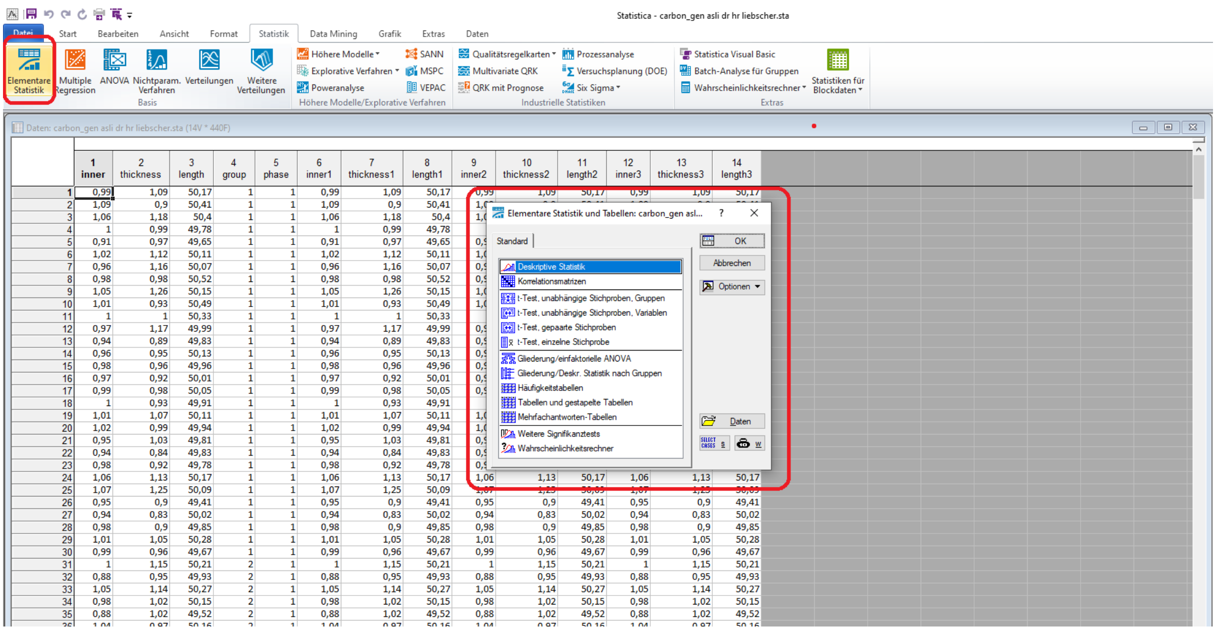

Figure 11: Statistica: Elementary statistics interface.

The illustration shows the "Statistics" surface, which contains important statistical tools such as descriptive statistics, correlation matrix, ANOVA (Analysis of Variance) as well as others that are listed in figure 10. Below is a brief explanation of each of these elements:

- Descriptive Statistics: involve methods for summarizing and describing the features of a data set. This includes measures such as mean, median, mode, standard deviation, and range, which help in understanding the central tendency, dispersion, and shape of the data distribution.
- Correlation Matrix: is used to examine the relationship between pairs of variables in a data set. It provides a matrix of correlation coefficients, which indicates the strength and direction of the linear relationship between variables.
- T-test, Independent Samples, Group: The t-test for independent samples is a statistical test used to compare the means of two independent groups to determine if they are significantly different from each other. This test assesses whether the mean differences

observed in the samples are likely due to chance or represent true differences in the population.

- T-test, Independent Samples, Variable: Similar to the previous test, the t-test for independent samples with variables makes it possible to compare the means of two groups, but considers several variables simultaneously.
- T-test, Paired Samples: The t-test for paired samples compares the means of two related groups to determine if there is a significant difference between them. It is used when the same subjects are measured under different conditions or at different times.
- T-test, Single Sample: The t-test for a single sample is used to determine whether a single sample mean is significantly different from a known or hypothesized population mean.
- One-factorial ANOVA: is a statistical technique used to compare the means of three or more independent groups to determine if there are statistically significant differences among them. This technique tests the null hypothesis that the means of all groups are equal.
- Descriptive Statistics by Group: This involves organizing and analyzing descriptive statistics separately for different groups in a data set, which allows for comparison and insight into group differences.
- Frequency Tables: summarize the distribution of categorical variables by calculating the frequency of each category or class.
- Tables and Stacked Tables: Tables present data in a structured format, while stacked tables make it possible to organize and display data in a stacked or nested manner, which is often useful for hierarchical data.
- Multiple Response Tables: summarize and analyze data from surveys or questionnaires where respondents can choose multiple response options for a single question.
- Additional Significance Tests: This refers to various statistical tests beyond those mentioned above, which are used to assess the significance of relationships or differences in data.
- Probability Calculator: is a tool used to calculate probabilities for different statistical distributions, allowing the calculation of probabilities associated with specific events or outcomes.

In this section, examples of descriptive statistical analysis will be provided through practical demonstrations using sample data. Short but clear explanations regarding the example will also be given to explain this descriptive statistical analysis.

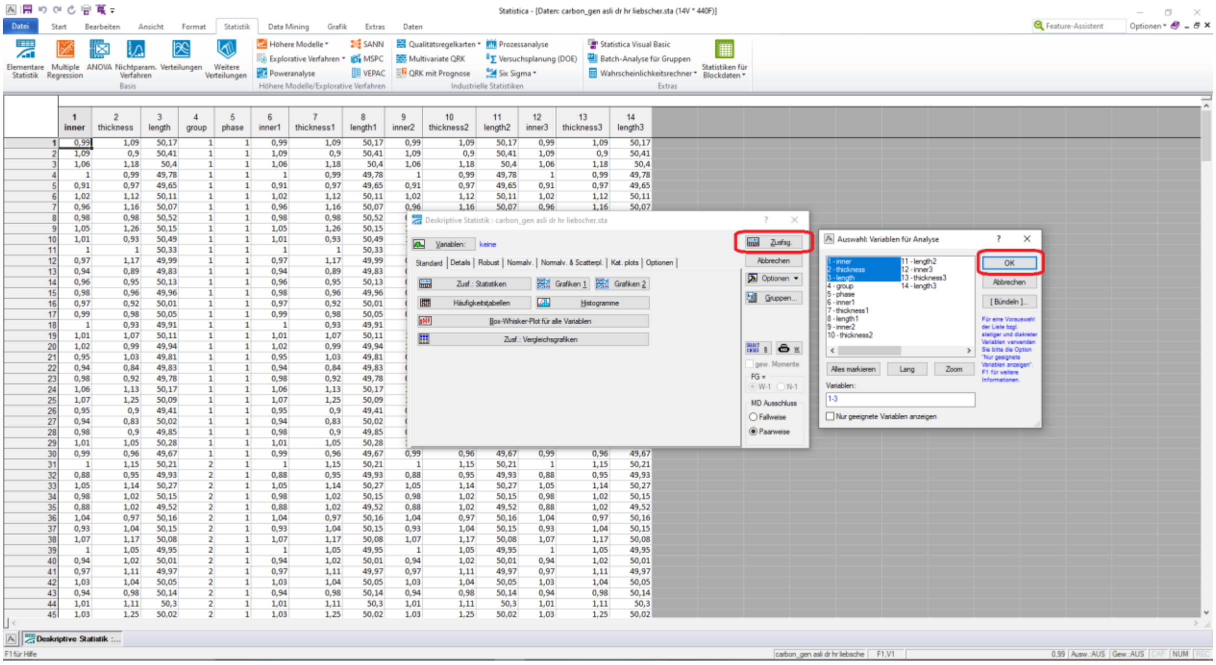

Figure 12: Statistica: Descriptive statistics interface.

Figure 11 provides an overview of the interface for selecting descriptive statistics. This interface features various customizable columns so that users can customize the results as desired. In addition, users can directly select variables to determine the descriptive data needed, simplifying the process and improving user-friendliness. For example, in figure x, the selected variables are inside, thickness and length. After that, by clicking ok, and clicking summary (zusfsg), the result will be given. Figure y is the desired descriptive data analysis result.

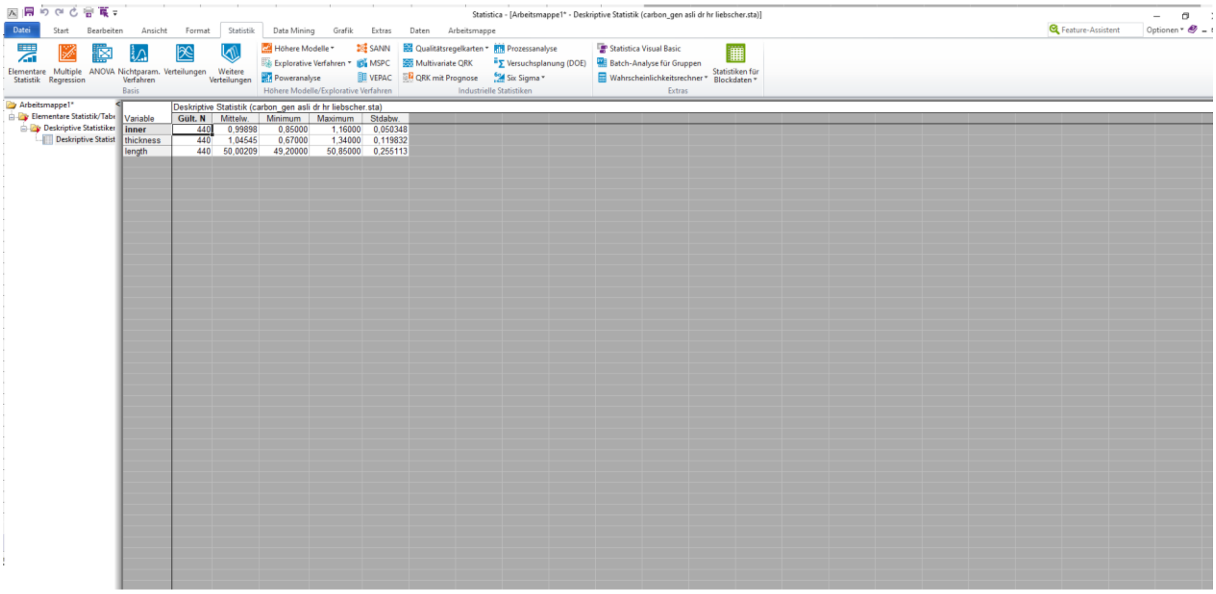

Figure 13: Statistica: Descriptive statistics results.

Figure 12 shows the statistical data processing results obtained which include various measures that summarize and describe the main characteristics of the data set. These measures include Valid N (number of valid cases), Mean (average value of the variable), Minimum (smallest observed value), Maximum (largest observed value), and Standard Deviation (a measure of the spread or variability of the data around the mean). The use of descriptive statistics helps provide important information about the central tendency, spread and also the distribution of the data being analyzed, as well as assisting users in understanding the underlying properties and informing further analysis.

3.1.2 Multivariate Control Chart in Statistica

Although Statistica is mainly used to visualize univariate control charts in this thesis, this section will demonstrate its ability to illustrate even multivariate control charts. Specifically, this section will demonstrate the usefulness of Statistica in creating a Hotelling'S T^2 control chart using modified carbon data. In this example, columns 1-5 contain the original data, while columns 6-14 consist of the self-generated data. Theoretically, this section is illustrating a Hotelling's T^2 control chart carbon excel data in Statistica.

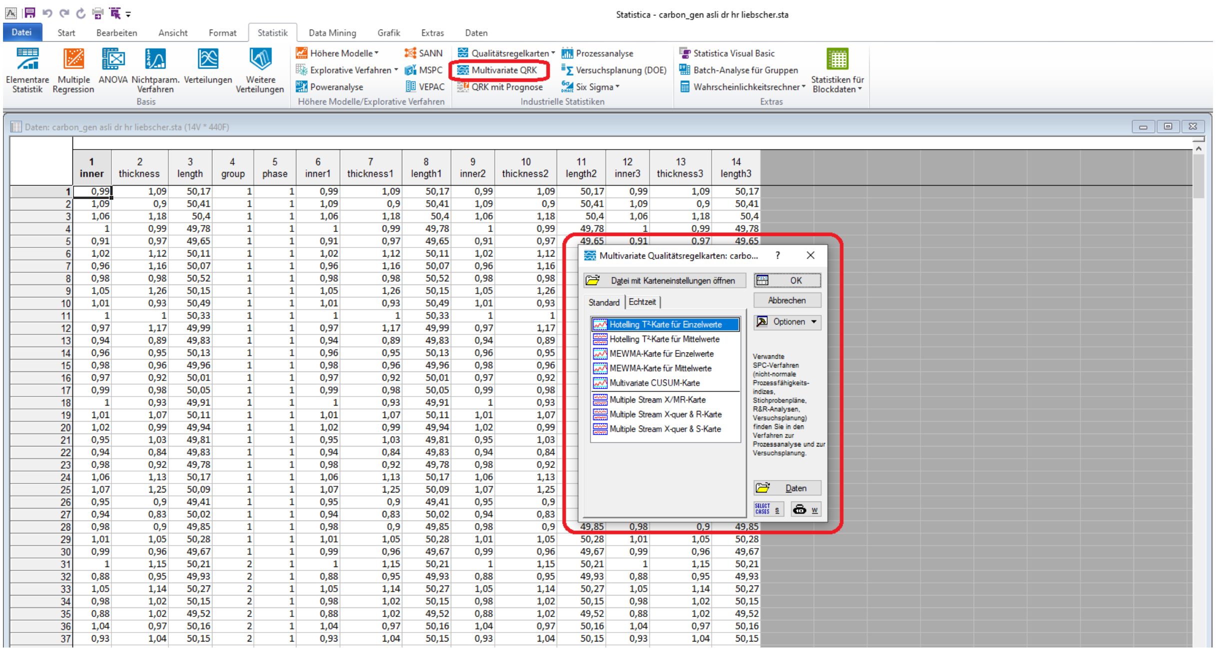

Figure 14: Statistica: Multivariate quality control charts interface.

The illustration shows the "Multivariate QRK" interface, which has functions for multivariate control charts, including options for Hotelling for single and average values, as well as MCUSUM and MEWMA. In "Multivariate QRK," QRK stands for "Qualitätsregelkarten". Hotelling's T^2 control charts are used to monitor multivariate data for process variation. "Hotelling für Einzelwert" is concerned with monitoring individual observations, while "Hotelling für Mittelwert" involves monitoring means of multivariate data sets. The former detects shifts in individual observations, while the latter identifies shifts in the mean vector of multivariate data.

MCUSUM (Multivariate Cumulative Sum) and MEWMA (Multivariate Exponentially Weighted Moving Average) are additional multivariate control chart techniques used to detect small shifts in the mean vector of multivariate data over time. These tools collectively enable practitioners to effectively monitor and maintain process quality and reliability by identifying deviations from expected behavior, facilitating timely corrective action. Here is an example of illustrating data on a multivariate control chart using Statistica.

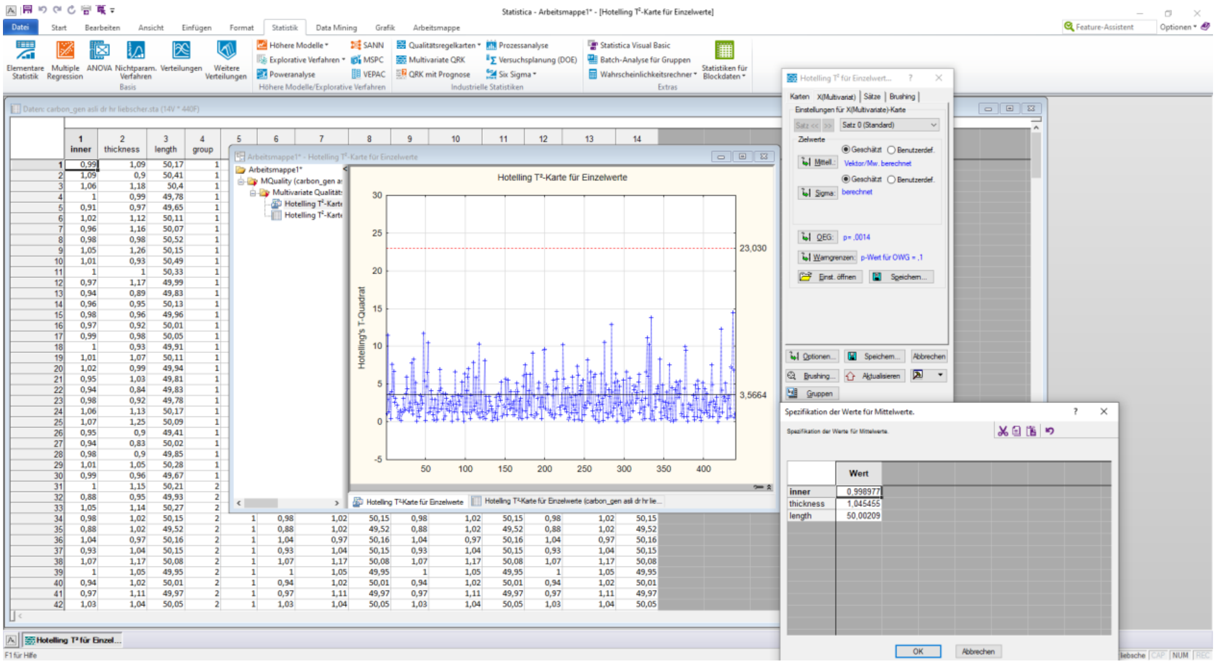

Figure 15: Statistica: Hotelling's T^2 control chart.

The visual representation of the Hotelling's T^2 control chart for carbon data, generated using the Statistica application, can be seen in the graph. It shows that the entire process is within the control limits. By clicking "Multivariate QRK" and then "aktuelle fortsetzen," which means continuing the current process, users can access the advanced statistical settings. For example, users can check the mean of each variable, here in this example are inner, thickness, and length, and modify the values as needed. Additionally, Statistica offers options to view descriptive statistics of the data under analysis. This flexibility allows users to perform a thorough examination and customization of their statistical analyses.

3.2 R Programming Language

R is a powerful language and environment designed for statistical computing and graphics. Developed as part of the GNU project, it draws inspiration from the S language created at Bell Laboratories. Although similar to S, R offers unique features and compatibility with much of the S code. The language offers a wide variety of statistical and graphical techniques, making it highly versatile and extensible. Most notable is its ability to produce publication-quality plots with ease. R is distributed under the GNU General Public License, which allows for widespread adoption and collaboration. R is compatible with a wide range of platforms, including UNIX, Linux, Windows, and MacOS, making it accessible to a broad user base [30].

R is a comprehensive set of software tools designed for data manipulation, calculations, and graphical display. R offers a powerful set of features, including efficient data handling and storage capabilities, specialized array operators for calculations and especially on matrix calculations, a cohesive set of tools for data analysis, and graphical facilities for data visualization on screen or hardcopy. In addition, R also offers a simple but powerful programming language equipped with conditionals, loops, functions, and I/O operations, which empowers users to perform complex data tasks with ease and flexibility [30].

What distinguishes R from software consisting of different tools, is the term "environment" which also reflects the cohesive design that R has. The language used in R is built around an actual computer language, thus allowing users to extend its functionality by defining new functions. Much of R is written in its own S dialect, making it easy for users to understand. For computationally intensive tasks, C, C++, and Fortran code can be integrated simultaneously at runtime. For advanced users, they can manipulate R objects directly using C code [30].

Many users think of R as a statistical system, although it is actually more accurately thought of as an environment that supports statistical techniques. R can be easily extended through the use of existing packages and many more are available through the CRAN website. It also provides comprehensive documentation, in a LaTeX-like format, which is available online and in hard copy [30].

3.2.1 Brief Introduction: S Programming Language

The S programming language, developed primarily by John Chambers along with contributions from Rick Becker and Allan Wilks at Bell Laboratories in 1976, is a versatile language that includes both imperative and object-oriented paradigms. Originally designed to offer an interactive approach to statistical computing, S evolved into a very important tool in data analysis and visualization. Its early versions focused on providing an alternative to calling Fortran subroutines directly, emphasizing interactive graphical tools and accessible documentation. Over time, S was ported to different platforms, with significant expansions and refinements culminating in the publication of the "New S Language" in 1988. This updated version introduced substantial changes such as the transition from macros to functions and expanded the concept of object-oriented. Despite initial challenges in transitioning to the new syntax, subsequent publications refined S further, introducing advanced features such as formula notation and enhanced object-oriented capabilities in versions such as S4. Today, the legacy of S lives on through its modern implementations, R and S-PLUS, which remain one of the most popular programming languages for statistical computing and data analysis [31].

3.2.2 Advantages and Disadvantages of R Programming Language

Like every programming language, R has its own strengths and weaknesses when used in practice. These advantages and disadvantages are presented in the form of a list. Here are the advantages and disadvantages of the R programming language based on [32]:

Table 2: R advantages and disadvantages.

No.	R Advantages	R Disadvantages
1.	Comprehensive Package Ecosystem	Steep Learning Curve
2.	Statistical Proficiency	Performance Limitations
3.	Data Visualization Excellence	Memory Management Issues
4.	Vibrant Community Support	Package Fragmentation
5.	Reproducible Research	Debugging Challenges
6.	Flexibility and Extensibility	Limited OOP Support
7.	Academic and Research Adoption	Data Security Concerns
8.	Efficient Data Handling	Lack of Commercial Support
9.	Graphical User Interfaces (GUIs)	Variability in Learning Resources
10.	Cost-Effectiveness	Limited GUI Options

R offers a comprehensive ecosystem of packages that are highly beneficial for data analysis and statistical tasks. R has extensive package repositories from CRAN, Bioconductor, and GitHub, covering a broad spectrum of needs. These packages provide ready-made functions and tools, significantly reducing the time needed to implement complex algorithms. R is also renowned for its statistical proficiency, providing a strong foundation in statistics. R excels in a variety of analyses, including linear modeling, time series analysis, and hypothesis testing. Built-in functions and extensive libraries cater to users with varying levels of statistical expertise, making it a versatile tool for beginners and advanced users alike.

Data visualization is another area where R shines, especially through powerful libraries such as ggplot2. This library allows the creation of elegant and flexible visualizations. Users can produce publication-quality plots, charts, and graphs, effectively communicating the insights derived from their data. The vibrant community support around R is a significant asset. An active and collaborative community contributes to the ongoing development of R, sharing knowledge, driving innovation, and creating new packages and functions. This collective effort ensures that R remains at the forefront of statistical computing.

R also excels at facilitating reproducible research. Tools such as RMarkdown and knitr allow users to integrate code, data, and analysis results into a single document. This capability ensures that research findings can be easily reproduced and shared, increasing transparency and collaboration in research. Flexibility and extensibility are core strengths of R. Its flexibility allows users to create custom functions and customize their analysis. In addition, R's interoperability with other languages, such as C, Python, and Java, enhances its extensibility, allowing users to utilize a variety of tools and frameworks.

The adoption of R in academia and research is widespread due to its statistical capabilities and open-source nature. Researchers use R to perform advanced analysis and contribute to the global knowledge base, making it the preferred language in this field. Efficient data handling is another key feature of R. The language offers seamless data import, cleaning, and manipulation capabilities, which are essential for efficiently processing real-world data sets. These capabilities make R an indispensable tool for scientists and data analysts.

Apart from its command-line interface, R provides several graphical user interfaces (GUIs), such as RStudio and R Commander. These GUIs improve usability and accessibility, especially for beginners, making it easier for new users to get started with R. Lastly, R is cost-effective as an open-source language. The language is freely available, making it an attractive option for individuals and organizations seeking powerful data analysis capabilities without significant financial investment. This cost-effectiveness, combined with its powerful features, makes R a popular choice across various sectors.

Despite the numerous advantages that R offers, here is a list of the disadvantages of the R programming language. R, despite its many advantages, presents a steep learning curve for beginners. Its unconventional syntax and vast ecosystem of packages can be challenging to master, requiring more time and effort compared to other programming languages.

Performance limitations are another issue with R. Its interpreted nature can cause performance issues when working with large data sets or complex calculations, making it less suitable for high-performance computing tasks. Memory management in R can be inefficient, often resulting in misallocation or slow performance, especially when working with large data sets. This inefficiency can be a significant drawback for users working with large amounts of data.

The large number of packages available in R can lead to fragmentation. This can result in overlapping functionality or poorly maintained packages, causing compatibility issues and making it difficult for users to find the most appropriate and reliable tools. Debugging in R can be challenging, especially for inexperienced users. The error messages provided by R may lack clarity, making it difficult to identify and resolve problems.

R's support for Object Oriented Programming (OOP) is limited compared to other languages. This limitation can hinder certain programming and software design tasks, potentially affecting the development of more complex applications. Data security issues arise due to the open-source nature of R. Handling sensitive data in a commercial environment requires careful data privacy measures to mitigate potential security risks.

The lack of official commercial support for R can pose challenges for organizations that rely heavily on the language. Without access to specialized professional help, these organizations may struggle to solve complex problems. Variability in learning resources for R is another issue. The quality and accuracy of tutorials and documentation can vary widely, leading to confusion and misinformation for learners trying to build their skills.

Finally, although R offers GUIs such as RStudio, the options may be considered less comprehensive compared to those provided by other data analysis tools. This limitation may impact the overall user experience, especially for those who prefer graphical interfaces over command-line interactions.

Choosing R provides an attractive proposition for data analysts and statisticians. With its customized suite of statistical analysis tools, R offers an extensive toolkit specifically designed to meet the needs of professionals in this field. In addition, R distinguishes itself with outstanding data visualization capabilities, empowering users to create highly customizable plots and visualizations to effectively communicate insights gained from complex data sets. The availability of an extensive repository of user-contributed packages in CRAN further enhances R's versatility, allowing users to tackle a wide array of analysis tasks easily and efficiently.

In addition to its technical capabilities, R benefits from a vibrant open-source community that encourages collaboration and innovation. This supportive environment not only encourages knowledge sharing, but also provides ample resources for problem solving and skill development, which contributes to R's widespread adoption and continued growth. In addition, R's seamless integration with other languages and tools enhances its adaptability to a wide range of applications, while its powerful data manipulation capabilities simplify critical tasks such as reshaping, combining, and transforming data. Combined with its ability to facilitate the creation of reproducible research documents, seamlessly integrating code, output, and narrative, R is emerging as a powerful ally for data analysis and statistical computing, which are indispensable.

3.2.3 Example of Using R to Analyze a Dataset

To illustrate the power of R, a simple example of analyzing the mean and median of a data set will be demonstrated. Using ggplot2, an interactive histogram that visually presents the distribution of the data will be created. This fusion of statistical analysis and dynamic visualization highlights the versatility of R for efficient data exploration.

```
# Load necessary libraries
library(ggplot2)

# Create a list of numbers
numbers <- c(23, 45, 67, 89, 12, 34, 56, 78, 90, 11)

# Calculate the mean and median
mean_value <- mean(numbers)
median_value <- median(numbers)

# Create a histogram of the numbers
histogram <- ggplot(data.frame(x = numbers), aes(x)) +
  geom_histogram(fill = "darkorchid", color = "black", bins = 10) +
  ggtitle("Distribution of Numbers") +
  xlab("Numbers") +
  ylab("Frequency") +
  theme_minimal()

# Print the histogram
print(histogram)

# Print the mean and median
cat("The mean of the numbers is:", mean_value, "\n")
cat("The median of the numbers is:", median_value, "\n")
```

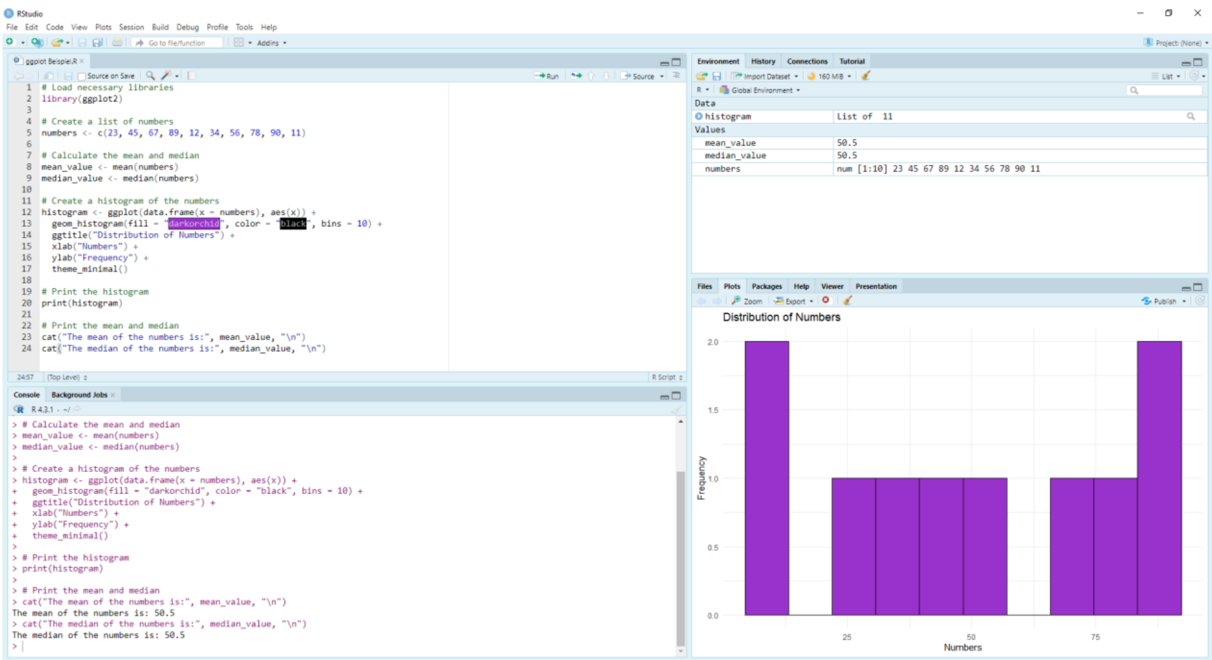

Figure 16: Rstudio: Printing histogram using ggplot2 library.

This code snippet demonstrates basic data analysis in R. It loads the ggplot2 library for data visualization, creates a list of numbers, computes their mean and median, generates a histogram to visualize their distribution, and also finally prints the histogram along with the mean and median values. The window with "Environment, History, Connections, Tutorial" is commonly known as the RStudio IDE (Integrated Development Environment) window. It serves as a central hub for R programming, providing various tools and features to streamline the coding process.

- Environment: Displays information about variables, functions, and other objects currently defined in the R workspace. This allows users to inspect and manage their work environment.
- History: Records commands executed during the current R session, providing a record of previous interactions. Users can review, edit, and re-execute previous commands from the history panel.
- Connections: Provides options for connecting to external data sources and databases, enabling seamless integration of data into R for analysis.
- Tutorials: Offers access to useful resources and tutorials for learning R programming, providing guidance and assistance for users at all skill levels.

The window with "Console, Terminal, and Background Jobs" is also part of the RStudio IDE, specifically the "Console" panel. It serves as the main interface for interacting with R, allowing users to execute R code, view output, and receive real-time feedback.

- Console: This is where R commands are entered and executed. Users can interact with R directly, typing in commands and receiving immediate output and results.
- Terminal: Provides access to the system terminal or command line interface within RStudio. This allows users to execute system commands and perform tasks outside of the R environment.
- Background Job: Allows users to run R scripts or commands in the background, separate from the main R session. This feature is useful for running long-running tasks or batch jobs while continuing to work interactively in the console.

The panel with "Files, Plots, Packages, Help, Viewer, and Presentation" in RStudio is a comprehensive toolbox for data analysis and development.

- Files: Offers a file navigation interface to access and manage files and directories on the local file system, facilitating file manipulation directly within RStudio.
- Plots: Displays graphical output generated by R code, allowing users to visualize data through plots, charts, and visualizations. Interactive features allow customization and exploration.
- Packages: Provides tools for managing R packages, including installation, update, and removal. Users can easily manage their package dependencies and ensure they have access to the latest functionality.
- Help: Offers access to extensive R documentation and help files, assisting users in understanding R functions, packages, and concepts. It provides comprehensive documentation to support learning and troubleshooting.

- Viewer: Renders HTML content generated by R code, such as price drop reports or web pages. Users can view and interact with the HTML content directly within RStudio, making it easy to develop dynamic reports and presentations.
- Presentations: Facilitates the creation and delivery of presentations directly within RStudio. Users can easily integrate R code, plots, and markdown content into their presentations, simplifying the process of sharing insights and findings.

The RStudio IDE window enhances the coding experience by offering tools for managing the R environment, tracking command history, connecting to data sources, and accessing educational materials. It serves as a comprehensive workspace for R development and analysis tasks. The "Console, Terminal, and Background Jobs" window in RStudio provides a versatile environment for coding, running commands, and managing tasks in the R programming language, and the "Files, Plots, Packages, Help, Viewer, and Presentations" panel in RStudio serves as a versatile toolkit for data analysis, package management, documentation, content preview, and presentation development.

4 Applications

This section illustrates the practical application of Statistica in visualizing univariate control charts as well as the application of R programming language in visualizing the main focus of this thesis, the three multivariate control charts, namely Hotelling's T^2 control chart, MCUSUM control chart, and MEWMA control chart. Each multivariate control chart will be visualized in three different scenarios, namely the normal case, as well as two special cases, namely jumps and shifts.

In the context of statistical process control, these control charts serve as invaluable tools for monitoring and detecting deviations from established norms in the quality control process. By visualizing these charts across various scenarios, including typical operating conditions and potential anomalies such as jumps and shifts, this section aims to gain a deeper understanding of their performance characteristics and ability to effectively identify and respond to changes in process behavior. By utilizing the visualization capabilities of R, this section seeks to provide not only practical insights but also actionable responses to improve quality management practices in various industrial settings.

4.1 Univariate Shewhart Control Chart

In this chapter, some examples of univariate Shewhart control charts will be visualized using Statistica. The charts that will be presented include X-bar chart (central tendency/ average of a process), S chart (standard deviations), p chart (proportion), and c chart (count). Each chart will be accompanied by a brief explanation.

4.1.1 X-bar Chart (Central tendency/ Average of a Process)

X-bar charts are used to monitor the central tendency or average of a process. In this chart, the sample mean ($\bar{X}$) of a subgroup of the process is plotted against the control limits. The center line of the graph indicates the overall average of the process, while the upper and lower control limits represents the acceptable range of variation. X-bar charts help identify shifts or trends in the process average so that timely action can be taken to maintain process stability and product quality. Below is a basic illustration of an X-bar chart:

Objective: Graphical representation of the time course of the mean values of the subgroups and corresponding control limits.

Standard in X-bar chart:

- Specified: μ_0, σ, ω and k with $\omega = 1{,}96$; $\kappa = 3$
- Warning limits: $LWL = \mu_0 - \frac{\omega\sigma}{\sqrt{n}}$, $UWL = \mu_0 + \frac{\omega\sigma}{\sqrt{n}}$
- Control limits: $LCL = \mu_0 - \frac{\kappa\sigma}{\sqrt{n}}$, $UCL = \mu_0 + \frac{\kappa\sigma}{\sqrt{n}}$

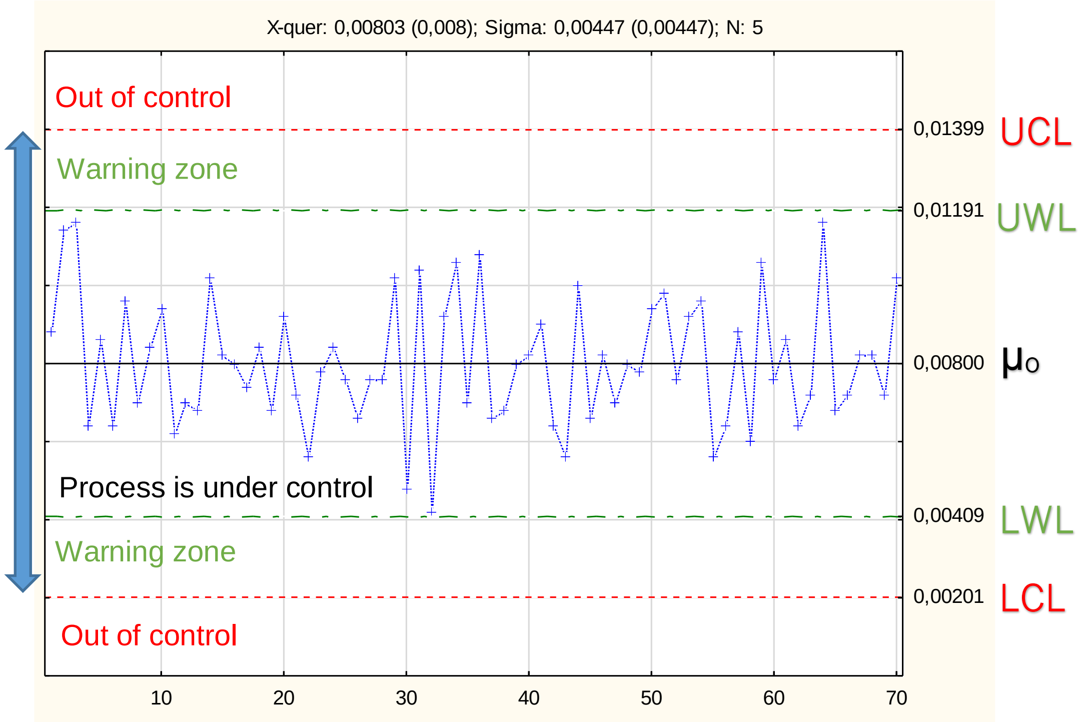

Figure 17: Statistica: X-bar control chart.

- X_{quer} = 0,008 (X-bar)
- σ = 0,00447
- μ_0 = 0,008
- Control limits:
  - ➢ UCL (Upper Control Limit) = 0,01399
  - ➢ LCL (Lower Control Limit) = 0,00201
- Warning limits:
  - ➢ UWL (Upper Warning Limit) = 0,01191
  - ➢ LWL (Lower Warning Limit) = 0,00409
- If:
  - ➢ LCL < points < LWL or UWL > points > UCL → a warning is issued
  - ➢ Points < LCL or Points > UCL → out of control → stop the process → resolve the problem

In statistical process control (SPC), key parameters are used to monitor and maintain process stability. In this example, these parameters include mean value (X-bar), standard deviation (sigma), and target or nominal value (mu-zero). Control limits, such as Upper Control Limit (UCL) and Lower Control Limit (LCL), define the acceptable range of variation within which the process should operate. In addition, warning limits, represented by Upper Warning Limit (UWL) and Lower Warning Limit (LWL), signal potential problems before the process gets out of control. If a data point falls between the LCL and LWL, or between the UWL and UCL, a warning is issued to indicate a potential deviation from the expected performance. However, if the data point goes beyond the LCL or UCL, it signals the process is out of control, prompting immediate action to stop operations and address the underlying issue before resuming production. These conditions enable effective process monitoring and management, ensuring consistency and quality of output. In this example, no points fall outside the under control zone.

4.1.2 S Chart (Standard Deviations)

An S chart, otherwise known as a standard deviation chart, has the function of monitoring the variability or spread of a process. This chart plots the sample standard deviation (S) of a subgroup taken from the process against the control limits. Similar to the X-bar chart, the center line represents the standard deviation of the overall process, while the upper and lower control limits serve as boundary lines where the process is running under control. The use of S charts helps detect changes in process variability, allowing adjustments to maintain consistent product quality. An illustration of an S chart is provided below.

Objective: Graphical representation of the time course of the empirical standard deviations of the subgroups and corresponding control limits.

Standard in S chart:

- Specified: α_w and α_k
- $\alpha_w = 0{,}05$; $\alpha_k = 0{,}0027$
- UWL: p = 0,95 ; UCL: p = 0,9973
- $UWL = \sigma_0 \sqrt{\frac{\chi^2_{n-1}(1-\alpha_W)}{n-1}}$; $UCL = \sigma_0 \sqrt{\frac{\chi^2_{n-1}(1-\alpha_K)}{n-1}}$

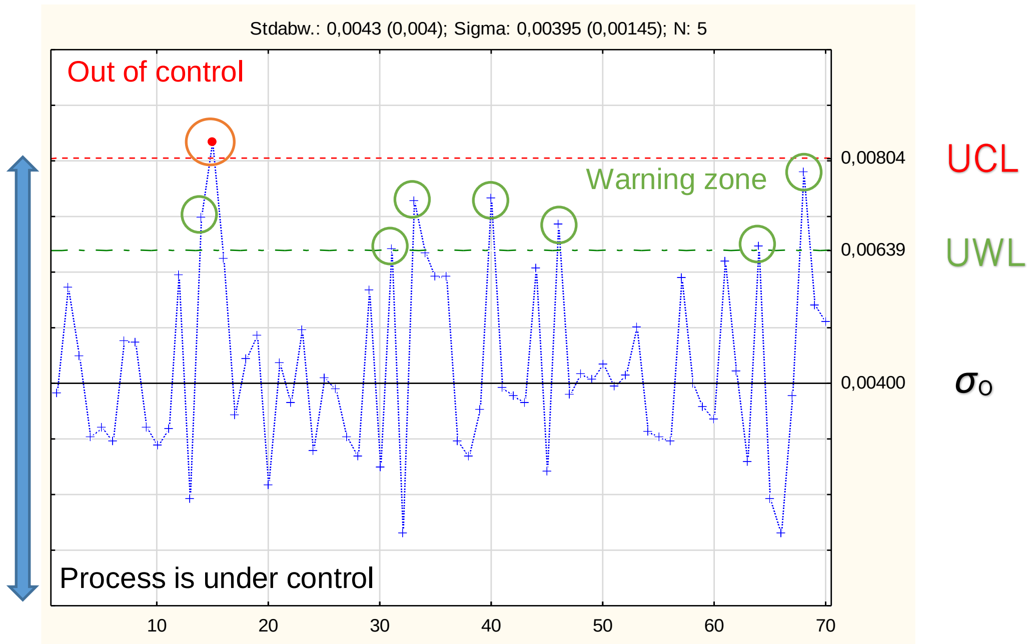

Figure 18: Statistica: S control chart.

- σ (Sigma) = 0,00395
- Standardabw. (Standard Deviation) = 0,0043 (0,004)
- UCL (Upper Control Limit) = 0,00804
- UWL (Upper Warning Limit) = 0,01191
- If:
  - UWL > points > UCL → a warning is issued
  - Points > UCL → out of control → stop the process → resolve the problem
- 7 points are within the warning range → a warning is issued
- 1 point is outside the control limit → out of control → stop the process → resolve the problem
- All other points are within the process control area → under control

In this scenario, several statistical parameters and control limits are defined to effectively monitor and manage the process. The standard deviation (σ) and Upper Control Limit (UCL) provide thresholds for acceptable variation in the process, with an Upper Warning Limit (UWL) indicating a range where caution is required but does not necessarily indicate a serious problem. If a data point falls within the UWL and UCL, a warning will be issued, signaling a potential deviation from expected performance. However, if any data point exceeds the UCL, this signifies the process is out of control, thus requiring immediate action to address the underlying issue before continuing operations.

In this particular case, 7 points were within the warning range, so a warning should be issued. In addition, one point exceeds the control limit, which indicates an out-of-control process that requires intervention to resolve the issue. All other points were within the process control area, indicating that the process was under control.

Influence of the Out-of-Control Points on the Quality Quantities

"Influence of the Out-of-Control Points on the Quality Quantities" addresses the impact of data points that exceed the control limits of a process on the quality of the final product or output. When a process runs within control limits, indicating consistent and predictable performance, it ensures that quality standards are met. However, if data points fall outside these limits, it signals uncertainty and potential defects or errors in the process. Such deviations have significant consequences:

First, out-of-control points contribute to increased variability in the process, leading to inconsistencies in the size, shape, color, or other specifications of the final product. In addition, such points may indicate the presence of defects or faults, ranging from minor defects to critical functional issues, which jeopardize product safety and usability. In addition, such deviations lead to wastage of resources such as raw materials, time, and labor, as defective products often have to be reworked or scrapped.

In addition, the presence of out-of-control points has an impact on customer satisfaction and loyalty, as customers expect consistent quality and deviations can lead to dissatisfaction and loss of business. The quality issues associated with these deviations also mean increased costs to the company, including the cost of rework, scrap, warranty claims and damage to brand reputation.

In addition, there is the potentially fatal impact of out-of-control points, which occur due to uncontrolled quality standards that can even lead to fines, penalties, or legal action, further damaging the company's reputation and impacting profits. In short, the impact of out-of-control points on quality volume is extensive and poses various challenges for companies. Timely identification and corrective actions are critical to maintaining product quality, customer satisfaction, and overall business success.

4.1.3 p Chart (Proportion Chart)

The use of p-charts in SPC is to monitor the proportion or percentage of defective items or events in a process. They are most often applied when the data being measured is in the form of binary results, such as pass/fail, yes/no, or fit/non-fit. P-diagrams are especially useful for processes where the sample size can vary. A simple example will be given below.

Objective: Graphical representation of the proportion of defective products in the subgroups over time.

Standard in p chart:

- Specified: ω, κ and p_0
- ω = 1,6449 (Statistica: -10)
- κ = 2,7822 (Statistica: -10)
- In Statistica: the lower limit (Warning and Control) must also be entered, although these are not required. Therefore -10 is entered.
- p = 0,05 (5 %)
- $W_o = p_0 + \omega \sqrt{\frac{1}{n_j} p_0 (1 - p_0)}$
- $K_o = p_0 + \kappa \sqrt{\frac{1}{n_j} p_0 (1 - p_0)}$

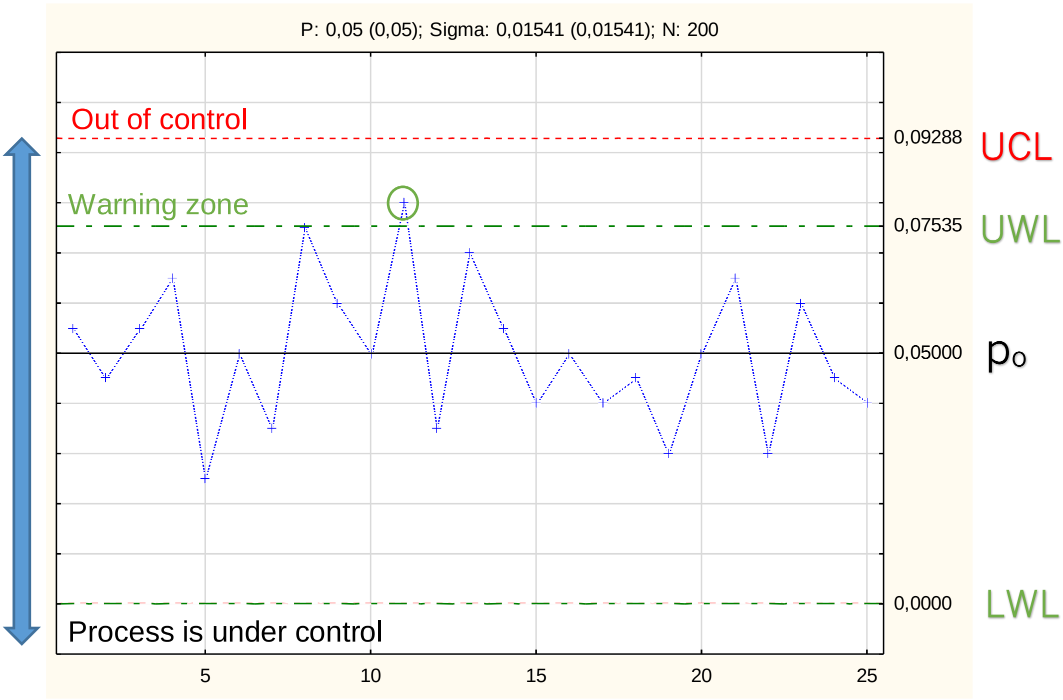

Figure 19: Statistica: p control chart.

- σ = 0,01541
- UCL (Upper Control Limit) = 0,09288
- Warning limits:
  - UWL (Upper Warning Limit) = 0,07535
  - LWL (Lower Warning Limit) = 0
- If:
  - Points < LWL or UWL > points > UCL→ a warning is issued
  - Points > UCL → out of control → stop the process → resolve the problem
- One point (11th point) is in the warning area → a warning is issued
- All other points are under control

In the specific scenario provided, the 11th data point was within the warning area, thus triggering a warning. However, all other points remained within the control limits, indicating that the process was generally stable and operating as expected.

4.1.4 c Chart (Count Chart)

C charts have the function of monitoring the number or frequency of defects per unit produced by a process. In particular, c charts handle discrete data, where defects are counted rather than measured on a continuous scale, unlike other charts that focus on measurements or proportions. C chart example will be provided below.

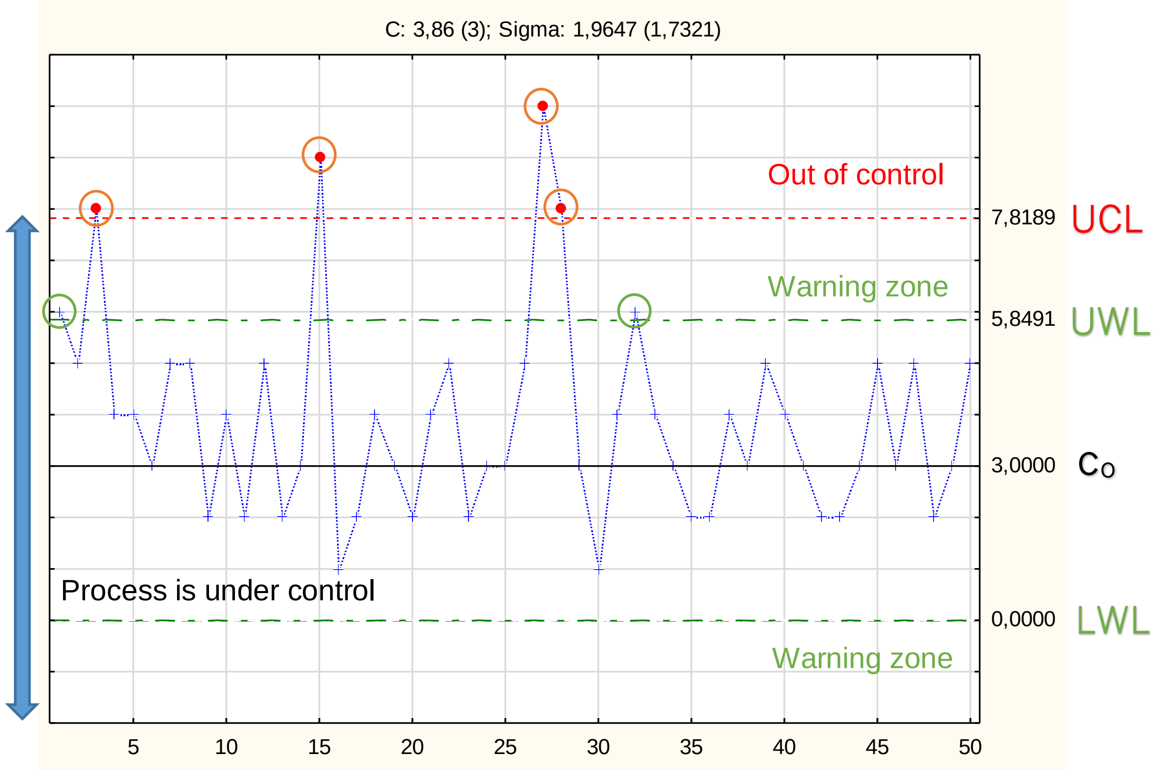

Figure 20: Statistica: c control chart.

- σ = 1,9647
- C_0 = 3,0
- UCL (Upper Control Limit) = 7,8189
- Warning limits:
  - UWL (Upper Warning Limit) = 5,8491
  - LWL (Lower Warning Limit) = 0
- If:
  - Points < LWL or UWL > points > UCL→ a warning is issued
  - Points > UCL → out of control → stop the process → resolve the problem
- 2 points are within the warning range → a warning is issued
- 4 points are outside the control limit → out of control → stop the process → resolve the problem
- All other points are within the process control area → under control

In the example given, two points were within the warning range, thus triggering a warning, while four points exceeded the control limits, thus requiring intervention to resolve the issue. This indicates a potential problem with the process that needs to be addressed to maintain quality control. By identifying these outliers, corrective actions can be taken to prevent further deviations and ensure the process remains stable.

4.2 CUSUM Control Chart

In this section, an example of a CUSUM (Cumulative Sum) control chart will be illustrated, a powerful tool used to monitor changes in a process over time. CUSUM charts are very effective for detecting small shifts in process averages, making them an invaluable asset in quality control. By continuously summing deviations from the target value, this chart helps identify trends that may indicate potential problems. Through the following example, how to interpret a CUSUM chart will be demonstrated, pointing out key features such as warning ranges and control limits, as well as explaining the steps required to address detected issues.

Objective: To determine medium-term changes in the process using data values from further back in time.

Standard in CUSUM chart:

- Specified: μ_0, σ, d und B
- d = m/2 mit m = 1 → d = ½
- n = 1
- B = 4, -4

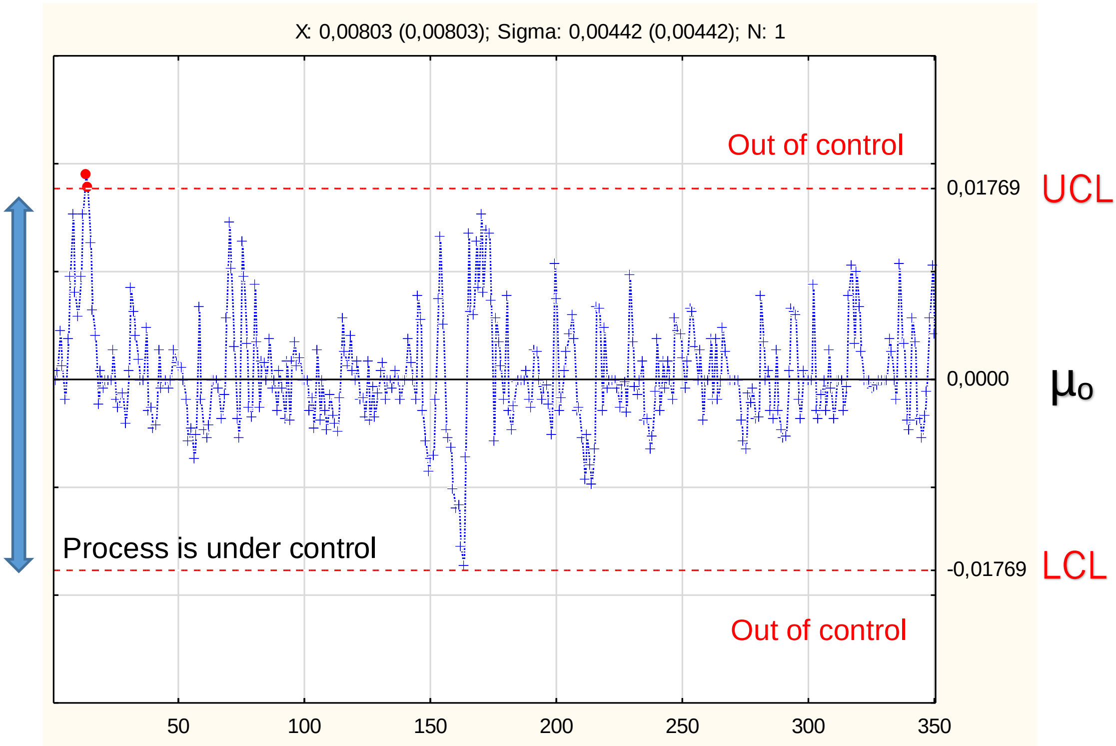

Figure 21: Statistica: CUSUM control chart.

- σ = 0,00442
- X (μ_0) = 0,00803

- Control limits:
  - UCL (Upper Control Limit) = 0,01769
  - LCL (Lower Control Limit) = – 0,01769
- Points < LCL or points > UCL → out of control → stop the process → resolve the problem
- 2 points are outside the control limit → out of control → stop the process → resolve the problem
- All other points are within the process control area → under control

The results of the CUSUM control chart analysis revealed important insights into the stability of the process being monitored. The standard deviation (σ) was calculated as 0.00442, and the target process mean (X or μ0) was 0.00803. The control limits were set at ±0.01769, with the Upper Control Limit (UCL) at 0.01769 and the Lower Control Limit (LCL) at -0.01769. According to the CUSUM chart, any data point that falls outside these control limits indicates that the process is out of control and requires immediate intervention.

In this particular analysis, two points were found to be outside the control limits, indicating that the process had deviated significantly from the target mean. At this condition, the process must be stopped immediately and the cause of the error must be addressed as soon as possible. The identified deviations can be caused by various reasons such as equipment malfunction, raw material inconsistency, human error, or environmental factors affecting the stability of the process.

To address these out-of-control points, several steps can be taken. First, a thorough investigation should be conducted to identify the root cause of the deviation. This may include checking equipment calibration and maintenance, ensuring the quality and consistency of raw materials, retraining staff on proper procedures, and reviewing environmental conditions such as temperature and humidity that may affect the process. Once the root cause is identified and resolved, the process can be restarted and closely monitored to ensure the process remains within control limits. Regular reviews and adjustments may also be required to prevent recurrence of the problem and maintain process stability.

4.3 EWMA Control Chart

In this section, we will present an example of a EWMA (Exponentially Weighted Moving Average) control chart. EWMA charts are very effective for detecting small, gradual shifts in process averages, making them essential for maintaining high quality standards. By applying exponential weighting to historical data, these charts provide a smooth and sensitive representation of process variation. Through clear and concise examples, how to interpret a EWMA chart will be explained, focusing on its key components such as control limits and detection of out-of-control conditions, as well as outlining the corrective actions required when problems are identified.

Objective: To determine medium-term changes in the process using data values from further back in time.

Standard in EWMA chart:

- Vorgegeben: μ_0, σ, λ, ω and κ
- $\lambda = 0{,}2$
- $\omega = 1{,}96$
- $\kappa = 3$
- $LWL = \mu_0 - \omega\sigma\sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})}$, $UWL = \mu_0 + \omega\sigma\sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})}$
- $LCL = \mu_0 - \kappa\sigma\sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})}$, $UCL = \mu_0 + \kappa\sigma\sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})}$

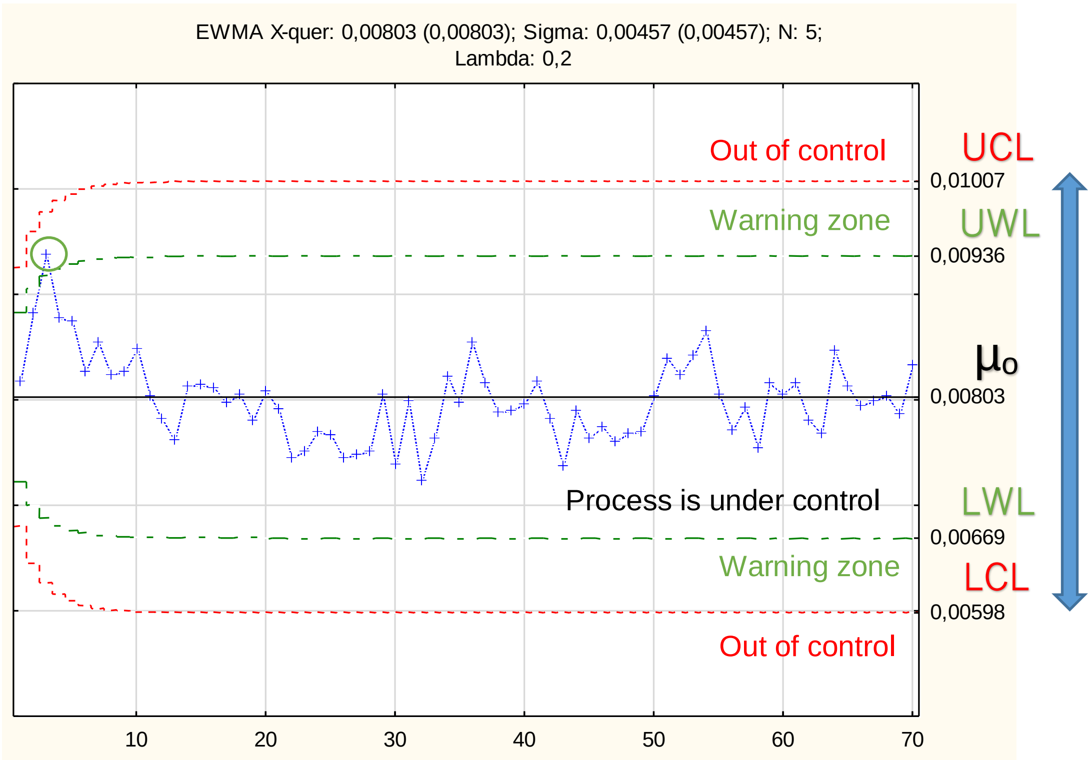

Figure 22: Statistica: EWMA control chart.

- σ = 0,00457
- X-quer (μ_0) = 0,00803
- λ = 0,2
- The limits are smaller at the beginning and become larger as the variance changes
- Control limits:

- ➢ UCL (Upper Control Limit) = 0,01007
- ➢ LCL (Lower Control Limit) = 0,00598

- Warning limits:
  - ➢ UWL (Upper Warning Limit) = 0,00936
  - ➢ LWL (Lower Warning Limit) = – 0,00669
- Points < LCL or points > UCL → out of control → stop the process → resolve the problem
- If:
  - ➢ Points < LWL or UWL > points > UCL→ a warning is issued
  - ➢ Points > UCL or points < LCL → out of control → stop the process → resolve the problem
- The third point is in the warning area → a warning is issued
- Regardless of the third point, all other points are within the process control area → the entire process is under control

The EWMA control chart results provide a detailed picture of the process stability and variability. With a standard deviation (σ) of 0.00457 and a target mean ($\bar{X}$ or $\mu 0$) of 0.00803, the chart uses a smoothing parameter (λ) of 0.2. The control limits, which start to narrow and widen as the variance increases, were set at 0.01007 for the Upper Control Limit (UCL) and 0.00598 for the Lower Control Limit (LCL). In addition, the warning limits were set at 0.00936 for the Upper Warning Limit (UWL) and -0.00669 for the Lower Warning Limit (LWL).

According to the chart, any data point that falls outside the UCL or LCL indicates the process is out of control, requiring immediate intervention. Points that fall between the warning limit and the control limit signal a warning, indicating a potential problem that should be monitored closely. In this analysis, the third point falls within the warning area, thus triggering a caution warning, while all other points remain within the control limits, indicating that the process is generally under control.

The warning indicated by the third point indicates a potential, albeit indirect, problem with the process. It can be caused by small fluctuations or emerging trends that, if not addressed, could lead to larger deviations. Potential causes of such alerts can be equipment wear, small inconsistencies in raw materials, or small variations in operating conditions. However, on this chart it can be concluded that the entire process is in the under control zone.

4.4 Hotelling's T^2 Control Charts

4.4.1 Hotelling's T^2 Control Chart: Normal Case

This practical example demonstrates the visualization of a Hotelling's T^2 control chart under general conditions, the normal case, using the R programming language. By utilizing the visualization capabilities of R, this example illustrates statistical process control techniques for monitoring multivariate data and detecting deviations from established norms. Through visually informative graphics, this example provides insight into the utility of graphics in quality management and process monitoring in an industrial context.

The provided R code snippet is designed to visualize a Hotelling's T^2 control chart, which is used to monitor the stability of multivariate processes. The data for this example is read from an Excel file containing carbon data. The following is a detailed description of each part of the code.

1. Load the necessary data

```
library(readxl)
datacarbon <- read_excel("C:/Users/Data/carbon.xlsx")
```

- library(readxl): Loads the readxl package, which provides functions to read Excel files.
- datacarbon <- read_excel("C:/Users/Data/carbon.xlsx"): Reads the Excel file containing the carbon data into the datacarbon dataframe using the read_excel function from the readxl package.

2. Calculate Hotelling's T^2 statistic

```
n <- nrow(datacarbon)
dim<- 3
```

- n <- nrow(datacarbon): Calculates the number of observations in the datacarbon dataframe.
- dim <- 3: Sets the number of dimensions (variables) being monitored, and in this example is three.

3. Calculate the initial mean vector (my0)

```
apply(as.matrix(datacarbon[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)
```

- apply(as.matrix(datacarbon[1:320,1:3]), 2, mean): Calculates the mean of the first 320 observations for the first three columns in the datacarbon dataframe. The apply function is used to apply the mean function to each column.
- my0 <- c(1.0, 1.045, 50): Manually sets the initial mean vector (my0) for the multivariate process. The values are chosen based on prior knowledge or calculated means.

4. Define the Hotelling's T^2 statistic function

```
hotelling_stat <- function(x,cov_matrix,sample_size) {
  sample_size*(as.numeric(x) - my0) %*% solve(cov_matrix) %*%
(as.numeric(x) - my0)
}
```

- hotelling_stat <- function(x, cov_matrix, sample_size): Defines a function to calculate the Hotelling T^2 statistic for a given sample.
    - x: Sample mean vector.
    - cov_matrix: Covariance matrix of the data.
    - sample_size: Size of the sample.

5. Calculate the covariance matrix and mean vector

```
cov_matrix <- cov(datacarbon[1:320,1:3])
mean_vector <- colMeans(datacarbon)
```

- cov_matrix <- cov(datacarbon[1:320,1:3]): Calculates the covariance matrix for the first 320 observations of the first three columns.
- mean_vector <- colMeans(datacarbon): Calculates the mean vector of the entire datacarbon dataframe.

6. Set the cutoff and warning limits

```
alpha<- 0.0027
cutoff <- qchisq(1-alpha,dim)
UWL<- qchisq(0.95,dim)
```

- alpha <- 0.0027: Sets the significance level for the control chart.
- cutoff <- qchisq(1 - alpha, dim): Calculates the Upper Control Limit (UCL) based on the chi-square distribution with the significance level and the number of dimensions.
- UWL <- qchisq(0.95, dim): Calculates the Upper Warning Limit (UWL) based on the chi-square distribution with a 95% confidence level and the number of dimensions.

7. Set the y-axis range

```
y_range <- 30
```

- y_range <- 30: Sets the range for the y-axis in the plot.

8. Calculate Hotelling's T^2 statistics for subgroups

```
sample_size<- 10
m<- n %/% sample_size
meanv<- matrix(ncol=dim,nrow=m)
hotelling<- vector(length=m)
for (i in 1:m){
  i1<-(i-1)*sample_size+1
  i2<-i*sample_size
  meanv[i,]<- apply(datacarbon[i1:i2,1:3],2,mean)
```

```
  hotelling[i]<- hotelling_stat(meanv[i,],cov_matrix,sample_size)
}
```

- sample_size <- 10: Sets the size of each subgroup.
- m <- n %/% sample_size: Calculates the number of subgroups.
- meanv <- matrix(ncol = dim, nrow = m): Initializes a matrix to store the mean vectors for each subgroup.
- hotelling <- vector(length = m): Initializes a vector to store the Hotelling's T^2 statistics for each subgroup.
- for (i in 1:m): Loop iterates over each subgroup to calculate the mean vector and Hotelling's T^2 statistic.
    - i1 <- (i - 1) * sample_size + 1: This calculates the starting index for the current subgroup.
    - i2 <- i * sample_size: This calculates the ending index for the current subgroup.
    - meanv[i,] <- apply(datacarbon[i1:i2, 1:3], 2, mean): This calculates the mean vector for the current subgroup.
    - hotelling[i] <- hotelling_stat(meanv[i,], cov_matrix, sample_size): This calculates the Hotelling's T^2 statistic for the current subgroup.

9. Plot Hotelling's T^2 Control Chart

```
plot(hotelling, type = "b", pch = 16, col = ifelse(hotelling > cutoff,
"red", "green"),
     main = "Hotelling's T^2 Control Chart for Carbon)", ylim = c(0,
y_range))
abline(h = cutoff, col = "red", lty = 2)
abline(h = UWL, col = "purple", lty = 2)
text(x = max(hotelling), y = cutoff, labels = "UCL", col = "red", cex =
1, pos = 3)
text(x = max(hotelling), y = UWL, labels = "UWL", col = "purple", cex =
1, pos = 3)
```

- plot(hotelling, type = "b", pch = 16, col = ifelse(hotelling > cutoff, "red", "green"), main = "Hotelling's T^2 Control Chart for Carbon)", ylim = c(0, y_range)): Plots the Hotelling's T^2 statistics for each subgroup. Points are colored red if they exceed the cutoff value, and green otherwise. The plot title is set, and the y-axis limit is set to 0-30.
- abline(h = cutoff, col = "red", lty = 2): Adds a horizontal dashed line at the UCL position in red.
- abline(h = UWL, col = "purple", lty = 2): Adds a horizontal dashed line at the UWL position in purple.
- text(x = max(hotelling), y = cutoff, labels = "UCL", col = "red", cex = 1, pos = 3): Adds a text label "UCL" near the UCL line in red.

- text(x = max(hotelling), y = UWL, labels = "UWL", col = "purple", cex = 1, pos = 3): Adds a text label "UWL" near the UWL line in purple.

This code snippet effectively sets up, calculates, and visualizes the Hotelling's T^2 control chart for multivariate process monitoring using data from an Excel file. The full code is included in the appendix.

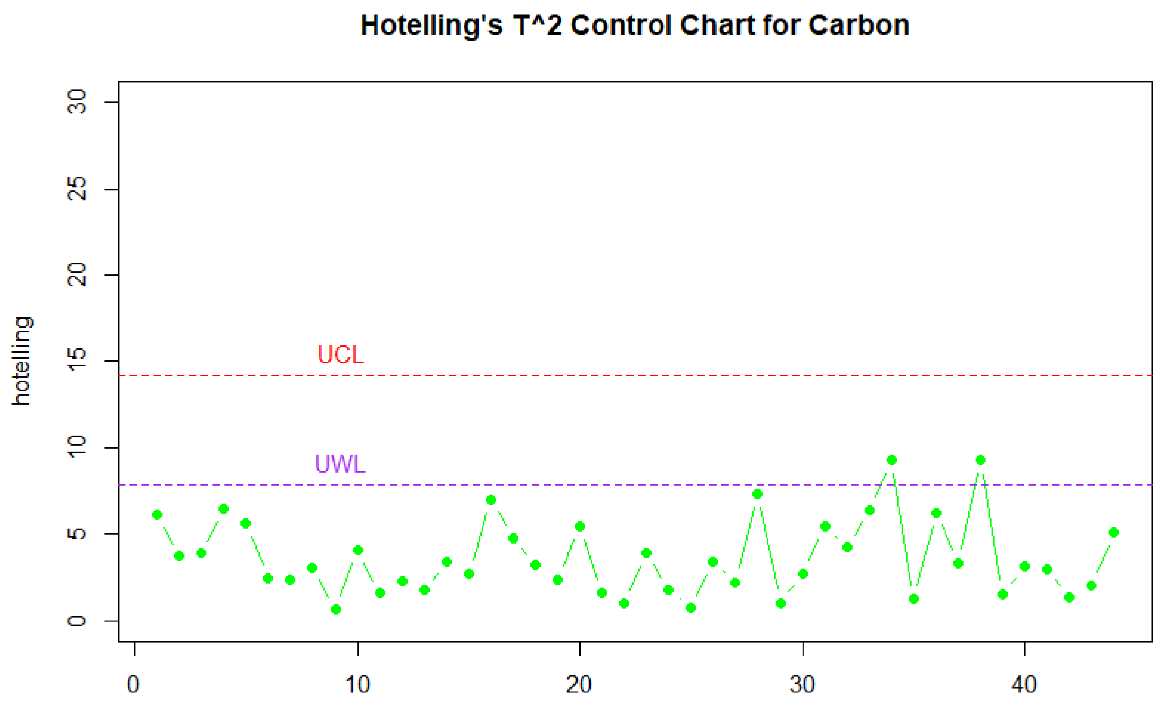

Figure 23: Hotelling's T^2 control chart for carbon.xlsx.

The data depicted in the Hotelling chart actually consists of n = 440 observations, but by sampling a size of 10, the total number of data points illustrated becomes 44 (440/10). In the illustration, the entire process on the control chart shows that the process is overall under control. Two points are seen to be within the warning zone, which indicates that the control chart issues a warning when the process is within this stage. However, it is important to note that the warning zone is still within the controlled region, which means no process is out of control.

4.4.2 Hotelling's T^2 Control Chart: Special Case (Jump)

A jump in a multivariate control chart refers to a sudden and significant change in the mean or average value of one or more variables being monitored. These changes often indicate a sudden change in the process or system being observed. Jumps can occur for a variety of reasons such as changes in equipment, process parameters, materials, or environmental factors. Detecting jumps early is critical to maintaining product quality and process efficiency.

Another application of multivariate analysis is to identify sudden changes in behavior or patterns in a data set. These charts are used in quality control and process monitoring to identify changes or deviations in multiple variables simultaneously. To modify existing data, various options are available, such as using Python, R, etc. In this section, the data will be modified using the R programming language to convert it into a unique format where there is a jump in the running process, which in practice occasionally occurs.

1. Load required library and read data

```
library(readxl)
dataxl <- read_excel("C:/Users/Data/carbon.xlsx")
```

- library(readxl): Loads the readxl package, to enable reading Excel files.
- dataxl <- read_excel("C:/Users/Data/carbon.xlsx"): Reads the Excel file located at "C:/Users/Data/carbon.xlsx" into a dataframe named dataxl.

2. Create and modify a new column "inner2"

```
dataxl$inner2 <- dataxl$inner #Create a new column in column 'd' with a
copy of the data from column 'inner'
dataxl$inner2[322:nrow(dataxl)] <- dataxl$inner2[322:nrow(dataxl)] + 0.02
#Add a value of 0.02 to each entry in column 'd' after row 321.
```

- dataxl$inner2 <- dataxl$inner: Creates a new column named inner2 in column D in the dataframe dataxl, which is a copy of the original inner column.
- dataxl$inner2[322:nrow(dataxl)] <- dataxl$inner2[322:nrow(dataxl)] + 0.023: Adds 0.02 to each entry in the inner2 column for rows starting from row 322 to the end of the dataframe.

3. Create and modify a new column "thickness2"

```
dataxl$thickness2 <- dataxl$thickness
dataxl$thickness2[322:nrow(dataxl)] <-
dataxl$thickness2[322:nrow(dataxl)] + 0.04
```

- dataxl$thickness2 <- dataxl$thickness: Creates a new column named thickness2 in column E in the dataframe dataxl, which is a copy of the original thickness column.
- dataxl$thickness2[322:nrow(dataxl)] <- dataxl$thickness2[322:nrow(dataxl)] + 0.04: Adds 0.04 to each entry in the thickness2 column for rows starting from row 322 to the end of the dataframe.

4. Create and modify a new column "length2"

```
dataxl$length2 <- dataxl$length
dataxl$length2[322:nrow(dataxl)] <- dataxl$length2[322:nrow(dataxl)] +
0.06
```

- dataxl$length2 <- dataxl$length: Creates a new column named length2 in column F in the dataframe dataxl, which is a copy of the original length column.
- dataxl$length2[322:nrow(dataxl)] <- dataxl$length2[322:nrow(dataxl)] + 0.06: Adds 0.06 to each entry in the length2 column for rows starting from row 322 to the end of the dataframe.

5. Save the Modified Data to a New Excel File

```
library(writexl)
write_xlsx(dataxl, "carbonjump.xlsx")
```

- library(writexl): Loads the writexl package, to enable writing data to Excel files.
- write_xlsx(dataxl, "carbonjump.xlsx"): Writes the modified dataframe dataxl to a new Excel file named "carbonjump.xlsx".

This code first reads the Excel file “carbon.xlsx” and then creates new columns (inner2, thickness2, and length2) as modified versions of the existing columns (inner, thickness, and length). For rows starting from 322 to the end of the data, a specific value is added to each of the new columns. Finally, the modified data was saved into a new Excel file named “carbonjump.xlsx”. The combined code to modify an excel data is in the appendix.

After all the codes have been successfully run, the latest excel data will be saved with the name "carbonjump.xlsx". The data in the latest columns, namely columns D-F (fourth to sixth columns) are copies of columns A-C; column D is a copy of column A, column E is a copy of column B, and column F is a copy of column C. The data in columns D-F are modified by adding their respective values constantly after n reaches 322, namely 0.02 in column D, 0.04 in column E, and 0.06 in column F. The following are the results of data modification before (left) and after special formatting (right).

To be able to compare the effectiveness of control charts, the data used for the case jump will be prepared into two data, namely carbonjump.xlsx and carbonjump2.xlsx. These two data will be used for all three control charts, Hotelling’s T2 control chart, MCUSUM control chart and MEWMA control chart. The code used is exactly the same as the previous method, except that the modified number is changed, with a value of +0.05 for column D, +0.07 for column E and +0.09 for column F.

Table 3: carbonjump.xlsx (left) and carbonjump2.xlsx (right).

A323 | 1,03

	A	B	C	D	E	F
315	1	1,26	49,99	1	1,26	49,99
316	1,02	1,05	50,38	1,02	1,05	50,38
317	1,07	1,23	50,02	1,07	1,23	50,02
318	1,06	1,07	50,4	1,06	1,07	50,4
319	1,04	1,22	49,76	1,04	1,22	49,76
320	1,05	1,31	50,41	1,05	1,31	50,41
321	1,01	0,97	50,14	1,01	0,97	50,14
322	1,06	1,15	50,11	1,06	1,15	50,11
323	1,03	0,96	49,88	1,05	1	49,94
324	0,98	1,18	50,21	1	1,22	50,27
325	1	0,99	50,04	1,02	1,03	50,1
326	1,01	0,96	50,22	1,03	1	50,28
327	1,03	1,1	50,12	1,05	1,14	50,18
328	1	1,08	50,38	1,02	1,12	50,44
329	0,93	1,05	50,23	0,95	1,09	50,29
330	1,15	1,14	50,29	1,17	1,18	50,35

A323 | 1,03

	A	B	C	D	E	F
315	1	1,26	49,99	1	1,26	49,99
316	1,02	1,05	50,38	1,02	1,05	50,38
317	1,07	1,23	50,02	1,07	1,23	50,02
318	1,06	1,07	50,4	1,06	1,07	50,4
319	1,04	1,22	49,76	1,04	1,22	49,76
320	1,05	1,31	50,41	1,05	1,31	50,41
321	1,01	0,97	50,14	1,01	0,97	50,14
322	1,06	1,15	50,11	1,06	1,15	50,11
323	1,03	0,96	49,88	1,08	1,03	49,97
324	0,98	1,18	50,21	1,03	1,25	50,3
325	1	0,99	50,04	1,05	1,06	50,13
326	1,01	0,96	50,22	1,06	1,03	50,31
327	1,03	1,1	50,12	1,08	1,17	50,21
328	1	1,08	50,38	1,05	1,15	50,47
329	0,93	1,05	50,23	0,98	1,12	50,32
330	1,15	1,14	50,29	1,2	1,21	50,38

The output is shown on the right side of table 3, with carbonjump displayed on the left and carbonjump2 on the right. Both datasets include different additional numeric values added to each entry after row 323. The next step is to visualize the data using multivariate Hotelling's T2 control charts in R. The code for this visualization is nearly identical to the code used for the standard Hotelling's T2 control chart. The detailed, step-by-step code is provided below.

1. Load the necessary data

```
library(readxl)
datajump <- read_excel("C:/Users/Data/carbonjump.xlsx") # or
datajump <- read_excel("C:/Users/Data/carbonjump2.xlsx")
```

- datajump <- read_excel("C:/Users/Data/carbonjump.xlsx") or datajump <- read_excel("C:/Users/Data/carbonjump2.xlsx"): Pick which version wanted to get read first, carbonjump.xlsx or carbonjump2.xlsx, and read it into dataframe "datajump".

2. Calculate Hotelling's T^2 statistic

```
n <- nrow(datajump)
dim<- 3
```

- n <- nrow(datajump): Calculates the number of observations in the datajump dataframe.

3. Calculate the initial mean vector (my0)

4. Define the Hotelling's T^2 statistic function

5. Calculate the covariance matrix and mean vector

```
cov_matrix <- cov(datajump[1:320,4:6])
mean_vector <- colMeans(datajump)
```

- cov_matrix <- cov(datajump1:320,4:6]): It calculates the covariance matrix for the first 320 observations from the fourth column to the sixth column (4:6).
- mean_vector <- colMeans(datajump): Calculates the mean vector of the entire datajump dataframe.

6. Set the cutoff and warning limits

7. Set the y-axis range

8. Calculate Hotelling's T^2 statistics for subgroups

```
sample_size<- 10
m<- n %/% sample_size
meanv<- matrix(ncol=dim,nrow=m)
hotelling<- vector(length=m)
for (i in 1:m){
  i1<-(i-1)*sample_size+1
  i2<-i*sample_size
  meanv[i,]<- apply(datajump[i1:i2,4:6],2,mean)
  hotelling[i]<- hotelling_stat(meanv[i,],cov_matrix,sample_size)
}
```

- meanv[i,] <- apply(datajump[i1:i2, 4:6], 2, mean): Calculates the mean vector for the aimed subgroup.

9. Plot Hotelling's T^2 Control Chart

```
plot(hotelling, type = "b", pch = 16, col = ifelse(hotelling > cutoff, "red", "green"),
     main = "Hotelling's T^2 Control Chart for Carbon: Jump", ylim = c(0, y_range))
abline(h = cutoff, col = "red", lty = 2)
abline(h = UWL, col = "purple", lty = 2)
text(x = max(hotelling), y = cutoff, labels = "UCL", col = "red", cex = 1, pos = 3)
text(x = max(hotelling), y = UWL, labels = "UWL", col = "purple", cex = 1, pos = 3)
```

- plot(hotelling, type = "b", pch = 16, col = ifelse(hotelling > cutoff, "red", "green"), main = "Hotelling's T^2 Control Chart for Carbon: Jump)", ylim = c(0, y_range)): The function is nearly the same as the function in the normal case, the difference is only in the title of the plot, which was previously only "Hotelling's T^2 Control Chart for Carbon", changed into "Hotelling's T^2 Control Chart for Carbon: Jump)".

The steps used are almost the same as the code used for the normal case, the main difference is the use of the carbonjump data into the "datajump" dataframe, the focus of column usage from columns 1-3 to columns 4-6, and another difference is the title on the plot (the full code is in the appendix).

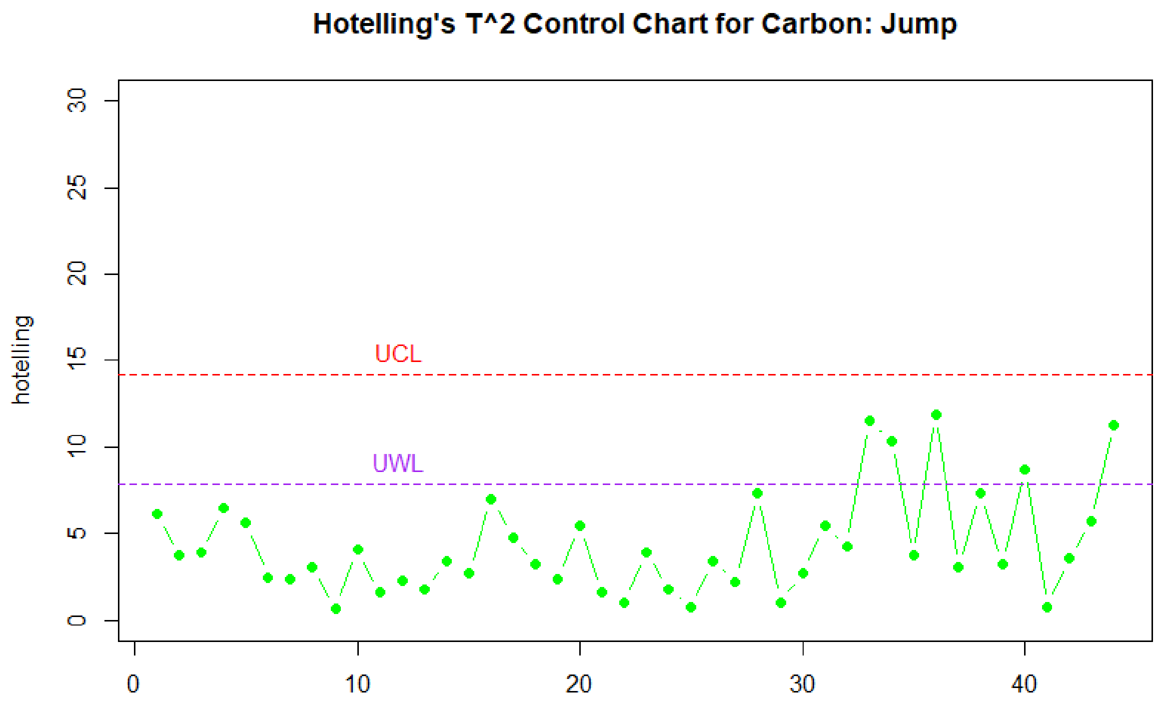

Figure 24: Hotelling's T^2 control chart for carbonjump.xlsx.

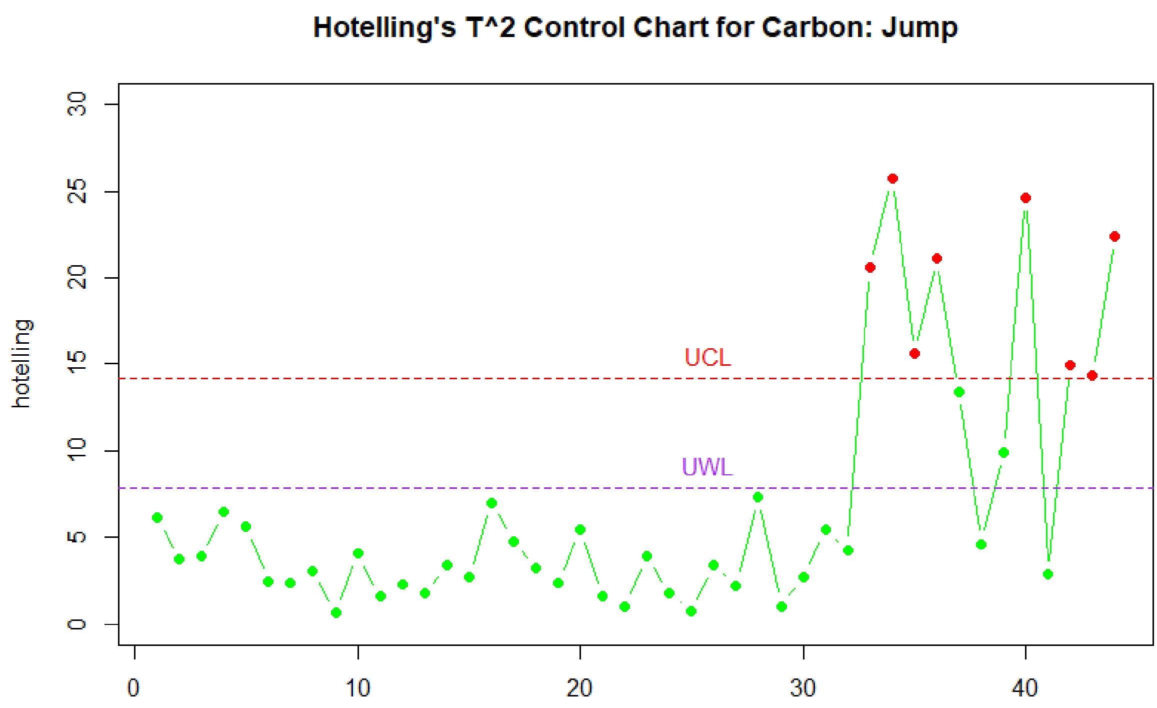

Figure 25: Hotelling's T^2 control chart for carbonjump2.xlsx.

The Hotelling's T^2 control chart results show the same results as the control chart in the normal case until n<330 (m<33). After that, there is a change in the movement of the control chart compared to normal conditions, where all points are under control. On both results at m_{33}, the change in data movement has begun to be seen. In figure 25, although there is a slight change and even reach warning zone, the overall movement of the process is still in control. Whereas in figure 26, a massive "jump" movement occurred, causing the process to be located in the out of control zone. From these two examples, it can be concluded that small shifts or changes in the process are difficult to detect using Hotelling's T^2 control chart, but Hotelling's T^2 chart remains effective even if there are very large changes in the data in a process.

In real-world practice, sudden jumps in the Hotelling's T^2 control chart can occur for many reasons. One common cause is a significant change in process mean or variability, which can result from factors such as changes in raw materials, equipment breakdowns, operator error, or shifts in environmental conditions. These changes can introduce new sources of variation that were not present in the initial data set, leading to jumps observed on the control chart.

When such anomalies occur, it is imperative to immediately investigate the root cause and take corrective action to return the process to a stable state. This may involve identifying and addressing the underlying factors that contributed to the change, such as recalibrating equipment, retraining operators, adjusting process parameters, or implementing quality control measures. In addition, it is critical to continue monitoring the process closely to ensure that corrective actions effectively mitigate the problem and prevent further deviation from the desired performance. Regular review and adjustment of control diagram parameters may also be necessary to adapt to long-term changes in the process.

4.4.3 Hotelling's T^2 Control Chart: Special Case (Shift)

A shift, on the other hand, is a gradual change in the mean or average value of one or more variables over time. Unlike jumps, shifts occur more slowly and can be indicative of long-term changes in the process. Shifts can be caused by factors such as equipment wear, changes in operator behavior, or adjustments made to the process. Detecting shift allows operators to intervene and make necessary adjustments to prevent quality issues or deviations from specifications. In this section, the carbon data is formatted using the R programming language to convert it into a unique format where there is a constant shift in the running process. Here is the code to modify it incrementally.

1. Load Required Libraries and Read Data

```
library(readxl)
dataxl <- read_excel("C:/Users/Data/carbon.xlsx")
```

- library(readxl): Loads the readxl package, which is used for reading Excel files.
- dataxl <- read_excel("C:/Users/Data/carbonedited.xlsx"): Reads the Excel file located at "C:/Users/Data/carbonedited.xlsx" into a dataframe named dataxl.

2. Create and modify a new column “inner3”

```
dataxl$inner3 <- dataxl$inner #Create a new column in column 'd' with a
copy of the data from column 'inner'
dataxl$inner3[321:440] <- dataxl$inner3[321:440] + (0.0002 * (1:120)) #
Add 0.0002 to the value of every entry in column D after row 250
```

- dataxl$inner3 <- dataxl$inner: Creates a new column named inner3 in column D in the dataframe dataxl, which is a copy of the original inner column.
- dataxl$inner3[321:440] <- dataxl$inner3[321:440] + (0.0002 * (1:120)): Adds an incrementing value to each entry in the inner3 column for rows 321 to 440. The incrementing value is calculated as 0.0002 * (1:120), which means that for row 321, 0.0002 is added, for row 322, 0.0004 is added, and so on, until for row 440, 0.024 is added.

3. Create and modify a new column “thickness3”

```
dataxl$thickness3 <- dataxl$thickness
dataxl$thickness3[321:440] <- dataxl$thickness3[321:440] + (0.0003 *
(1:120))
```

- dataxl$thickness3 <- dataxl$thickness: Creates a new column named thickness3 in column E in the dataframe dataxl, which is a copy of the original thickness column.
- dataxl$thickness3[321:440] <- dataxl$thickness3[321:440] + (0.0003 * (1:120)): Adds an incrementing value to each entry in the thickness3 column for rows 321 to 440. The incrementing value is calculated as 0.0003 * (1:120), which means that for row 321, 0.0003 is added, for row 322, 0.0006 is added, and so on, until for row 440, 0.036 is added.

4. Create and modify a new column “length3”

```
dataxl$length3 <- dataxl$length
dataxl$length3[321:440] <- dataxl$length3[321:440] + (0.0004 * (1:120))
```

- dataxl$length3 <- dataxl$length: Creates a new column named length3 in column F in the dataframe dataxl, which is a copy of the original length column.
- dataxl$length3[321:440] <- dataxl$length3[321:440] + (0.0004 * (1:120)): Adds an incrementing value to each entry in the length3 column for rows 321 to 440. The incrementing value is calculated as 0.0004 * (1:120), which means that for row 321, 0.0004 is added, for row 322, 0.0008 is added, and so on, until for row 440, 0.048 is added.

5. Save the modified data to a new excel file

```
library(writexl)
write_xlsx(dataxl, "carbonshift.xlsx")
```

- library(writexl): Loads the writexl package, which is used for writing data to Excel files.
- write_xlsx(dataxl, "carbonshift.xlsx"): Writes the modified dataframe dataxl to a new Excel file named "carbonshift.xlsx".

This code first reads the Excel file "carbon.xlsx" and then creates new columns (inner3, thickness3, and length3) as modified versions of the existing columns (inner, thickness, and length). For rows 321 to 440, the specific values that increase with each row are added to each of the new columns. Finally, the modified data is saved into a new Excel file named "carbonshift.xlsx" (full code is in the appendix).

After all the codes have been successfully run, the latest excel data will be saved with the name "carbonshift.xlsx". The data in the latest columns, namely columns D-F (fourth to sixth columns) are copies of columns A-C; column D is a copy of column A, column E is a copy of column B, and column F is a copy of column C. The data in columns D-F are modified by adding their respective values incrementally after n reaches 321, namely 0.0002 in column D, 0.0003 in column E, and 0.0004 in column F. The following are the results of data modification before and after special formatting.

Furthermore, for the use of the second shift data, namely carbonshift2.xlsx, similar modifications were made. The values entered will be different, namely 0.0005 for column D, 0.0006 for column E and 0.0007 for column F. Both data for the shift case will also be applied to the three control charts, Hotelling's T^2 control chart, MCUSUM control chart and MEWMA control chart.

Table 4: carbonshift.xlsx (left) and carbonshift2.xlsx (right).

A322 fx 1,06

	A	B	C	D	E	F
315	1	1,26	49,99	1	1,26	49,99
316	1,02	1,05	50,38	1,02	1,05	50,38
317	1,07	1,23	50,02	1,07	1,23	50,02
318	1,06	1,07	50,4	1,06	1,07	50,4
319	1,04	1,22	49,76	1,04	1,22	49,76
320	1,05	1,31	50,41	1,05	1,31	50,41
321	1,01	0,97	50,14	1,01	0,97	50,14
322	1,06	1,15	50,11	1,0602	1,1503	50,1104
323	1,03	0,96	49,88	1,0304	0,9606	49,8808
324	0,98	1,18	50,21	0,9806	1,1809	50,2112
325	1	0,99	50,04	1,0008	0,9912	50,0416
326	1,01	0,96	50,22	1,011	0,9615	50,222
327	1,03	1,1	50,12	1,0312	1,1018	50,1224
328	1	1,08	50,38	1,0014	1,0821	50,3828
329	0,93	1,05	50,23	0,9316	1,0524	50,2332
330	1,15	1,14	50,29	1,1518	1,1427	50,2936

A322 fx 1,06

	A	B	C	D	E	F
315	1	1,26	49,99	1	1,26	49,99
316	1,02	1,05	50,38	1,02	1,05	50,38
317	1,07	1,23	50,02	1,07	1,23	50,02
318	1,06	1,07	50,4	1,06	1,07	50,4
319	1,04	1,22	49,76	1,04	1,22	49,76
320	1,05	1,31	50,41	1,05	1,31	50,41
321	1,01	0,97	50,14	1,01	0,97	50,14
322	1,06	1,15	50,11	1,0605	1,1506	50,1107
323	1,03	0,96	49,88	1,031	0,9612	49,8814
324	0,98	1,18	50,21	0,9815	1,1818	50,2121
325	1	0,99	50,04	1,002	0,9924	50,0428
326	1,01	0,96	50,22	1,0125	0,963	50,2235
327	1,03	1,1	50,12	1,033	1,1036	50,1242
328	1	1,08	50,38	1,0035	1,0842	50,3849
329	0,93	1,05	50,23	0,934	1,0548	50,2356
330	1,15	1,14	50,29	1,1545	1,1454	50,2963

The results of the data modification can be seen in table 4, with carbonshift on the left and carbonshift2 on the right. It can be seen that the value of each new data starts to change after n322, where there are increasing values of each variable. The code used to visualize this data control chart is almost the same as the jump case, the only thing that changes is the data loaded and the title on the plot. Here is the step-by-step, briefly explaining the changes required.

1. Load the necessary data

```
library(readxl)
datashift <- read_excel("C:/Users/Data/carbonshift.xlsx") # or
datashift <- read_excel("C:/Users/Data/carbonshift2.xlsx")
```

- datashift <- read_excel("C:/Users/Data/carbonshift.xlsx") or datashift <- read_excel("C:/Users/Data/carbonshift2.xlsx"):: Pick only one, carbonshift.xlsx or carbonshift2 to get read into dataframe “datashift”.

2. Calculate Hotelling’s T^2 statistic

```
n <- nrow(datashift)
dim<- 3
```

- n <- nrow(datashift): Calculates the number of observations in dataframe “datashift”.

3. Calculate the initial mean vector (my0)

4. Define the Hotelling’s T^2 statistic function

5. Calculate the covariance matrix and mean vector

```
cov_matrix <- cov(datashift[1:320,4:6])
mean_vector <- colMeans(datashift)
```

- cov_matrix <- cov(datashift[1:320,4:6]): Dataframe “datashift” is used.
- mean_vector <- colMeans(datashift): Calculates the mean vector of the entire datashift dataframe.

6. Set the cutoff and warning limits

7. Set the y-axis range

8. Calculate Hotelling’s T^2 statistics for subgroups

```
sample_size<- 10
m<- n %/% sample_size
meanv<- matrix(ncol=dim,nrow=m)
hotelling<- vector(length=m)
for (i in 1:m){
  i1<-(i-1)*sample_size+1
  i2<-i*sample_size
  meanv[i,]<- apply(datashift[i1:i2,4:6],2,mean)
  hotelling[i]<- hotelling_stat(meanv[i,],cov_matrix,sample_size)
```

- meanv[i,]<- apply(datashift[i1:i2,4:6],2,mean): Datashift dataframe, column 4-6.

9. Plot Hotelling's T^2 Control Chart

```
plot(hotelling, type = "b", pch = 16, col = ifelse(hotelling > cutoff,
"red", "green"),
     main = "Hotelling’s T^2 Control Chart for Carbon: Shift", ylim =
c(0, y_range))
abline(h = cutoff, col = "red", lty = 2)
abline(h = UWL, col = "purple", lty = 2)
text(x = max(hotelling), y = cutoff, labels = "UCL", col = "red", cex =
1, pos = 3)
text(x = max(hotelling), y = UWL, labels = "UWL", col = "purple", cex =
1, pos = 3)
```

- plot(hotelling, type = "b", pch = 16, col = ifelse(hotelling > cutoff, "red", "green"), main = "Hotelling's T^2 Control Chart for Carbon: Shift)", ylim = c(0, y_range)): Changing the title of the plot, which was previously only "Hotelling's T^2 Control Chart for Carbon: Jump", to "Hotelling's T^2 Control Chart for Carbon: Shift)".
- Full code is in the appendix.

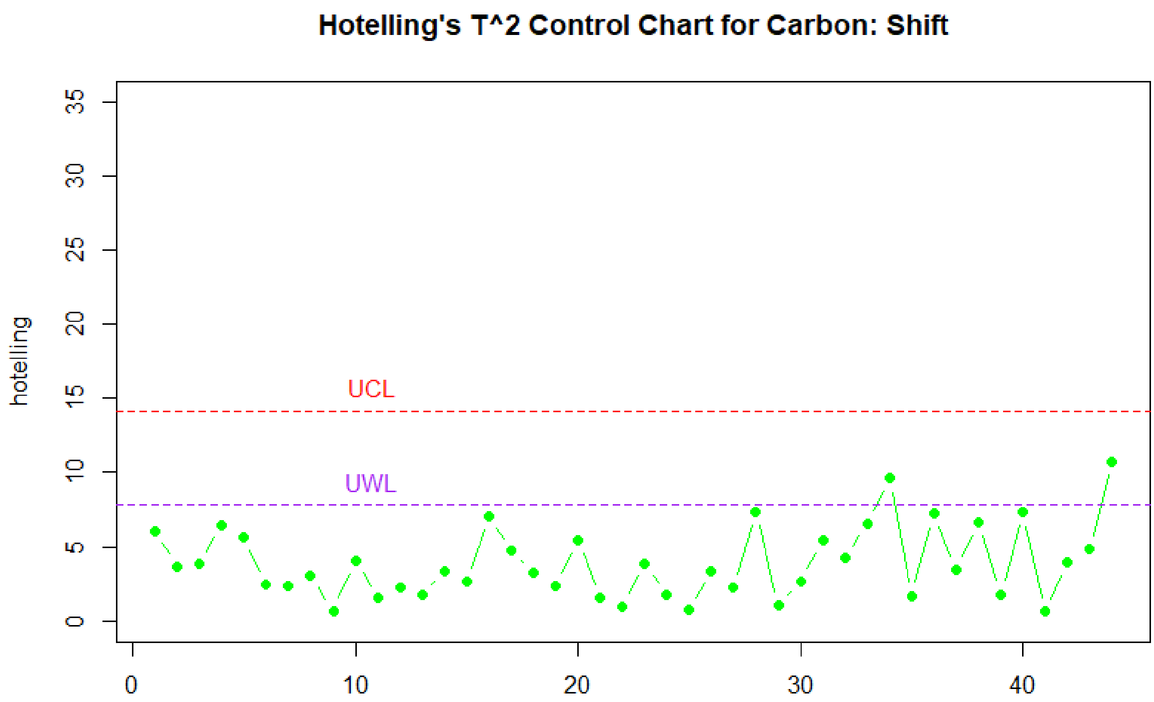

Figure 26: Hotelling's T^2 control chart for carbonshift.xlsx.

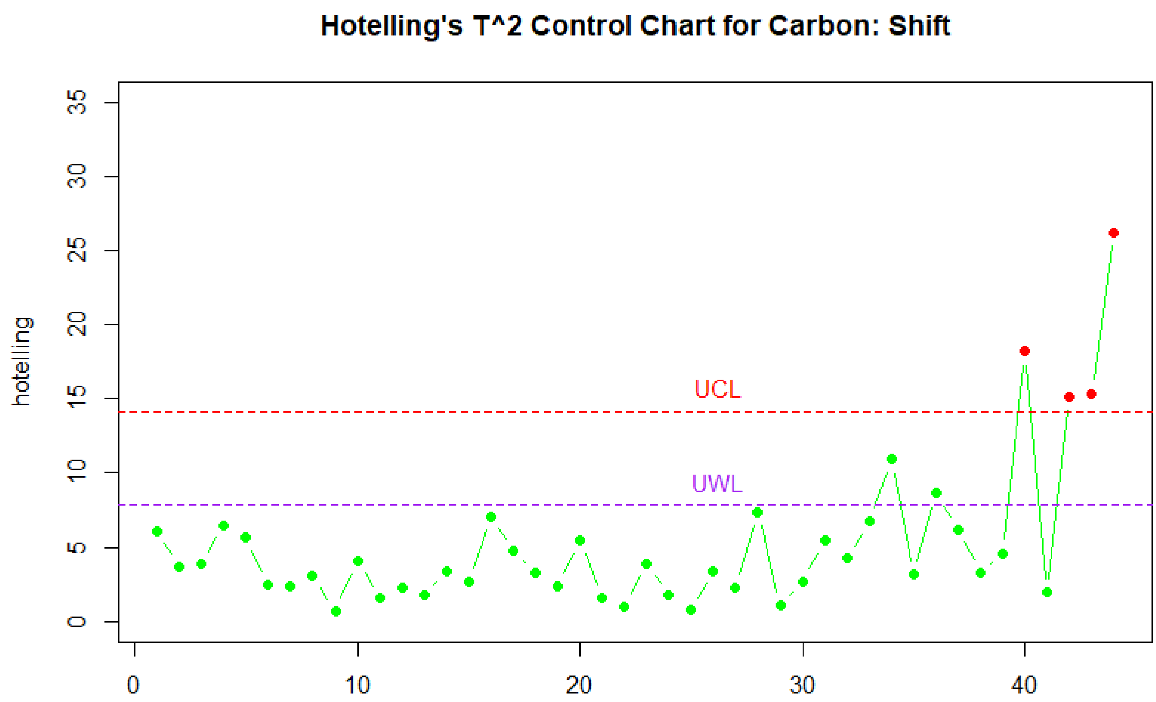

Figure 27: Hotelling's T^2 control chart for carbonshift2.xlsx.

In this case, a significant difference was not initially evident in both graphs, but gradually deviations began to be detected and the process continued to gradually increase. In chart 26, since the process changes are minimal, the changes are not very detectable, while in chart 27, despite the large changes, Hotelling's T^2 is still able to describe the process with satisfactory results.

Shifts in real-life practice usually occur due to various factors such as changes in raw materials, equipment malfunction, human error, or environmental conditions. When such shifts occur, it is imperative to immediately investigate the root cause and implement corrective actions to restore stability. This may include rechecking equipment, providing additional training to staff, refining process parameters, or strengthening quality control measures. The most important thing is to continue to monitor and review the process regularly to prevent future deviations and enforce consistent performance.

4.5 MCUSUM Control Charts

4.5.1 MCUSUM Control Chart: Normal Case

The illustrative example to be shown in this section is the implementation of Multivariate Cumulative Sum (MCUSUM) control charts using the R programming language. The visualization is facilitated by using the qcr-package. In R, a package is a collection of functions, data sets, and documentation bundled together to solve a specific problem or perform a specific task. Packages extend the functionality of R by providing pre-written code that can be easily used by users for various purposes, such as data manipulation, statistical analysis, visualization, and more.

Packages are beneficial in R because they allow users to access a wide variety of tools and functions without the need to write all the code from scratch. Packages promote code reuse, efficiency, and collaboration by allowing users to share their work with others and leverage the work of the wider R community. In addition, packages often undergo rigorous testing and development, to ensure reliability and accuracy in their implementation. This example focuses on monitoring the quality of carbon data from Excel files. Here is a detailed explanation of each code section step by step.

1. Install and load the qcr package

```
install.packages("qcr")
library(qcr)
```

- install.packages("qcr"): This command installs the qcr package from CRAN if it is not already installed.
- library(qcr): Loads the qcr package into the R session, making its functions available for use.

2. Preparing data for MCUSUM

```
library(readxl)
datacarbon <- read_excel("C:/Users/Data/carbon.xlsx")
```

- library(readxl): Loads the readxl package, which provides functions to read Excel files.
- datacarbon <- read_excel("C:/Users/Data/carbon.xlsx"): Reads the Excel file containing the carbon data into the datacarbon dataframe.

3. Preparing the data for multivariate quality control

```
data_selected <- mqcd(datacarbon)
```

- data_selected <- mqcd(datacarbon): Prepares the datacarbon for multivariate quality control analysis using the mqcd function from the qcr package into data_selected dataframe.

4. Determine initial mean vector (my0)

```
# Determine my0
apply(as.matrix(datacarbon[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)´
```

- apply(as.matrix(datacarbon[1:320,1:3]), 2, mean): Calculates the mean of the first 320 observations for the first three columns in the datacarbon dataframe. The apply function is used to apply the mean function to each column.

- my0 <- c(1.0, 1.045, 50): To manually sets the initial mean vector (my0) for the multivariate process. The values are chosen based on calculated means.

5. Determine number of dimensions

```
dim<- 3
```

- dim <- 3: Sets the number of dimensions (variables) being monitored, which in this case is three.

6. Compute warning limits

```
UWL<- qchisq(0.95,dim)
```

- UWL <- qchisq(0.95, dim): Calculates the Upper Warning Limit (UWL) based on the chi-square distribution with a 95% confidence level and the number of dimensions. qchisq represents the quantile function for the chi-squared distribution.

7. Compute MCUSUM

```
mcusum_result <- mqcs.mcusum(data_selected, limits = NULL,
                           Xmv = my0,
                           S = NULL,
                           k = 0.5,
                           h = 10,
                           method = "sw")
```

- mcusum_result <- mqcs.mcusum(...): Calculates the MCUSUM statistics for the multivariate data. The parameters used are:
  - data_selected: The prepared multivariate data.
  - limits = NULL: No predefined control limits are used.
  - Xmv = my0: Initial mean vector.
  - S = NULL: Covariance matrix, set to NULL to use the default.
  - k = 0.5: Reference value, determining the sensitivity of the chart.
  - h = 10: Decision interval, which defines the control limit for the MCUSUM chart.
  - method = "sw": The method used for calculation, in this case, "sw".

8. Summary of MCUSUM result

```
summary(mcusum_result)
```

- summary(mcusum_result): This command provides a summary of the MCUSUM results, including statistics and key findings.

9. Plot MCUSUM chart

```
plot(mcusum_result, title = "MCUSUM Control Chart for Carbon")
abline(h = UWL, col = "darkorchid1", lty = 2)
```

```
text(x = 400, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 3)
```

- plot(mcusum_result, title = "MCUSUM Control Chart for Carbon"): Plots the MCUSUM control chart for the carbon data with the specified title.
- abline(h = UWL, col = "darkorchid1", lty = 2): Adds a horizontal line at the UWL position on the plot. The line is colored darkorchid1 and uses a dashed line type (lty = 2).
- text(x = 400, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos = 3): Adds a text label "UWL" near the UWL line on the plot. The text is positioned at x = 400 and y = UWL, colored darkorchid1, with a character expansion size of 1, and placed above the line (pos = 3).

This code snippet effectively organizes, calculates, and visualizes MCUSUM control charts for multivariate process monitoring using data from Excel files (full code in the appendix).

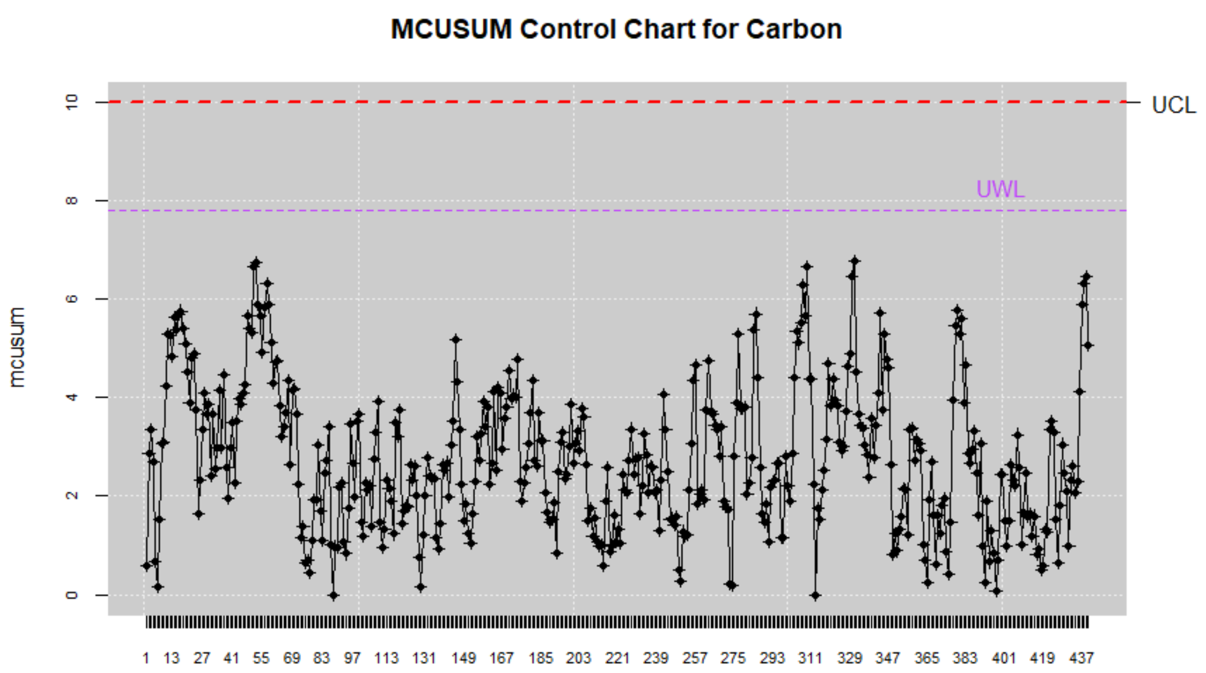

Figure 28: MCUSUM control chart for carbon.xlsx.

The Upper Control Limit (UCL) is represented by a red dotted line with a value of 10, this is the threshold at which the process is considered out of control. Any point that exceeds this limit signifies a significant shift in the process mean that requires immediate investigation and corrective action. The Upper Warning Limit (UWL) is represented by a purple dotted line with a value of about 8, this limit serves as an early warning indicator. Points that exceed this limit indicate that the process may deviate from the target and should be closely monitored to prevent it from getting out of control.

The majority of the data points are within the process control area, below the UWL, which indicates that the process is generally stable and within the control limits. A few points approaching the UWL, especially in the latter part of the graph, indicate an increasing trend in cumulative deviation that requires closer monitoring. However, no point exceeds the UCL, which indicates that the entire process is in the under control zone.

4.5.2 MCUSUM Control Chart: Special Case (Jump)

In this case, the excel data used is carbonjump.xlsx and carbonjump2.xlsx, the data that is specifically modified for the MCUSUM jump case (is identical to section 4.4.2 Hotelling's T^2: Jump). Both carbonjump.xlsx and carbonjump2.xlsx data will be visualized into the MCUSUM control chart. The next step is to program R to visualize the MCUSUM in both jump cases.

1. Install and load the qcr package

2. Preparing data for MCUSUM

```
library(readxl)
datajump <- read_excel("C:/Users/Data/carbonjump.xlsx") # or
datajump <- read_excel("C:/Users/Data/carbonjump2.xlsx")
```

- datajump <- read_excel("C:/Users/Data/carbonjump.xlsx") or datajump <- read_excel("C:/Users/Data/carbonjump2.xlsx): Decide, which excel data will get read first and read that data into the datajump dataframe.

3. Preparing the data for multivariate quality control

```
mcusum_jump <- mqcd(datajump[,c(4:6)])
```

- mcusum_jump <- mqcd(datajump[,c(4:6)]): Prepares the datajump (specific column 4-column 6) for multivariate quality control analysis using the mqcd function from the qcr package into mcusum_jump dataframe.

4. Determine initial mean vector (my0)

```
# Determine my0
apply(as.matrix(datajump[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)
```

- apply(as.matrix(datajump [1:320,1:3]), 2, mean): Calculates the mean of the first 320 observations for the first three columns in the datajump dataframe. The apply function is used to apply the mean function to each column.

5. Determine number of dimensions

6. Compute warning limits

7. Compute MCUSUM

```
mcusum_result <- mqcs.mcusum(mcusum_jump, limits = NULL,
                             Xmv = my0,
                             S = NULL,
                             k = 0.5,
                             h = 10,
                             method = "sw")
```

- mcusum_result <- mqcs.mcusum(mcusum_jump...): Calculates the MCUSUM statistics for the multivariate data using dataframe mcusum_jump.

8. Summary of MCUSUM result

9. Plot MCUSUM chart

```
plot(mcusum_result, title = "MCUSUM Control Chart for Carbon: Jump")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 400, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 3)
```

- plot(mcusum_result, title = "MCUSUM Control Chart for Carbon: Jump"): Plots the MCUSUM control chart for the carbonjump data with the specified title.

This code snippet effectively organizes, calculates, and visualizes MCUSUM control charts for multivariate process monitoring using data from Excel files (full code in the appendix).

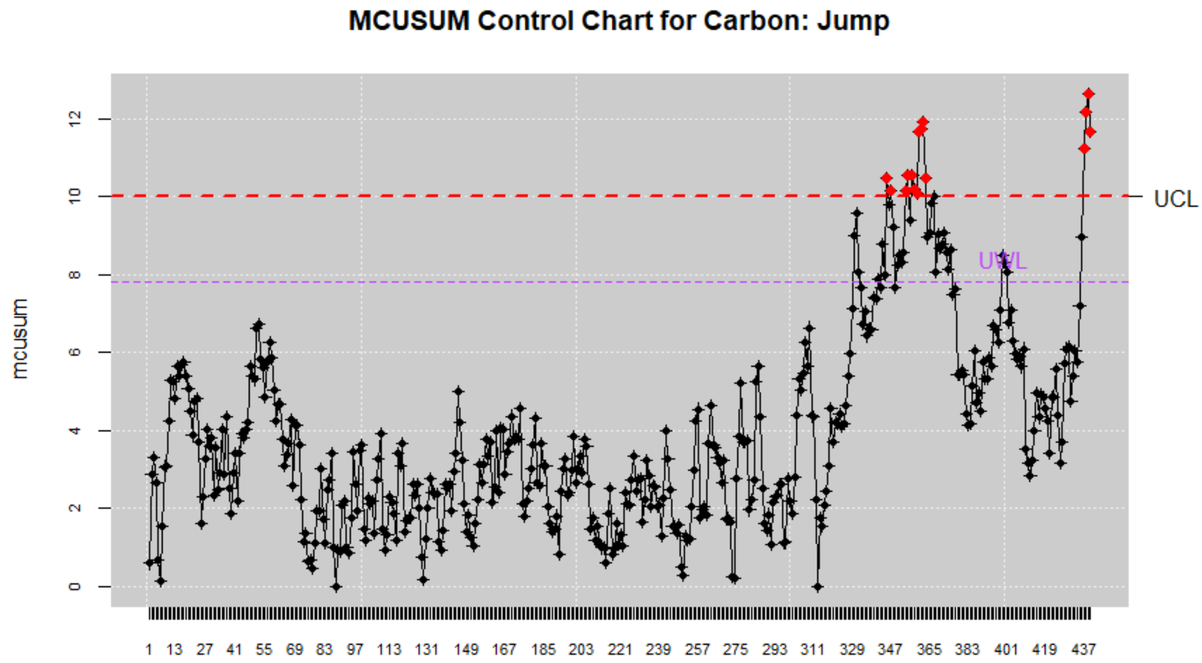

Figure 29: MCUSUM control chart for carbonjump.xlsx.

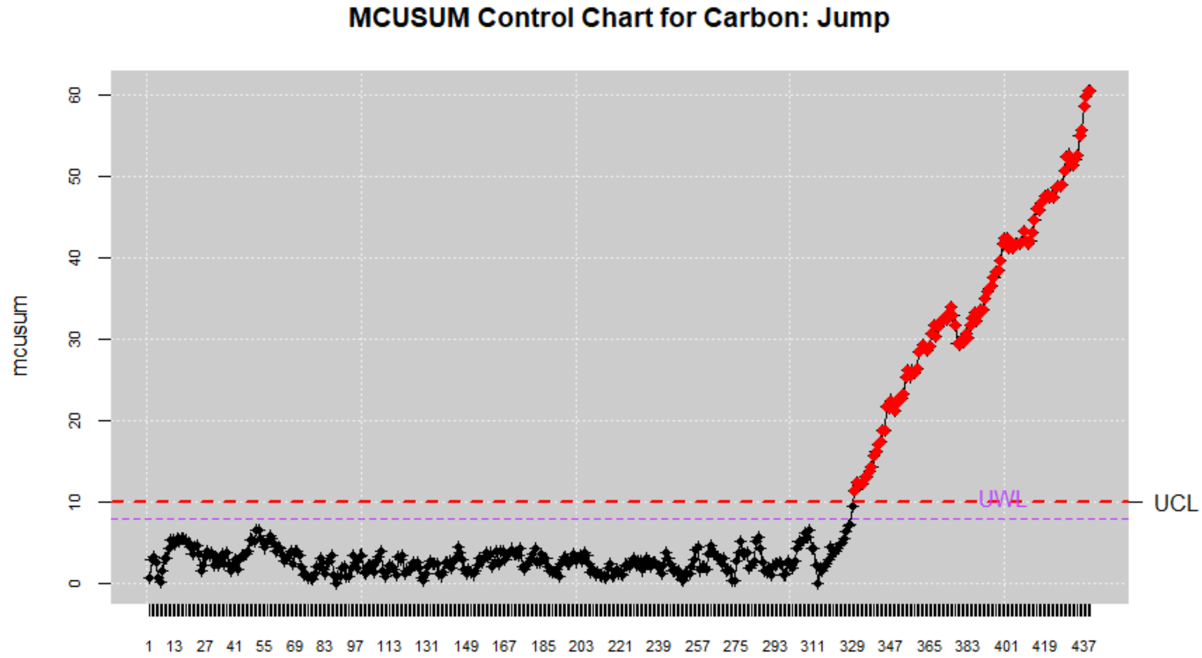

Figure 30: MCUSUM control chart for carbonjump2.xlsx.

Figure 29 is the visualization result of the "carbonjump" data and figure 30 of the "carbonjump2" data using R. These MCUSUM control charts help in identifying shifts and trends in the process mean over time. During the initial phase (1 < n < 322), the control diagram results are identical to the normal case, the process appears stable and within the control limits, showing no significant deviations. However, after point 322, changes begin to occur in the process. In figure 29, although the carbonjump data has a modest increase, the change can be identified with the MCUSUM chart very well. Whereas in figure 30, carbonjump2 data, the illustration results are very extreme, so it can be concluded that MCUSUM is a control chart that is sensitive to small changes.

4.5.3 MCUSUM Control Chart: Special Case (Shift)

The excel data used for the shift case is the same as section 4.4.3 (carbonshift.xlsx and carbonshift2.xlsx). Step-by-step explanations will be explained with an emphasis on changes only.

1. Install and load the qcr package

2. Preparing data for MCUSUM

```
library(readxl)
datashift <- read_excel("C:/Users/Data/carbonshift.xlsx") # or
datashift <- read_excel("C:/Users/Data/carbonshift2.xlsx")
```

- datashift <- read_excel("C:/Users/Data/carbonshift.xlsx") or datashift <- read_excel("C:/Users/Data/carbonshift2.xlsx): Decide, which excel data is needed first and read that data into the datashift dataframe.

3. Preparing the data for multivariate quality control

```
mcusum_shift <- mqcd(datashift[,c(4:6)])
```

- mcusum_shift <- mqcd(datashift[,c(4:6)]): Prepares the datashift, column 4-column 6 for multivariate quality control analysis using the mqcd function from the qcr package into mcusum_shift dataframe.

4. Determine initial mean vector (my0)

```
# Determine my0
apply(as.matrix(datashift[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)
```

- apply(as.matrix(datashift [1:320,1:3]), 2, mean): Calculates the mean of the first 320 observations for the first three columns in the datashift dataframe. to apply the mean function to each column, the apply function is used.

5. Determine number of dimensions

6. Compute warning limits

7. Compute MCUSUM

```
mcusum_result <- mqcs.mcusum(mcusum_shift, limits = NULL,
                             Xmv = my0,
                             S = NULL,
                             k = 0.5,
                             h = 10,
                             method = "sw")
```

- mcusum_result <- mqcs.mcusum(mcusum_shift...): Calculates the MCUSUM statistics for the multivariate data, using mcusum_shift dataframe.

8. Summary of MCUSUM result

9. Plot MCUSUM chart

```
plot(mcusum_result, title = "MCUSUM Control Chart for Carbon: Shift")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 383, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 3)
```

plot(mcusum_result, title = "MCUSUM Control Chart for Carbon: Shift"): Plots the MCUSUM control chart while specifying its title for “shift” case, for carbonshift.

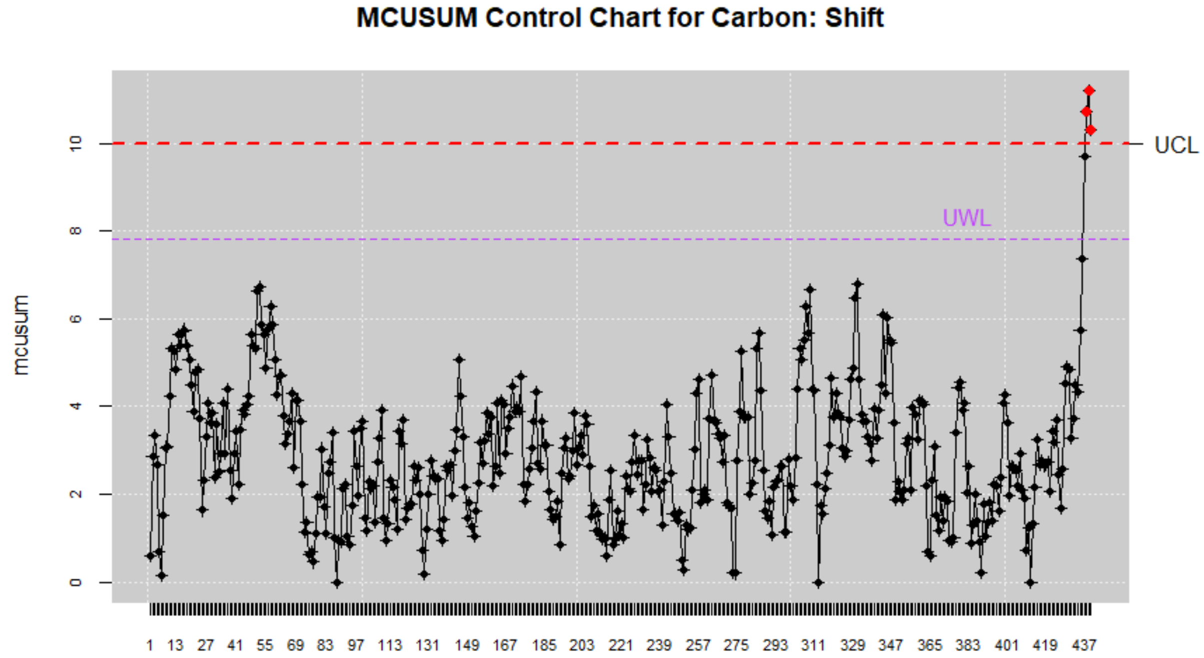

Figure 31: MCUSUM control chart for carbonshift.xlsx.

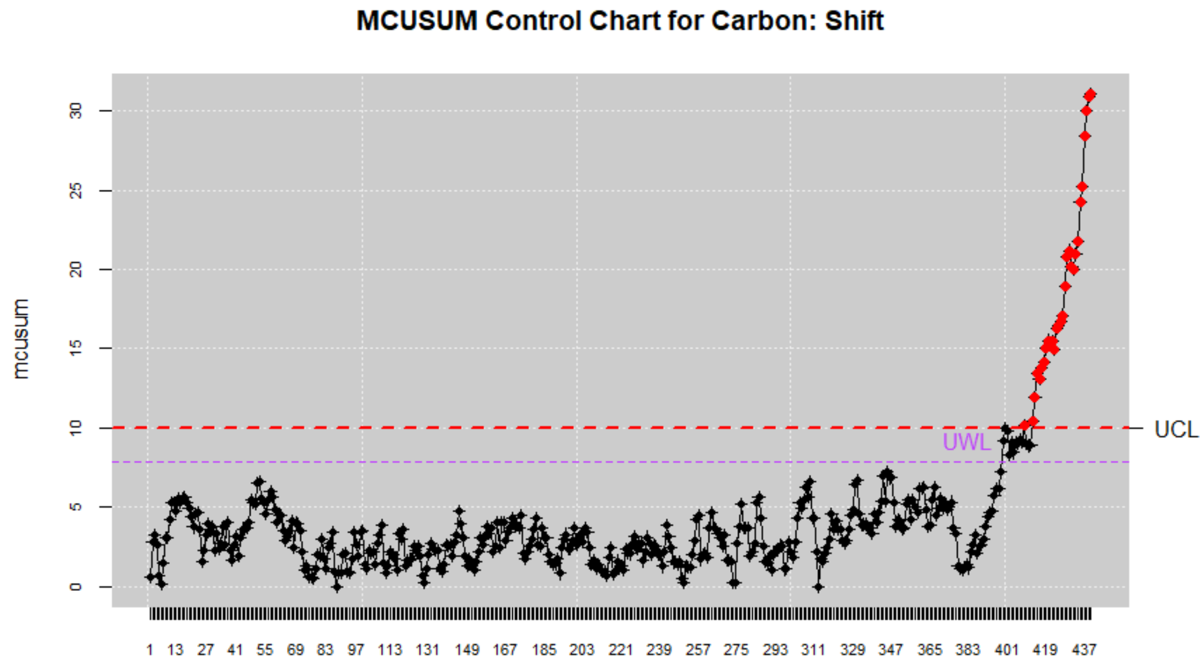

Figure 32: MCUSUM control chart for carbonshift2.xlsx.

At first, the process on both charts remained stable, with most points fluctuating within the control limits, indicating that the process was under control. However, as the sample points increase, there is a noticeable upward trend starting from sample point 321. This upward trend becomes more pronounced as the process continues. In the data monitoring carbonshift.xlsx, although the changes were minimal, MCUSUM still captured the changes very well. Whereas in the carbonshift2.xlsx data due to the sensitivity of MCUSUM to changes in movement, the visualization results are very extreme.

Possible causes of these shifts could include many possibilities, such as changes in raw material properties, equipment malfunction, operator error, or environmental changes that impact the process. To address the shifts indicated by the MCUSUM control chart, it is imperative to conduct a root cause analysis to identify the underlying issues. Corrective measures may involve recalibrating equipment, retraining operators, implementing stricter quality controls on raw materials, or adjusting process parameters to compensate for identified variations. In addition, increasing the frequency of monitoring can help detect such deviations early, thus enabling prompt intervention and minimizing the impact on overall process quality.

4.6 MEWMA Control Charts

4.6.1 MEWMA Control Chart: Normal Case

In this section, visualization is facilitated by using the same method, using qcr package. The code used to visualize the MEWMA control chart using the qcr package is identical to the creation of the MCUSUM control chart using the qcr package (see: section 4.5.1). The main difference lies only in the "# compute MEWMA" programming part and the title of the plot. Here is a brief step-by-step, which only details the differences between this code and the code used in MCUSUM.

1. Install and load the qcr package

2. Preparing data for MEWMA

3. Preparing the data for multivariate quality control

4. Determine initial mean vector (my0)

5. Determine number of dimensions

6. Compute warning limits

7. Compute MEWMA

```
mewma_result <- mqcs.mewma(data_selected, limits = NULL,
                           Xmv = my0,
                           S = NULL,
                           lambda = 0.1, # 0,1-0,9
                           method = "sw")
```

- mewma_result <- mqcs.mewma(...): This calculates the MEWMA statistics for the multivariate data. The parameters used are:
    - data_selected: The prepared multivariate data.
    - limits = NULL: No predefined control limits are used.

- ➢ Xmv = my0: The initial mean vector.
- ➢ S = NULL: The covariance matrix, set to NULL (default).
- ➢ lambda = 0.1: Smoothing parameter, determining the weight given to recent observations.
- ➢ method = "sw": The method used for calculation, in this case, "sw".

8. Summary of MEWMA result

```
summary(mewma_result)
```

- summary(mewma_result): This command provides a summary of the MEWMA results, including statistics and key findings.

9. Plot MEWMA chart

```
plot(mewma_result, title = "MEWMA Control Chart for Carbon")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 400, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 3)
```

- plot(mewma_result, title = "MEWMA Control Chart for Carbon"): Plot the MEWMA control chart for carbon data with “MEWMA Control Chart for Carbon” as its title.

The combination of all these parameters and code in R enables comprehensive calculation and visualization of MEWMA control charts for a given multivariate data set. Using the mqcs.mewma function, the code utilizes the initial mean vector and smoothing parameters to detect shifts in the process mean over time. The resulting MEWMA chart provides valuable insights into the stability and control of multivariate processes (full code is in the appendix).

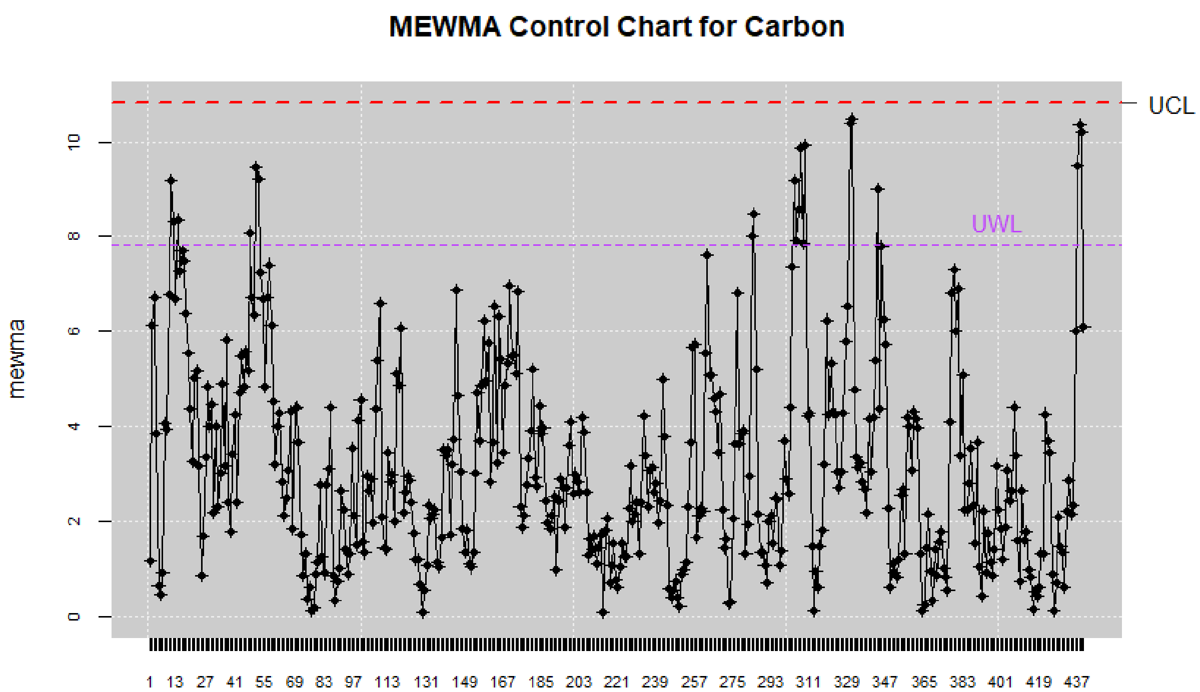

Figure 33: MEWMA control chart for carbon.xlsx.

The MEWMA control chart depicted in the figure represents the monitoring of a multivariate process for carbon data. The chart uses the Multivariate Exponentially Weighted Moving Average (MEWMA) method to detect small shifts in the process mean. The plotted points represent the MEWMA statistics for each sample, which are calculated using a smoothing parameter (λ) that gives more weight to the most recent data while considering the entire history of the data.

In this chart, the UCL, marked with a red dotted line, signifies the limit at which the process is considered to be out of control. The UWL, marked with a purple dotted line, indicates the warning zone where the process might deviate towards an out-of-control condition. The majority of the data points are below the UWL, which indicates that the process is generally under control, although some points occasionally breach the warning limit.

Towards the end of the sample range, there are some points that approach and even exceed the UWL, indicating an increasing trend that may indicate a potential shift in the process. However, there are no points that exceed the UCL, indicating that despite the fluctuations and warnings, the process has not reached a critical out-of-control condition. This pattern may indicate the need for closer monitoring and investigation into possible causes of variation to prevent future UCL violations.

4.6.2 MEWMA Control Chart: Special Case (Jump)

To demonstrate the MEWMA control chart in the jump case, the data to be used is the same data used in the Hotelling's T^2 and MCUSUM session of the jump case, namely the carbonjump.xlsx and carbonjump2.xlsx. The code used to illustrate it is very similar to the code in illustrating in section 4.5.2 (MCUSUM case: Jump), here is the code used, and the explanation will be focused on each of the differences between them.

1. Install and load the qcr package

2. Preparing data for MEWMA

3. Preparing the data for multivariate quality control

```
mewma_jump <- mqcd(datajump[,c(4:6)])
```

- mewma_jump <- mqcd(datajump[,c(4:6)]): In MCUSUM: Jump, the dataframe used was "mcusum_jump", but here is "mewma_jump".

4. Determine initial mean vector (my0)

5. Determine number of dimensions

6. Compute warning limits

7. Compute MEWMA

```
mewma_result <- mqcs.mewma(mewma_jump, limits = NULL,
                           Xmv = my0,
                           S = NULL,
                           lambda = 0.1, # 0,1-0,9
                           method = "sw")
```

- mewma_result <- mqcs.mewma(mewma_jump, ...): This calculates the MEWMA statistics for the multivariate data from mewma_jump dataframe.

8. Summary of MEWMA result

```
summary(mewma_result)
```

- summary(mewma_result): This command provides a summary of the MEWMA results (includes statistics and findings).

9. Plot MEWMA chart

```
plot(mewma_result, title = "MEWMA Control Chart for Carbon: Jump")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 400, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 3)
```

- plot(mewma_result, title = "MEWMA Control Chart for Carbon: Jump"): Plot the control chart for with “MEWMA Control Chart for Carbon: Jump” as its title.
- Full code is in the appendix.

The main difference between the code used here and the code used in section 4.5.2 (MCUSUM: Jump) is only in the data used and the chart title at the plotting stage.

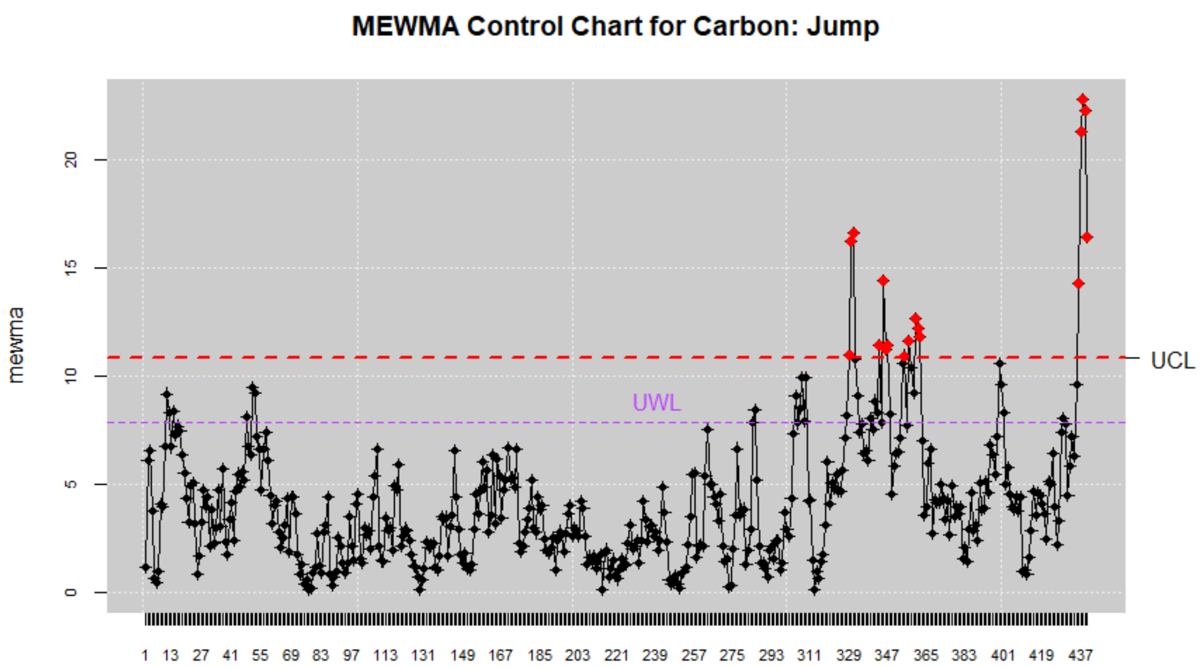

Figure 34: MEWMA control chart for carbonjump.xlsx.

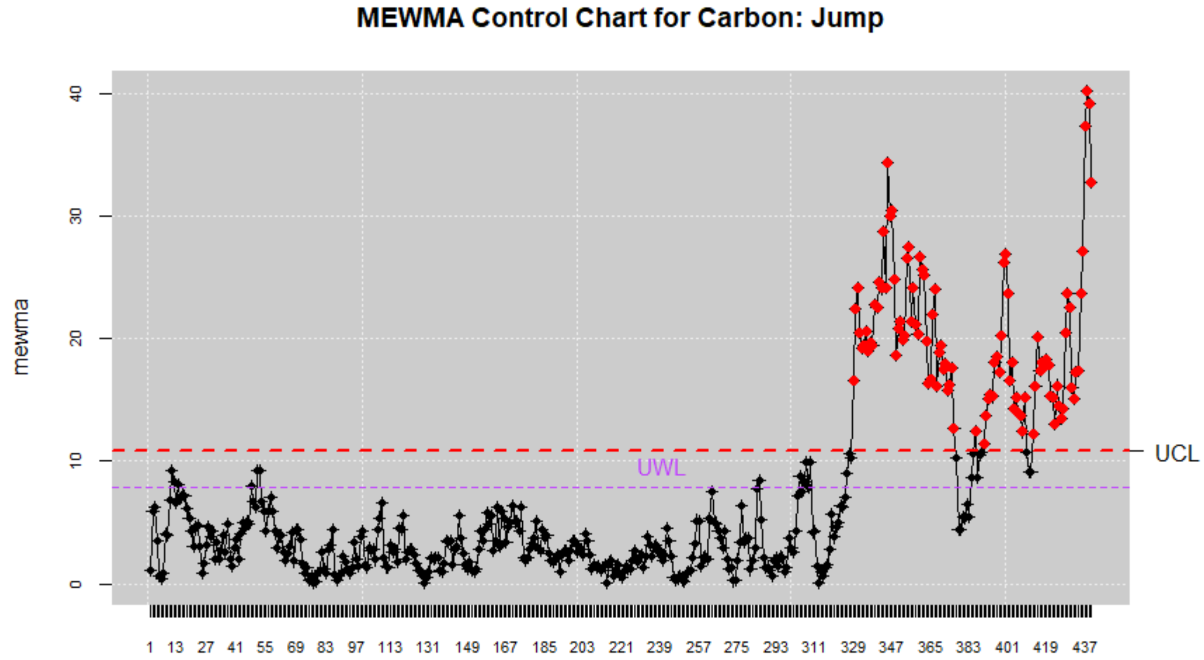

Figure 35: MEWMA control chart for carbonjump2.xlsx.

The MEWMA (Multivariate Exponentially Weighted Moving Average) control chart illustrates the behavior of the statistics over time. Initially, for samples 1 to about 320, the process in both charts remains stable and under control, with the MEWMA statistics fluctuating below the Upper Warning Limit (UWL). This indicates that the process is functioning within acceptable limits without any significant change in motion in the mean vector.

However, after sample 320, changes began to appear, as evidenced by the MEWMA statistic approaching and then exceeding the Upper Control Limit (UCL). This significant upward trend indicates a jump in both process charts under observation, indicating that the process is no longer under control. The red dots outside the UCL highlight out-of-control points, indicating potential problems that require investigation. However, it can be seen that MEWMA manages to detect small and moderate changes well.

Jumps can be caused by several factors such as the influence of the quality of the materials used, changes in the manufacturing process, equipment malfunction, or even human error. To solve this problem, corrective actions should be taken as soon as possible, such as checking whether equipment is functioning as planned, retraining personnel, checking raw materials and etc. that are directly related to the process. In addition, continuously monitoring the process using a control chart can prevent future deviations by enabling quick detection and intervention.

4.6.3 MEWMA Control Chart: Special Case (Shift)

To illustrate the MEWMA control chart in the shift scenario, the data used is the same as the data used for “shift” case, the specifically modified carbon data saved as “carbonshift” and “carbonshift2”. The code used also very similar to the code used in section MCUSUM case: Jump. Certain details explanation of the changes will also be explained.

1. Install and load the qcr package

2. Preparing data for MEWMA

3. Preparing the data for multivariate quality control

```
mewma_shift <- mqcd(datashift[,c(4:6)])
```

- mewma_shift <- mqcd(datashift[,c(4:6)]): Prepares the datashift for multivariate quality control analysis using the mqcd function from the qcr package into a new dataframe, “mewma_shift”.

4. Determine initial mean vector (my0)

5. Determine number of dimensions

6. Compute warning limits

7. Compute MEWMA

```
mewma_result <- mqcs.mewma(mewma_shift, limits = NULL,
                           Xmv = my0,
                           S = NULL,
                           lambda = 0.1, # 0,1-0,9
                           method = "sw")
```

- mewma_result <- mqcs.mewma(mewma_shift, ...): Dataframe used for the calculating of MEWMA statistics is mewma_shift dataframe.

8. Summary of MEWMA result

```
summary(mewma_result)
```

- summary(mewma_result): To provide summary of MEWMA result including statistics and findings.

9. Plot MEWMA chart

```
plot(mewma_result, title = "MEWMA Control Chart for Carbon: Shift")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 437, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 1)
```

- plot(mewma_result, title = "MEWMA Control Chart for Carbon: Shift"): Plotting a MEWMA control chart and naming it as “MEWMA Control Chart for Carbon: Shift)..

As in chapter 4.6.2 MEWMA (Jump) with 4.5.2 MCUSUM (Jump), the code usage comparison between this chapter MEWMA (Shift) and chapter 4.6.2 MCUSUM (Shift) is very similar, the difference lies only in the data used (here is carbonshift and carbonshift2) and the title on the control chart (the full code is in the appendix).

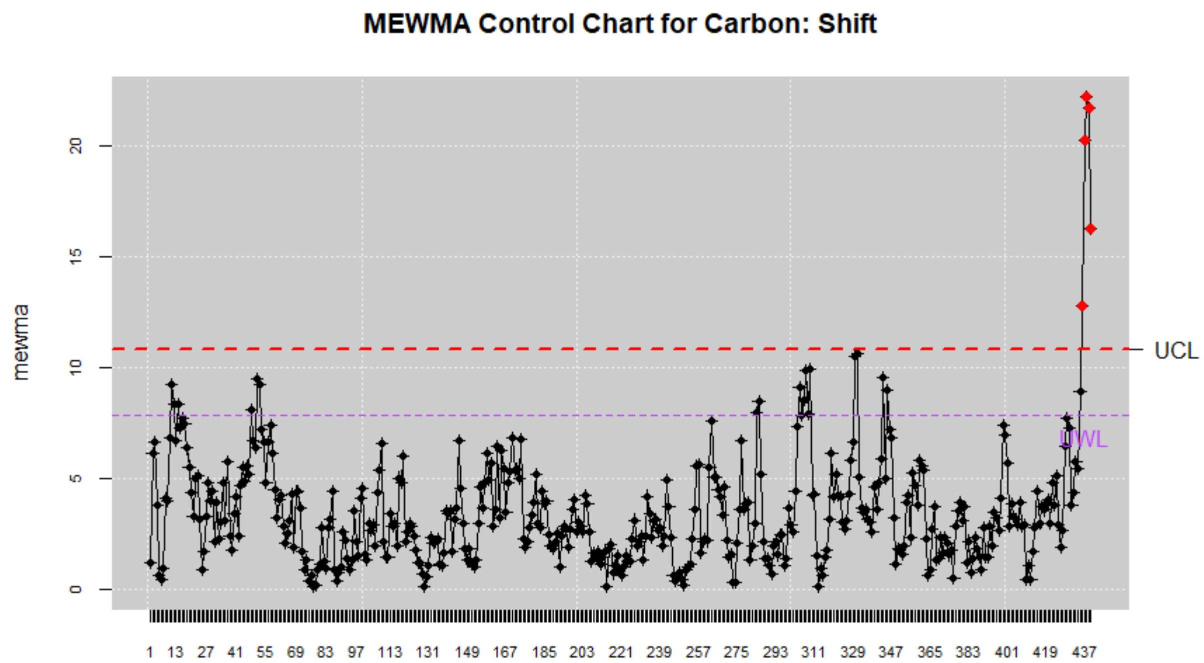

Figure 36: MEWMA control chart for carbonshift.xlsx.

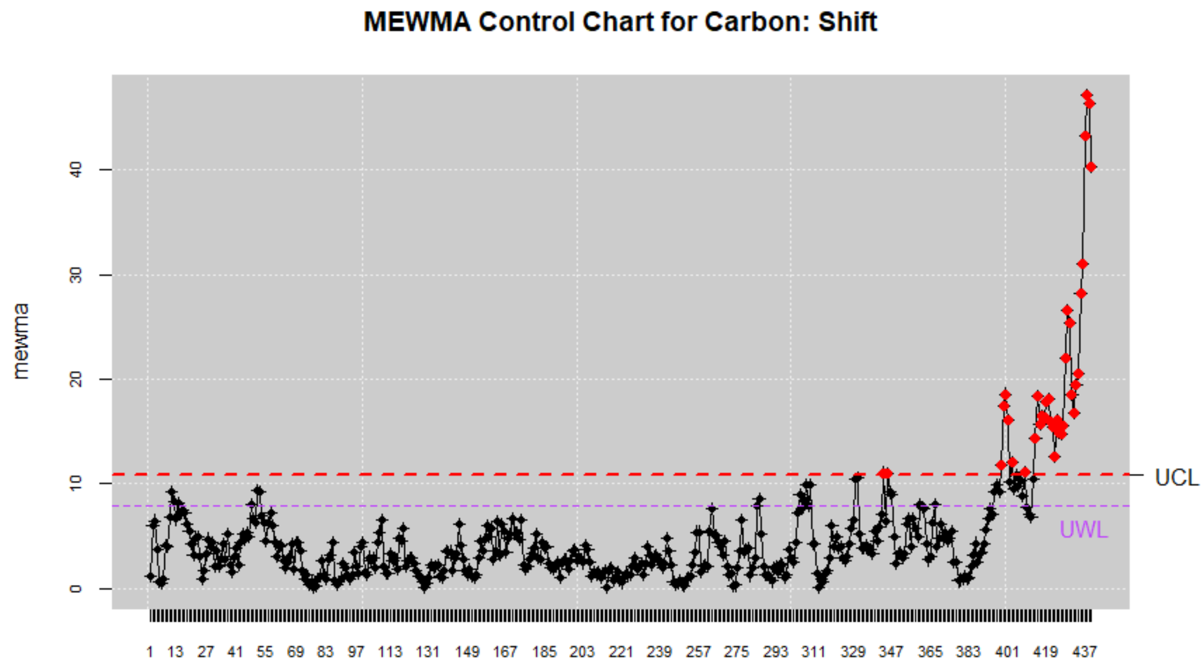

Figure 37: MEWMA carbon chart for carbonshift2.xlsx.

The MEWMA control charts for both charts, illustrate significant deviations from the planned process behavior. Initially, the process remains inside the control zone, indicating the process is still under control. However, a noticeable shift begins around sample 320, with the MEWMA value consistently increasing and eventually exceeding the Upper Control Limit (UCL). It is shown again that MEWMA control charts can detect small and moderate changes very well, whereas MCUSUM is only sensitive to small changes and Hotelling's T2 is best for large changes.

In real-life practice, such shifts may occur due to various reasons. Changes in raw material quality, variations in environmental conditions, equipment wear or damage, and human error are common causes. For example, suppliers may change the composition of raw materials, leading to characteristic changes in the final product. Similarly, maintenance issues can cause equipment to operate outside of its optimal parameters, thus affecting the process yield.

To address this issue, a systematic approach is required. First, conduct a thorough root cause analysis to identify the specific factors causing the shift. This may include inspecting equipment, reviewing process changes, analyzing supplier data, and interviewing operators. Once the root cause is identified, implement corrective actions such as recalibrating equipment, adjusting process parameters, training personnel, or changing suppliers.

In addition, improving process monitoring through the use of control charts such as MEWMA can aid in the early detection of future shifts, allowing for prompt intervention before the process gets out of control. Continuous improvement practices and regular audits can also ensure that processes remain stable and within control limits.

5 Summary and Conclusion

T^2 Hotelling’s control charts are the most effective for detecting large and sudden shifts in process averages need to be detected quickly, such as in certain manufacturing processes where equipment breakdowns can cause significant deviations. This method assumes a known and constant variance-covariance matrix, so it lacks flexibility in situations where these parameters may change over time.

MCUSUM control chart is perfect for industries where maintaining consistent quality with minimal variation is critical, due to its sensitivity to small and medium shifts. It aggregates information over time using the cumulative sum (CUSUM) statistic, which allows the detection of shifts that may be missed by methods that focus on single observations. Unlike Hotelling's T^2, MCUSUM charts are less affected by the correlation structure in the data, thus providing a strong alternative for processes with correlated variables. However, MCUSUM control charts are generally less effective for detecting large shifts or large jumps compared to detecting small to moderate shifts.

MEWMA control charts, with their ability to effectively detect gradual changes, are advantageous in situations where trends over time are critical to monitor, such as in environmental monitoring or service quality assessment. These charts use weighted averages of past and current observations, with greater emphasis on recent data. This characteristic allows MEWMA to handle trends and gradual changes more effectively than MCUSUM and Hotelling's T^2 graphs.

The selection of the control chart method should be guided by the specific characteristics and requirements of the process under investigation. Factors such as the stability of the variance-covariance matrix, the correlation structure of the data, and the nature of the shifts to be detected should be carefully considered. This customized approach ensures that the most appropriate control chart is applied, thereby improving the reliability and effectiveness of process monitoring and quality control.

In conclusion, Hotelling's T^2, MCUSUM, and MEWMA control charts each offer distinct advantages and fulfill different aspects of process monitoring. By understanding the unique strengths and limitations of each method, users can make informed decisions to optimize quality control efforts.

6 Outlook

For the future advancement and practical application of multivariate control charts, the incorporation of Shiny, an R package for building interactive web applications, is very promising. Shiny allows users to create dynamic and easy-to-use interfaces that can significantly improve the utility and accessibility of multivariate control charts such as Hotelling's T^2, MCUSUM, and MEWMA. By integrating Shiny, static data visualizations can be transformed into interactive tools that provide deeper insights and greater flexibility for practitioners in quality control and process monitoring.

First, the Shiny app can facilitate real-time monitoring and analysis of process data. Users can upload new data sets, adjust parameters, and instantly see the impact on control charts. This capability is especially valuable for industries where process parameters change frequently, and timely detection of shifts is critical. For example, in a manufacturing environment, the Shiny app can allow engineers to continuously input new production data and visualize the corresponding control charts on the fly, ensuring that any deviations from the expected process average can be quickly identified and addressed.

Secondly, the interactive nature of Shiny can enhance the process of exploratory data analysis. Users can interact with the visualizations to zoom in on specific time periods, filter data based on specific criteria, and compare different segments of the process. This can be particularly useful in identifying patterns or trends that may not be immediately apparent in static graphs. For example, with the Shiny app, one can investigate the performance of different batches in a production process, isolating the effects of different variables and uncovering information that can lead to process improvements.

In addition, Shiny allows customization of control charts to better meet the needs of specific applications. Users can modify chart settings, such as control limits and weighting factors, to see how different configurations affect the detection of process mean shifts. This flexibility enables a more customized approach to process monitoring, which accommodates the unique characteristics of different processes. For example, in a production line where certain variables are more important than others, the Shiny app can be used to prioritize these variables in the control chart analysis, ensuring that any significant deviations can be highlighted quickly.

In conclusion, the integration of Shiny into multivariate control chart visualization has great potential to improve the effectiveness of quality control processes. By providing real-time, interactive, and customizable tools, Shiny can help practitioners to better monitor and respond to process changes, which will ultimately result in a more robust and efficient production system.

List of Figures

List of Tables

References

[1] Cloud Software Group, Inc.-2024 – What is statistical process control?:

https://www.spotfire.com/glossary/what-is-statistical-process-control#:~:text=Statistical%20process%20control%20(SPC)%20monitors,monitored%20to%20control%20that%20process.

(retrieved on 26.05.2024).

[2] Babu, Vikash-2023 – The Role of Statistical Process Control (SPC) in Quality Management for Polymers:

https://www.linkedin.com/pulse/role-statistical-process-control-spc-quality-management-vikash-babu/

(retrieved on 26.05.2024).

[3] Hillier, Will-2023 – What's the Difference Between Descriptive and Inferential Statistics?:

https://careerfoundry.com/en/blog/data-analytics/inferential-vs-descriptive-statistics/

(retrieved on 26.05.2024).

[4] Universität München – Deskriptive Statistik

https://www.statistik.uni-muenchen.de/formulare/skripte_u_aehnliches/deskriptivestatistik_groll.pdf

(retrieved on 26.05.2024)

[5] Nick, T. G. (2007): Descriptive Statistics. Methods in Molecular Biology™, pp. 47–48.

[6] O'Connor, J. J., Robertson, E. F., University of St Andrews-2003 – Karl Pearson:

https://mathshistory.st-andrews.ac.uk/Biographies/Pearson/

(retrieved on 26.05.2024).

[7] Guo, Anyi-2021 – Pearson vs. Spearman Correlation: What's the difference?

https://anyi-guo.medium.com/correlation-pearson-vs-spearman-c15e581c12ce

(retrieved on 26.04.2024)

[8] University of York-2005 – Charles Spearman:

https://www.york.ac.uk/depts/maths/histstat/spearman_biog.htm

(retrieved on 26.05.2024)

[9] Karthik, RPubs-2017 – Multivariate Descriptive Statistics

https://rpubs.com/bskc/289145

(retrieved on 26.05.2024)

[10] Bagchi, T. P.-2000 – Statistical Process Control:

https://www.researchgate.net/publication/263559764_Statistical_Process_Control

(retrieved on 26.05.2024).

[11] Liebscher, E. (2022), 6. Statistische Qualitätskontrolle in der Prozessüberwachung, StatProzesskontrolle1_v.pptx [Slide: 6]. Presented during lecture: Zuverlässigkeit und Qualitätssicherung.

[12] Eberly College of Science Department of Statistics – Lesson 4: Multivariate Normal Distributions:

https://online.stat.psu.edu/stat505/book/export/html/636

(retrieved on 26.05.2024)

[13] Qiu, P. (2014). Introduction to Statistical Process Control, pp. 6-9. CRC Press.

[14] Montgomery, D. C. (2009). Introduction to Statistical Quality Control. John Wiley & Sons.

[15] ASQ – Control Chart:

https://asq.org/quality-resources/control-chart

(retrieved on 27.05.2024).

[16] Best, M., Neuhauser, D., National Library of Medicine-2006 – Walter A Shewhart, 1942, and the Hawthorne factory:

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2464836/ (retrieved on 27.05.2024).

(retrieved on 27.05.2024).

[17] Qiu, P. (2014). Introduction to Statistical Process Control, pp. 73-112. CRC Press.

[18] Castagliola, P., Figueiredo, F. O., Maravelakis, P. E. (2017). The CUSUM Median Chart for known and Estimated Parameters, 17(1), pp. 345-370. REVSTAT – Statistical Journal.

[19] A. F. Bissell (1969). Cusum Techniques for Quality Control. Journal of the Royal Statistical Society. Series C (Applied Statistics), 18(1), pp. 1–30.

[20] Liebscher, E. (2022), 6. Statistische Qualitätskontrolle in der Prozessüberwachung, 6.5 Die CUSUM-Karte, StatProzesskontrolle3_v.pptx [Slide: 1-3]. Presented during lecture: Zuverlässigkeit und Qualitätssicherung.

[21] Montana State University – 9.7 Exponentially Weighted Moving Average Control Charts:

https://math.montana.edu/jobo/st528/documents/chap9d.pdf

(retrieved on 27.05.2024).

[22] StatPoint Technologies, Inc.-2009 – EWMA Charts:

https://cdn2.hubspot.net/hubfs/402067/PDFs/EWMA_Charts.pdf

(retrieved on 27.05.2024).

[23] Liebscher, E. (2022), 6. Statistische Qualitätskontrolle in der Prozessüberwachung, 6.6 Die EWMA-Karte, StatProzesskontrolle3_v.pptx [Slide: 11-14]. Presented during lecture: Zuverlässigkeit und Qualitätssicherung.

[24] SAGE Publications, Inc.-2016 – Hotelling's T^2 A Two-Group Multivariate Analysis:

https://us.sagepub.com/sites/default/files/upm-assets/70364_book_item_70364.pdf

(retrieved on 27.05.2024).

[25] Kahle, W., Liebscher, E. (2013): Zuverlässigleitsanalyse und Qualitätssicherung. Oldenbourg Verlag München, pp. 266-269.

[26] Liebscher, E. (2024). multivcharts.pdf [unpublished manuscript]. Provided by Eckhard Liebscher.

[27] Lowry, C. A., et al. (1992). A Multivariate Exponentially Weighted Moving Average Control Chart. Technometrics, 34(1), pp. 46–53.

[28] StatSoft-2024 – The Statistica Story:

https://www.statsoft.de/en/the-statistica-story/#:~:text=We%2C%20the%20company%20StatSoft%20and,the%20name%20STATISTICA%20in%201991.

(retrieved on 27.05.2024).

[29] TIBCO-2018 – Manufacturing Intelligence in the Age of Industry 4.0 and the IoT:

https://www.spotfire.com/content/dam/spotfire/documents/solution-brief/manufacturing-intelligence-in-the-age-of-industry-40-and-the-iot.pdf

(retrieved on 27.05.2024).

[30] The R Foundation – What is R?:

https://www.r-project.org/about.html

(retrieved on 27.05.2024).

[31] Becker, R. A. – S (Programming Language):

https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W120512S.pdf

(retrieved on 27.05.2024).

[32] The Knowledge Academy – Advantages and Disadvantages of R Programming Language:

https://www.theknowledgeacademy.com/blog/advantages-and-disadvantages-of-r-programming/

(retrieved on 27.05.2024).

Appendix

Code: Hotelling's T^2 control chart: Normal case

```
# Title: Plotting Hotelling's Control Chart (Case: Normal)

# Load the necessary data
library(readxl)
datacarbon <- read_excel("C:/Users/Data/carbon.xlsx")

# Calculate Hotelling's T^2 statistic
n <- nrow(datacarbon)
dim<- 3

# Calculate the initial mean vector (my0)
apply(as.matrix(datacarbon[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)

# Define the Hotelling's T^2 statistic function
hotelling_stat <- function(x,cov_matrix,sample_size) {
  sample_size*(as.numeric(x) - my0) %*% solve(cov_matrix) %*%
(as.numeric(x) - my0)
}

# Calculate the covariance matrix and mean vector
cov_matrix <- cov(datacarbon[1:320,1:3])
mean_vector <- colMeans(datacarbon)

# Set specific cutoff for Hotelling's T^2
alpha<- 0.0027
cutoff <- qchisq(1-alpha,dim)     # Set desired cutoff value
UWL<- qchisq(0.95,dim)

# Set the y-axis range to 30
y_range <- 30

# Calculate Hotelling's T^2 statistics for subgroups
sample_size<- 10
m<- n %/% sample_size
meanv<- matrix(ncol=dim,nrow=m)
hotelling<- vector(length=m)
for (i in 1:m){
  i1<-(i-1)*sample_size+1
  i2<-i*sample_size
  meanv[i,]<- apply(datacarbon[i1:i2,1:3],2,mean)
  hotelling[i]<- hotelling_stat(meanv[i,],cov_matrix,sample_size)
}

# Plotting Hotelling's T^2 Chart
plot(hotelling, type = "b", pch = 16, col = ifelse(hotelling > cutoff,
"red", "green"),
     main = "Hotelling's T^2 Control Chart for Carbon", ylim = c(0,
y_range))
abline(h = cutoff, col = "red", lty = 2)
abline(h = UWL, col = "purple", lty = 2)
text(x = max(hotelling), y = cutoff, labels = "UCL", col = "red", cex =
1, pos = 3)
text(x = max(hotelling), y = UWL, labels = "UWL", col = "purple", cex =
1, pos = 3)
```

Code: Modifying an excel data: carbonjump.xlsx

```
# Title: Modifying an Excel Data (carbonjump.xlsx)

# Load required library and read data
library(readxl)
dataxl <- read_excel("C:/Users/Data/carbon.xlsx")

# Create and modify a new column "inner2"
dataxl$inner2 <- dataxl$inner #Create a new column in column 'd' with a
copy of the data from column 'inner'
dataxl$inner2[322:nrow(dataxl)] <- dataxl$inner2[322:nrow(dataxl)] + 0.02
#Add a value of 0.02 to each entry in column 'd' after row 321.

# Create and modify a new column "thickness2"
dataxl$thickness2 <- dataxl$thickness
dataxl$thickness2[322:nrow(dataxl)] <-
dataxl$thickness2[322:nrow(dataxl)] + 0.04

# Create and modify a new column "length2"
dataxl$length2 <- dataxl$length
dataxl$length2[322:nrow(dataxl)] <- dataxl$length2[322:nrow(dataxl)] +
0.06

# Save the modified data to a new Excel file
library(writexl)
write_xlsx(dataxl, "carbonjump.xlsx")
```

Code: Modifying an excel data: carbonjump2.xlsx

```
# Title: Modifying an Excel Data (carbonjump2.xlsx)

# Load required library and read data
library(readxl)
dataxl <- read_excel("C:/Users/Data/carbon.xlsx")

# Create and modify a new column "inner2"
dataxl$inner2 <- dataxl$inner
dataxl$inner2[322:nrow(dataxl)] <- dataxl$inner2[322:nrow(dataxl)] + 0.05

# Create and modify a new column "thickness2"
dataxl$thickness2 <- dataxl$thickness
dataxl$thickness2[322:nrow(dataxl)] <-
dataxl$thickness2[322:nrow(dataxl)] + 0.07

# Create and modify a new column "length2"
dataxl$length2 <- dataxl$length
dataxl$length2[322:nrow(dataxl)] <- dataxl$length2[322:nrow(dataxl)] +
0.09

# Save the modified data to a new Excel file
library(writexl)
write_xlsx(dataxl, "carbonjump2.xlsx")
```

Code: Hotelling's T^2 control chart: "Jump" case

```
# Title: Plotting Hotelling's Control Chart (Case: Jump)

# Load the necessary data
library(readxl)
datajump <- read_excel("C:/Users/Data/carbonjump.xlsx") # or
datajump <- read_excel("C:/Users/Data/carbonjump2.xlsx")

# Calculate Hotelling's T^2 statistic
n <- nrow(datajump)
dim<- 3

# Calculate the initial mean vector (my0)
apply(as.matrix(datajump[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)

# Define the Hotelling's T^2 statistic function
hotelling_stat <- function(x,cov_matrix,sample_size) {
  sample_size*(as.numeric(x) - my0) %*% solve(cov_matrix) %*%
(as.numeric(x) - my0)
}

# Calculate the covariance matrix and mean vector
cov_matrix <- cov(datajump[1:320,4:6])
mean_vector <- colMeans(datajump)

# Set specific cutoff for Hotelling's T^2
alpha<- 0.0027
cutoff <- qchisq(1-alpha,dim)     #23.030   # Set desired cutoff value
UWL<- qchisq(0.95,dim)

# Set the y-axis range to 30
y_range <- 30

# Calculate Hotelling's T^2 statistics for subgroups
sample_size<- 10
m<- n %/% sample_size
meanv<- matrix(ncol=dim,nrow=m)
hotelling<- vector(length=m)
for (i in 1:m){
  i1<-(i-1)*sample_size+1
  i2<-i*sample_size
  meanv[i,]<- apply(datajump[i1:i2,4:6],2,mean)
  hotelling[i]<- hotelling_stat(meanv[i,],cov_matrix,sample_size)
}

# Plotting Hotelling's T^2 Chart
plot(hotelling, type = "b", pch = 16, col = ifelse(hotelling > cutoff,
"red", "green"),
     main = "Hotelling's T^2 Control Chart for Carbon: Jump", ylim = c(0,
y_range))
abline(h = cutoff, col = "red", lty = 2)
abline(h = UWL, col = "purple", lty = 2)
text(x = max(hotelling), y = cutoff, labels = "UCL", col = "red", cex =
1, pos = 3)
text(x = max(hotelling), y = UWL, labels = "UWL", col = "purple", cex =
1, pos = 3)
```

Code: Modifying an excel data: carbonshift.xlsx

```
# Title: Modifying an Excel Data (carbonshift.xlsx)

# Load required library and read data
library(readxl)
dataxl <- read_excel("C:/Users/Data/carbon.xlsx")

# Create and modify a new column "inner3"
dataxl$inner3 <- dataxl$inner #Create a new column in column 'd' with a
copy of the data from column 'inner'
dataxl$inner3[321:440] <- dataxl$inner3[321:440] + (0.0002 * (1:120)) #
Add 0.0002 to the value of every entry in column D after row 250

# Create and modify a new column "thickness3"
dataxl$thickness3 <- dataxl$thickness
dataxl$thickness3[321:440] <- dataxl$thickness3[321:440] + (0.0003 *
(1:120))

# Create and modify a new column "length3"
dataxl$length3 <- dataxl$length
dataxl$length3[321:440] <- dataxl$length3[321:440] + (0.0004 * (1:120))

# Save the modified data to a new Excel file
library(writexl)
write_xlsx(dataxl, "carbonshift.xlsx")
```

Code: Modifying an excel data: carbonshift.xlsx

```
# Title: Modifying an Excel Data (carbonshift2.xlsx)

# Load required library and read data
library(readxl)
dataxl <- read_excel("C:/Users/Data/carbon.xlsx")

# Create and modify a new column "inner3"
dataxl$inner3 <- dataxl$inner #Create a new column in column 'd' with a
copy of the data from column 'inner'
dataxl$inner3[321:440] <- dataxl$inner3[321:440] + (0.0005 * (1:120)) #
Add 0.0005 to the value of every entry in column D after row 250

# Create and modify a new column "thickness3"
dataxl$thickness3 <- dataxl$thickness
dataxl$thickness3[321:440] <- dataxl$thickness3[321:440] + (0.0006 *
(1:120))

# Create and modify a new column "length3"
dataxl$length3 <- dataxl$length
dataxl$length3[321:440] <- dataxl$length3[321:440] + (0.0007 * (1:120))

# Save the modified data to a new Excel file
library(writexl)
write_xlsx(dataxl, "carbonshift2.xlsx")
```

Code: Hotelling's T2 control chart: "Shift" case

```
# Title: Plotting Hotelling's Control Chart (Case: Shift)

# Load the necessary data
library(readxl)
datashift <- read_excel("C:/Users/Data/carbonshift.xlsx") # or
datashift <- read_excel("C:/Users/Data/carbonshift2.xlsx")

# Calculate Hotelling's T^2 statistic
n <- nrow(datashift)
dim<- 3

# Calculate the initial mean vector (my0)
apply(as.matrix(datashift[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)

# Define the Hotelling's T^2 statistic function
hotelling_stat <- function(x,cov_matrix,sample_size) {
  sample_size*(as.numeric(x) - my0) %*% solve(cov_matrix) %*%
(as.numeric(x) - my0)
}

# Calculate the covariance matrix and mean vector
cov_matrix <- cov(datashift[1:320,4:6])
mean_vector <- colMeans(datashift)

# Set specific cutoff for Hotelling's T^2
alpha<- 0.0027
cutoff <- qchisq(1-alpha,dim)    #23.030   # Set desired cutoff value
UWL<- qchisq(0.95,dim)

# Set the y-axis range to 30
y_range <- 35

# Calculate Hotelling's T^2 statistics for subgroups
sample_size<- 10
m<- n %/% sample_size
meanv<- matrix(ncol=dim,nrow=m)
hotelling<- vector(length=m)
for (i in 1:m){
  i1<-(i-1)*sample_size+1
  i2<-i*sample_size
  meanv[i,]<- apply(datashift[i1:i2,4:6],2,mean)
  hotelling[i]<- hotelling_stat(meanv[i,],cov_matrix,sample_size)
}

# Plotting Hotelling's T^2 Chart
plot(hotelling, type = "b", pch = 16, col = ifelse(hotelling > cutoff,
"red", "green"),
     main = "Hotelling's T^2 Control Chart for Carbon: Shift", ylim =
c(0, y_range))
abline(h = cutoff, col = "red", lty = 2)
abline(h = UWL, col = "purple", lty = 2)
text(x = max(hotelling), y = cutoff, labels = "UCL", col = "red", cex =
1, pos = 3)
text(x = max(hotelling), y = UWL, labels = "UWL", col = "purple", cex =
1, pos = 3)
```

Code: MCUSUM control chart: Normal case

```
# Title: Plotting MCUSUM Control Chart using qcr Package (Case: Normal)

# Install and load the qcr package
install.packages("qcr")
library(qcr)

# Preparing data for mcusum
library(readxl)
datacarbon <- read_excel("C:/Users/Data/carbon.xlsx")

# Preparing the data for multivariate control quality
data_selected <- mqcd(datacarbon)

# Determine my0
apply(as.matrix(datacarbon[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)

# Determine how many dimensions
dim<- 3

# Compute warning and control limits
UWL<- qchisq(0.95,dim)

# Compute mcusum
mcusum_result <- mqcs.mcusum(data_selected, limits = NULL,
                             Xmv = my0,
                             S = NULL,
                             k = 0.5,
                             h = 10,
                             method = "sw")

# Summary of mcusum result
summary(mcusum_result)

# Plot mcusum chart
plot(mcusum_result, title = "MCUSUM Control Chart for Carbon")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 400, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 3)
```

Code: MCUSUM control chart: “Jump” case

```
# Title: Plotting MCUSUM Control Chart using qcr Package (Case: Jump)

# Install and load the qcr package
install.packages("qcr")
library(qcr)

# Preparing data for MCUSUM
library(readxl)
datajump <- read_excel("C:/Users/Data/carbonjump.xlsx") # or
datajump <- read_excel("C:/Users/Data/carbonjump2.xlsx")

# Preparing the data for multivariate control quality
mcusum_jump <- mqcd(datajump[,c(4:6)])

# Determine my0
apply(as.matrix(datajump[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)

# Determine how many dimensions
dim<- 3

# Compute warning and control limits
UWL<- qchisq(0.95,dim)

# Compute MCUSUM
mcusum_result <- mqcs.mcusum(mcusum_jump, limits = NULL,
                             Xmv = my0,
                             S = NULL,
                             k = 0.5,
                             h = 10,
                             method = "sw")

# Summary of MCUSUM result
summary(mcusum_result)

# Plot MCUSUM chart
plot(mcusum_result, title = "MCUSUM Control Chart for Carbon: Jump")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 400, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 3)
```

Code: MCUSUM control chart: “Shift” case

```
# Title: Plotting MCUSUM Control Chart using qcr Package (Case: Shift)

# Install and load the qcr package
install.packages("qcr")
library(qcr)

# Preparing data for MCUSUM
library(readxl)
datashift <- read_excel("C:/Users/Data/carbonshift.xlsx") # or
datashift <- read_excel("C:/Users/Data/carbonshift2.xlsx")

# Preparing the data for multivariate control quality
mcusum_shift <- mqcd(datashift[,c(4:6)])

# Determine my0
apply(as.matrix(datashift[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)

# Determine how many dimensions
dim<- 3

# Compute warning and control limits
UWL<- qchisq(0.95,dim)

# Compute MCUSUM
mcusum_result <- mqcs.mcusum(mcusum_shift, limits = NULL,
                             Xmv = my0,
                             S = NULL,
                             k = 0.5,
                             h = 10,
                             method = "sw")

# Summary of MCUSUM result
summary(mcusum_result)

# Plot MCUSUM chart
plot(mcusum_result, title = "MCUSUM Control Chart for Carbon: Shift")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 383, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 3)
```

Code: MEWMA control chart: Normal case

```
# Title: Plotting MEWMA Control Chart using qcr Package (Case: Normal)

# Install and load the qcr package
install.packages("qcr")
library(qcr)

# Preparing data for mewma
library(readxl)
datacarbon <- read_excel("C:/Users/Data/carbon.xlsx")

# Preparing the data for multivariate control quality
data_selected <- mqcd(datacarbon)

# Determine my0
apply(as.matrix(datacarbon[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)

# Determine how many dimensions
dim<- 3

# Compute warning and control limits
UWL<- qchisq(0.95,dim)

# Compute MEWMA
mewma_result <- mqcs.mewma(data_selected, limits = NULL,
                           Xmv = my0,
                           S = NULL,
                           lambda = 0.1, # 0,1-0,9
                           method = "sw")

# Summary of MEWMA result
summary(mewma_result)

# Plot MEWMA chart
plot(mewma_result, title = "MEWMA Control Chart for Carbon")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 400, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 3)
```

Code: MEWMA control chart: “Jump” case

```
# Title: Plotting MEWMA Control Chart using qcr Package (Case: Jump)

# Install and load the qcr package
install.packages("qcr")
library(qcr)

# Preparing the data for MEWMA
library(readxl)
datajump <- read_excel("C:/Users/Data/carbonjump.xlsx") # or
datajump <- read_excel("C:/Users/Data/carbonjump2.xlsx")

# Preparing the data for multivariate control quality
mewma_jump <- mqcd(datajump[,c(4:6)])

# Determine my0
apply(as.matrix(datajump[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)

# Determine how many dimensions
dim<- 3

# Compute warning and control limits
UWL<- qchisq(0.95,dim)

# Compute MEWMA
mewma_result <- mqcs.mewma(mewma_jump, limits = NULL,
                           Xmv = my0,
                           S = NULL,
                           lambda = 0.1, # 0,1-0,9
                           method = "sw")

# Summary of MEWMA result
summary(mewma_result)

# Plot MEWMA chart
plot(mewma_result, title = "MEWMA Control Chart for Carbon: Jump")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 240, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 3)
```

Code: MEWMA control chart: “Shift” case

```
# Title: Plotting MEWMA Control Chart using qcr Package (Case: Shift)

# Install and load the qcr package
install.packages("qcr")
library(qcr)

# Preparing the data for MEWMA
library(readxl)
datashift <- read_excel("C:/Users/Data/carbonshift.xlsx") # or
datashift <- read_excel("C:/Users/Data/carbonshift2.xlsx")

# Preparing the data for multivariate control quality
mewma_shift <- mqcd(datashift[,c(4:6)])

# Determine my0
apply(as.matrix(datashift[1:320,1:3]),2,mean)
my0<- c(1.0,1.045,50)

# Determine how many dimensions
dim<- 3

# Compute warning and control limits
UWL<- qchisq(0.95,dim)

# Compute MEWMA
mewma_result <- mqcs.mewma(mewma_shift, limits = NULL,
                           Xmv = my0,
                           S = NULL,
                           lambda = 0.1, # 0,1-0,9
                           method = "sw")

# Summary of MEWMA result
summary(mewma_result)

# Plot MEWMA chart
plot(mewma_result, title = "MEWMA Control Chart for Carbon: Shift")
abline(h = UWL, col = "darkorchid1", lty = 2)
text(x = 437, y = UWL, labels = "UWL", col = "darkorchid1", cex = 1, pos
= 1)
```

carbon.xlsx

inner	thickness	length
0,99	1,09	50,17
1,09	0,9	50,41
1,06	1,18	50,4
1	0,99	49,78
0,91	0,97	49,65
1,02	1,12	50,11
0,96	1,16	50,07
0,98	0,98	50,52
1,05	1,26	50,15
1,01	0,93	50,49
1	1	50,33
0,97	1,17	49,99
0,94	0,89	49,83
0,96	0,95	50,13
0,98	0,96	49,96
0,97	0,92	50,01
0,99	0,98	50,05
1	0,93	49,91
1,01	1,07	50,11
1,02	0,99	49,94
0,95	1,03	49,81
0,94	0,84	49,83
0,98	0,92	49,78
1,06	1,13	50,17
1,07	1,25	50,09
0,95	0,9	49,41
0,94	0,83	50,02
0,98	0,9	49,85
1,01	1,05	50,28
0,99	0,96	49,67
1	1,15	50,21
0,88	0,95	49,93
1,05	1,14	50,27
0,98	1,02	50,15
0,88	1,02	49,52
1,04	0,97	50,16
0,93	1,04	50,15
1,07	1,17	50,08
1	1,05	49,95
0,94	1,02	50,01
0,97	1,11	49,97
1,03	1,04	50,05
0,94	0,98	50,14
1,01	1,11	50,3

1,03	1,25	50,02
0,94	0,91	50
1,01	1,28	49,51
1,03	1,29	50,31
1,03	1,12	49,72
0,99	1,07	49,98
0,93	1,08	49,6
1,06	1,23	50,61
0,87	0,73	49,86
0,95	0,94	49,86
1	1,03	49,76
0,95	1,07	49,9
1,03	1,24	49,87
1,02	1,05	49,7
0,98	0,94	49,86
0,99	0,99	50,02
1,02	1,16	49,87
0,92	0,89	49,79
1,01	1,03	50,07
0,99	1,02	50,09
1,05	1,24	49,97
1,03	1,15	49,94
0,93	1,07	50
1,06	0,94	49,86
1,02	1,24	49,83
0,96	1,08	50,01
0,97	0,9	49,7
1,06	0,98	50,15
0,99	1,01	50,09
0,99	1,1	49,88
1,07	1,07	50,42
0,92	0,94	49,63
1,01	1,09	50,3
1,05	1,01	50,25
1	1,04	50,33
1,04	1,02	50,04
1,06	1,04	50,29
0,93	1	49,85
0,99	1,02	49,93
1,01	0,98	50,37
0,97	1,09	50,17
0,99	1,01	50,21
0,96	1,03	49,38
1,06	1,13	50,08
1,07	1,19	50,2
1,02	1,09	50,14

1,09	1,16	50,28
1,01	1,13	49,73
0,98	0,91	50,13
0,93	0,95	50,07
0,95	0,83	49,87
0,92	0,83	50,01
1	1,03	49,76
1,01	1,06	49,97
0,93	0,9	50,23
1	1,07	50,25
1,08	1,19	50
0,98	0,95	49,79
1,05	1,14	50,62
0,99	0,84	49,57
0,95	0,93	49,85
1,03	1,11	50,1
1,02	0,92	50,25
0,99	0,92	49,93
1,03	0,98	50,07
0,98	1,28	50,17
0,94	0,99	49,77
1	1,16	50,21
0,92	0,77	49,86
1	1,19	50,16
1	0,94	50,04
0,98	1	49,75
0,91	0,7	49,45
1	1,01	50,15
0,97	0,92	49,99
1,07	1,27	50,26
1,03	1,02	50,23
0,98	0,96	49,94
1,02	1,01	49,99
1,09	1,15	49,8
1,01	1,1	49,93
1,01	1,21	49,84
1,03	1,02	50,2
0,98	0,99	50,17
0,96	1,07	50
1,06	1,14	49,89
1	1,19	49,86
1,04	1,12	49,87
0,97	0,92	49,75
1	1,07	49,92
1,01	1,17	50,06
0,98	0,92	49,97

0,98	0,96	49,88
0,98	0,95	49,78
1	0,93	49,7
1,01	1,01	49,99
0,93	0,93	49,76
0,97	1	50,31
0,91	0,92	49,71
0,94	1,08	49,9
0,9	1,02	49,72
1,02	1,07	50,21
1,03	1,13	50,01
1,02	0,99	49,99
1,04	1,15	49,89
1,03	1,16	49,89
1,03	0,86	50,08
0,97	0,95	50,01
0,94	0,94	50,09
0,97	0,92	49,46
0,94	0,88	49,76
1,05	1,04	50,14
0,91	0,97	49,6
0,98	0,96	49,69
1,06	1,03	50,16
0,97	0,96	49,78
1,01	1,22	50,08
0,97	0,99	49,76
0,95	0,97	49,49
1,02	1,13	50,28
0,93	1,04	49,49
1,07	1,25	50,05
0,99	0,99	50,09
0,93	0,9	49,66
0,97	0,99	49,81
0,93	0,86	49,71
1,01	1	50,03
0,98	1,12	49,84
0,93	0,9	50,13
0,99	1,15	49,53
1,1	1,02	50,44
1,05	1,1	50,03
0,99	1,28	49,97
0,99	1,08	49,85
1,06	1,22	50,1
1,09	1,24	50,31
0,99	1,18	49,99
0,93	0,78	49,55

1,02	1,15	50,15
1	1,23	50,12
0,89	0,76	49,37
1,07	1,14	49,96
0,98	0,96	50,01
0,97	0,94	49,89
0,97	1,08	50,02
0,98	0,96	49,79
0,98	1,18	50,09
1,1	1,15	50,31
1,07	1,3	50,36
1,04	1,14	50,34
0,97	1	50,28
0,98	1,03	49,84
0,94	1,08	49,7
0,94	1,02	49,94
0,9	0,75	50,04
1,14	1,34	50,46
1,04	1,08	50,33
0,96	0,81	50,05
0,9	0,95	49,6
1,03	1,1	50,48
0,94	0,95	49,78
0,99	1,1	49,81
1,05	1,05	50,12
1	0,98	50,13
1	1,14	50,02
0,92	0,97	49,6
1,08	1,02	50,42
0,97	1,01	50,01
0,97	1,08	49,99
1,06	1,12	49,81
1,03	1,24	50,53
1,11	1,14	50,22
0,94	0,91	49,8
0,97	1,13	49,99
1,02	1,18	49,92
0,98	0,98	49,77
1,07	1,03	50,08
0,98	1,16	50
0,85	0,81	49,4
1,08	1,07	49,9
1,05	1,04	50,08
1,1	1,15	50,29
1,05	1,02	50,18
0,95	0,97	49,74

0,92	0,81	49,68
0,99	1	49,87
1,02	1,13	50,12
1,05	1,18	49,85
1,07	1,16	49,98
0,99	1,14	49,97
0,91	0,77	49,7
0,96	0,88	49,78
0,99	1,02	49,88
1,02	1,18	49,97
1,09	1,16	50,21
0,95	0,94	49,87
0,87	0,93	49,64
0,92	1,12	49,68
1	0,99	49,98
1,07	1,26	50,28
1,02	1,03	50,47
1,1	1,13	50,39
1,02	1,09	50,06
1	0,83	49,8
1	1,17	49,89
0,98	1,09	50,07
0,98	1,16	49,98
1,05	1,11	50,33
0,94	1,03	49,66
1,13	1,25	50,58
1,07	1,14	50,36
1,04	1,05	50,55
1,04	0,97	50,12
0,93	1,06	49,42
0,95	1,11	50,12
1,03	1,02	50,24
1,04	1,08	50,07
1,05	1,19	50,6
0,96	1,2	50,16
0,99	0,99	49,9
1,05	1,2	50,02
1,04	1,13	49,93
1,07	1,03	50,14
1	1,03	50,02
1,01	1,18	50,27
0,98	0,92	49,8
1	0,97	49,48
1,03	0,98	50,27
0,92	0,97	49,92
0,99	1	50,03

0,89	0,67	49,67
0,95	0,86	49,63
0,95	0,81	50,05
1,02	1,17	50,1
1,01	0,99	50,08
1	0,98	49,84
0,98	1,22	50,02
1,04	0,95	49,98
1	0,94	50,05
0,9	0,69	49,29
1,07	1,03	50,03
1,01	1,12	50,16
1,02	1,2	50,12
0,97	1,07	50,38
1,05	1,05	49,93
1,01	1,18	49,58
1	1,04	50,06
0,9	0,85	49,52
1,03	1,17	49,87
0,97	1,19	50,15
0,93	0,9	49,69
0,96	1	50,03
1,06	1,1	50,09
1	1,01	49,82
1,02	1,08	49,57
0,98	1,06	50
1	1,07	49,98
0,96	1,01	49,62
0,93	1,16	49,73
0,95	1,1	49,85
0,98	1,04	50,01
1,03	1,19	49,92
1,08	1,31	50,11
0,98	0,97	49,85
0,98	1,17	49,93
0,99	0,9	50,24
1,04	1,13	49,95
1,03	0,88	50,05
0,98	0,89	50,26
1	1,26	49,99
1,02	1,05	50,38
1,07	1,23	50,02
1,06	1,07	50,4
1,04	1,22	49,76
1,05	1,31	50,41
1,01	0,97	50,14

1,06	1,15	50,11
1,03	0,96	49,88
0,98	1,18	50,21
1	0,99	50,04
1,01	0,96	50,22
1,03	1,1	50,12
1	1,08	50,38
0,93	1,05	50,23
1,15	1,14	50,29
1,06	1,28	50,5
1,04	0,99	50,22
1	1,01	49,32
0,99	1,04	49,85
0,96	0,81	49,2
1,02	1,06	50,01
0,96	0,98	49,7
1,02	1,03	50,08
1,01	1,03	50,11
1,03	0,91	50,02
1,01	1,17	49,88
1	1,12	49,63
0,96	0,83	49,66
1	0,93	49,6
1,03	1,14	50,42
1,13	1,2	50,16
0,96	0,92	49,85
1,03	1,1	50,06
0,99	1,22	50,24
0,94	0,91	50,04
1,02	1,15	49,85
1,03	1,11	50,22
1	1,14	50,3
1,01	1,12	50,12
1,09	1,17	50,12
1,03	1,06	50,09
0,94	1	50,04
1,04	1,16	50,67
0,96	1,09	49,97
1	1,07	49,92
1,1	1,15	50,36
1	1,08	50,09
1,01	1,07	50,09
0,95	0,89	49,68
0,95	0,85	49,58
1,01	1,05	50,12
1,06	0,96	49,95

1,01	1,03	49,65
0,91	0,9	50,01
1,05	1,25	50,06
0,99	0,85	49,84
1,01	1,11	50,52
1,01	1,03	50,24
0,97	0,99	50,05
1	0,99	49,7
1,03	1,12	50,17
0,93	1,1	49,93
0,93	1,06	50,38
0,85	0,84	49,53
0,97	1,04	50,02
1,01	1,07	50,09
0,97	0,94	50,12
1,03	1,01	49,83
0,95	0,93	50,04
1,06	1,09	50,02
1,03	1,13	50,31
1,02	1,14	50,28
0,92	0,85	49,79
1,03	1,14	50,09
1	1,02	49,91
0,99	1,15	50,39
1,05	1,07	49,85
1,02	1,09	49,92
1	0,93	50,42
1	1,22	49,79
1,04	1,16	50,28
1,01	1,08	49,88
0,99	1	50,03
1,05	1,1	50,26
1,09	1,05	50,37
1,02	1,03	49,71
0,98	1,03	50,17
0,91	0,96	49,96
1,02	0,97	50,16
0,93	0,86	49,99
1,01	0,98	49,87
0,98	1,03	50,1
0,97	0,89	50,06
0,99	1,11	50,07
1,06	1,22	50,11
0,91	0,93	49,31
0,95	0,82	50,08
1	1,13	50

1,03	0,94	49,99
1,06	1,13	50,04
1,06	1,16	50,01
0,97	1,11	49,92
1,03	1,05	50,44
1	1,06	50,02
1	0,94	49,84
0,98	0,9	49,87
0,96	0,97	49,88
0,99	0,96	50,7
0,98	1,19	50,19
1,05	1,12	50,23
0,99	1,12	49,65
0,97	0,99	49,55
0,99	1,18	49,84
1,07	1,3	50,11
1,07	1,04	50,16
1	1,09	50,13
0,92	0,93	49,92
0,97	1,15	50,15
1,01	1,16	49,87
1	1,03	49,82
1,12	1,19	50,57
1,01	1,32	50,15
1,16	1,22	50,85
1,03	1,24	49,68
1,01	1,13	50,06
0,98	0,93	49,94

Affidavit

Name : El Islami

First Name : Muhammad Akhiruddin

Matriculation Number : 28631

Course of Study : Industrial Engineering

"I, Muhammad Akhiruddin El Islami, hereby declare in lieu of oath that I have written this thesis entitled "Multivariate Control Charts" independently, have not used any sources or aids other than those indicated, and have acknowledged all citations. I affirm that the paper has not been submitted in the same or a similar version to any other course of study as an examination."

Merseburg, 28.05.2024

Muhammad Akhiruddin El Islami