



TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

GRADUATION PROJECT REPORT

*as a partial fulfillment of the degree of
Bachelor of Science in Business Administration*

Product Efficiency Measurement and Performance Ranking :

PCA-DEA

A Combined Model Approach for Dimension Reduction

By:
Farah Aboucha

Academic Advisor:
Phd. Lassad El Moubarki

Professional Advisor:
Lauriane Fessaguet

Realized within:



Academic Year:
2022 - 2023

Approval

Academic Advisor

Phd. Lassad El Moubarki



14/06/2023

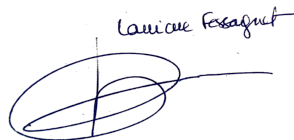
Name

Signature

Date

Professional Advisor

Lauriane Fessaguet



12/06/2023

Name

Signature

Date

Academic Evaluator

Name

Signature

Date

Declaration

I certify that I am the author of this report and that any assistance I have received in its preparation is fully acknowledged and disclosed in this report. I have also cited any source from which I used data, ideas, or words, either quoted or paraphrased. Further, this report meets all the rules of quotation and referencing in use at Tunis Business School and adheres to the fraud policies listed in the Tunis Business School's honor code. No portion of the work referred to in this report has been submitted to support an application for another degree or qualification to this or any other university or academic institution.

Farah Aboucha



12/06/2023

Name

Signature

Date

Acknowledgements

This report represents the culmination of four years of my academic journey at Tunis Business School, along with two years of professional experience at VistaPrint. I am indebted to everyone who has played a role in shaping me into a better version of myself.

First and foremost, I would like to express my gratitude to my academic supervisor, Phd. Lassad El Moubarki, for his guidance and support throughout the compilation of this paper.

I would like to thank my professional supervisor, Lauriane Fessaguet, for placing immense trust in my abilities and consistently encouraging me to surpass my own expectations.

I would like to extend my gratitude to my manager, Silvia Lussana, whose exceptional mentorship and encouragement have inspired me to overcome challenges and push the boundaries of my work.

I would also like to acknowledge the incredible support I have received from all members of Cimpres, Vista, and the Europe Product team. Their collaborative spirit have created an environment conducive to growth and development.

Lastly, I would like to express my appreciation to every teacher, instructor, and professor who has contributed to my academic journey. Each of them has left a great impact on my intellectual growth and has played a significant role in shaping me into the person I am today.

Dedication

I dedicate this work to:

My dearest friends, for being by my side all these years and giving me the necessary moral boost when I need it the most.

My incredible family, for their unwavering support, encouragement, and enduring motivation.

My brother, Mohamed Iheb, for being the constant source of inspiration from the very beginning.

My parents, Seima Lakhal and Mondheur Aboucha, for loving me unconditionally, encouraging me passionately, and supporting me endlessly.

mom, dad, all I have ever achieved is eternally dedicated to you.

Abstract

In today's competitive global market, accurately measuring and evaluating product performance is crucial for organizations. It enables the identification of high-performing products that significantly contribute to revenue generation and profitability, as well as the detection of underperforming products that may require improvement or strategic changes. To enhance the evaluation of product performance, this study employed a combination of Principle Components Analysis (PCA) with Data Envelopment Analysis (DEA). The study included a sample of 34 product groups, with 4 input variables and 3 output variables. First, the PCA was applied to reduce the dimensionality of the input variables to 2 principal components, namely Quality Control and Cost Management. Similarly, the output variables were transformed into 2 principal components, namely Business Performance and Demand Intensity. The transformed components were then used as inputs and outputs for DEA to measure the relative efficiency of the 34 product groups. Four different models were implemented, and the fourth model was ultimately selected based on the study's findings. This model identified 4 efficient product groups that served as benchmarks for the remaining 30 inefficient product groups. To facilitate Benchmarking, a ranking according to their efficiency scores was proposed. Furthermore, a ranking based on peers' weights was provided, allowing for a comprehensive comparison of the inefficient product groups. An excess analysis was finally conducted to shift the focus of improvement efforts towards the specific variable that influences efficiency the most.

Keywords: DEA, PCA, Dimension Reduction, Efficiency, Benchmarking

List of Tables

1.1	Cimpress Vista key information	2
2.1	Inputs and Outputs variables per Product Group	10
3.1	Label of PC 1 from Input variables	21
3.2	Label of PC 2 from Input variables	21
3.3	Label of PC 1 from Output variables	21
3.4	Label of PC 2 from Output variables	21
3.5	Summary of Principle Components labels per Input and Output variables	22
3.6	The DMUs with efficiency score of 1 - Model 1	23
3.7	The DMUs with efficiency score of 1 - Model 2	25
3.8	The DMUs with efficiency score of 1 - Model 3	27
3.9	The DMUs with efficiency score of 1 - Model 4	28
3.10	The peers weights	30
3.11	Summary of peers relationship	31
3.12	Ranking of DMUs based on Efficiency scores	33
3.13	Ranking of DMUs based on Peers Weights	34
3.14	Excess values per DMU	35

List of Figures

1.1	Historical Vista Revenue (2018-2022)	1
2.1	The Stochastic Frontier Model for Efficiency Analysis	6
2.2	The PCA-DEA Combined Model Methodology	7
2.3	DEA approach example for Benchmarking	8
2.4	Difference between CRS and VRS in DEA frontier	13
2.5	The Input-Oriented BCC Model	13
3.1	The Input Scree Plot	15
3.2	The Output Scree Plot	15
3.3	The Importance of Input components	16
3.4	The Importance of Output components	16
3.5	PCA graph of Input Variables	17
3.6	PCA graph of Output Variables	18
3.7	Contribution of Input Variables to PC 1	19
3.8	Contribution of Input Variables to PC 2	19
3.9	Contribution of Output Variables to PC 1	20
3.10	Contribution of Output Variables to PC 2	20
3.11	The DEA frontier with Model 1: the Basic DEA Model	23
3.12	The distribution of efficiency scores of Model 1	24
3.13	The DEA frontier with Model2: Input Dimension Reduction	24
3.14	The distribution of efficiency scores of Model 2	25
3.15	The DEA frontier with Model3: Output Dimension Reduction	26
3.16	The distribution of efficiency scores of Model 3	27
3.17	The DEA frontier with Model4: joint Input-Output Dimension Reduction	28
3.18	The distribution of efficiency scores of Model 4	29

List of Abbreviations

DMU Decision-Making Unit

DEA Data Envelopment Analysis

PG Product Group

CCR Charnes - Cooper - Rhodes

BCC Banker - Cooper - Charnes

SFA Stochastic Frontier Analysis

FDH Free Disposal Hulls

PCA Principal Component Analysis

PC Principal Component

PCA-DEA Principal Component Analysis - Data Envelopment Analysis

CRS Constant Return to Scale

VRS Variable Return to Scale

NPS Net Promoter Score

Contents

1	Introduction	1
1.1	Company presentation	1
1.1.1	General presentation	1
1.1.2	Mission, Vision and Values	2
1.1.3	Industry	2
1.1.4	Products and services	3
1.2	Role presentation	3
1.2.1	Job description	3
1.2.2	Problematic	4
1.2.3	Solution	4
1.3	Conclusion	4
2	Methodology	5
2.1	Literature review	5
2.1.1	Review of efficiency measurement techniques and limitations	5
2.1.2	Introduction of PCA to DEA	7
2.1.3	PCA-DEA for Benchmarking	7
2.1.4	Review of existing literature	8
2.2	Data understanding and pre-processing	9
2.2.1	Data structure	9
2.2.2	Pre-processing of the data	10
2.2.3	Packages and libraries	10
2.3	Data modeling	11
2.3.1	Input and output orientation	11
2.3.2	Model specifications	12
2.3.3	Model selection	13
3	Implementation and Results	14
3.1	PCA Application	14
3.1.1	Selection of the Principle Components	14
3.1.2	Interpretation of the retained Principle Components	17
3.1.3	Descriptive labeling of Principle Components	20
3.2	PCA-DEA Application	22
3.2.1	Model 1: PCA-DEA Analysis with no variable reduction	22
3.2.2	Model 2: PCA-DEA Analysis with Input dimension reduction	24
3.2.3	Model 3: PCA-DEA Analysis with Output dimension reduction	26
3.2.4	Model 4: PCA-DEA Analysis with joint Input-Output dimension reduction	28
3.3	Benchmarking and ranking	29
3.3.1	Peers Analysis	29
3.3.2	Ranking of the DMUs	32
3.3.3	Excess analysis	34

3.4	Business Findings	36
4	General Conclusion	37
4.1	Discussion of the findings	37
4.2	Limitations	38
4.3	Recommendations	39

Executive Summary

This study aims to provide VistaPrint, a global leader in e-commerce and mass customization, with a comprehensive ranking system for a sample of product groups based on multiple variables. The primary objective was to improve the performance of inefficient products and influence future product introductions. To achieve this, a combined approach of Principal Component Analysis (PCA) with Data Envelopment Analysis (DEA) was employed, streamlining the evaluation process and reducing complexity.

The study started with a PCA application, which effectively reduced the initial set of 4 input and 3 output variables to 2 inputs and 2 outputs that explain the majority of the variance in the model. Subsequently, an input-oriented model was employed, focusing on minimizing the new input variables that were labeled Quality Control and Cost Management. Second, the basic DEA model and three other combined PCA-DEA models were tested to evaluate the efficiency of the product groups. Model 1, the basic DEA model, identified 23 efficient product groups out of the initial 34. Models 2 and 3 further refined the evaluation by reducing the dimensionality of inputs and outputs, respectively. Finally, Model 4 incorporated the joint reduction of both inputs and outputs, resulting in a more detailed efficiency evaluation and identifying 4 efficient product groups. The findings underscore the advantage of employing the PCA-DEA model when dealing with a large number of variables, as it provides a clearer distinction between efficient and inefficient product groups. Additionally, specific product groups were identified that consistently demonstrated efficiency across different models, as well as those that exhibited improved efficiency after dimension reduction. Based on these findings, the fourth model was selected for further analysis with 4 benchmarks that can serve as a reference for future product introductions, contributing to the ongoing improvement of efficiency. The ranking of product groups based on efficiency scores and peers' weights offers valuable insights for decision-makers to optimize efficiency and enhance overall performance. Furthermore, the excess analysis prioritized efforts towards improving Quality Control, recognizing its significant impact on efficiency. By focusing on enhancing Quality Control measures, VistaPrint can effectively address the identified areas of improvement and further optimize their operations.

Chapter 1

Introduction

The purpose of this chapter is two-fold : First, present the host company Cimpres Vista with an outlook of its industry, mission, vision, values, products, and services. Second, the chapter will cover a description of the position, learning outcomes, duties and responsibilities.

1.1 Company presentation

Cimpres is a global leader in the field of mass customization, owning several brands, including Vista, National Pen, BuildASign, Drukwerkdeal, Exaprint, Pixartprinting, and Printi. In what follows, a presentation of Vista’s industry, products and services is provided.

1.1.1 General presentation

Vista is a key part of Cimpres, and it is the regroupement of VistaCreate, 99designs by Vista, and VistaPrint. Vista is the design and marketing partner to millions of small businesses around the world with over 1.5 billion dollars annual revenue in 2022, as shown in Figure 1.1. It offers affordable and professional options to more than 17 million small businesses and consumers each year and provides a range of products and services to help these businesses grow. Vista was founded in 1999 by Robert Keane, and it has a unique business model supported by patented technologies, high-volume manufacturing facilities, and direct marketing expertise. It is a multinational corporation with over 10,000 employees in 24 countries, 22 multilingual websites, production facilities in Windsor, Ontario and Venlo, Netherlands, and supplies to over 120 countries. Table 1.1 shows the important key information of Cimpres Vista.

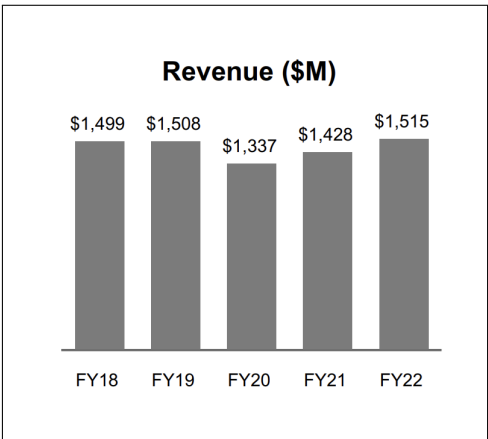


Figure 1.1: Historical Vista Revenue (2018-2022)*

*Source: [2]

Table 1.1: Cimpres Vista key information

Legal name of the company	Cimpres
Founder	Robert Keane
Date of foundation	January the 1st ,1995
Place of foundation	Paris
Number of employees	10 000
Number of customers	16 000 000
Activity	E-commerce
Website	www.cimpres.com

1.1.2 Mission, Vision and Values

The mission-vision-values represents 3 main pillars of corporate development. They involve articulating the purpose and objectives of the company, envisioning the desired future state and goals, and outlining the core beliefs and principles that drive the organization's behavior and decision-making.

- **Mission:** Vista has earned the trust of millions of small companies worldwide as a reliable partner for design and marketing solutions. Over the years, Vista has been dedicated to supporting businesses in promoting their brands by delivering high-quality products and services at an affordable price. This commitment has made Vista a preferred choice for businesses seeking cost-effective yet impactful solutions to enhance their brand visibility and marketing efforts.
- **Vision:** The company has a clear vision, OWN THE NOW!, which signifies Vista's unwavering dedication and strong willingness to assist business owners in seizing the current opportunities and maximizing their potential. This vision reflects the company's commitment to providing the necessary support to enable small business owners to take charge of their success in the ever-evolving business landscape.
- **Values:**
 - Obsess over customers and creators
 - Act like an owner
 - Be bold
 - Be data driven
 - Find, grow, and support great people
 - Relentlessly learn and improve

1.1.3 Industry

Vista is an e-commerce company that operates majorly in the online printing industry. The online printing industry refers to the sector within the printing industry that offers printing services to customers through websites or online portals. This term can be traced back to the late 1990s and early 2000s, when the internet was becoming more accessible to the general public. Vista offers a wide range of printing services through a user-friendly platform. Customers have the option to personalize their orders by uploading their own designs or choosing from templates provided by the company. Vista has established itself as a prominent player in the e-commerce sector, meeting the growing demand for customizable and cost-effective printing solutions. By leveraging technology and a customer-centric approach, the company has successfully positioned itself as a trusted provider of printing services in the e-commerce landscape.

1.1.4 Products and services

Vista offers comprehensive design, digital, and print solutions that enhance the presence of small businesses in both physical and digital realms, empowering them to achieve success.

1. Products: are the customized print solutions provided through VistaPrint worldwide e-commerce sites.
 - Business Cards: *Plastic, Magnetic, Folded*
 - Packaging and Stationary: *Product Boxes, stickers*
 - Signage: *Banners, Posters, Flags*
 - Marketing Materials: *Flyers, Postcards, Booklets*
 - Consumer: *Calendars, Blankets, mugs*
 - Clothing and Bags: *T-shirts, Headwear, Tote bags*
2. Services: are the digital solutions supported by VistaCreate, 99Designs by Vista and VistaPrint.
 - Design Services:
 - Logo and identity
 - Web and app design
 - Business and advertising
 - Clothing and merchandise
 - Art and illustration
 - Book and magazine
 - Website Creation:
 - Customized Templates
 - Advanced features: Blog page, online store, bookings system.
 - Search engines Optimization
 - Mobile view Editor
 - Digital Marketing:
 - Social media marketing
 - Email marketing
 - Online listings management

1.2 Role presentation

The Product Operations Specialist position is classified under the Product Department. This position focuses on supporting the Back-end experience of a specific line of business, which is, in this case, the Signage category for the Europe market.

1.2.1 Job description

The Product Operations Specialist role ensures that the company's products are developed, produced, and delivered to customers in a timely and efficient manner while meeting their needs and expectations. This position's responsibilities include:

- Manage the product development process from ideation to launch through collaborating with cross-functional teams such as:

- Coordinate with manufacturing and logistics teams to ensure that products are produced and shipped on time.
 - Collaborate with third-party fulfillers to monitor performance and troubleshoot issues that may arise during the order fulfillment process.
 - Collaborate with marketing teams to develop product messaging and positioning.
 - Collaborate with the Customer care and Technology teams to address order issues impacting customers including product quality, artwork issues and system bugs.
- Analyze product performance data and making recommendations for product improvements.
 - Setup and maintain product configurations including sku creation, scene linking, gallery configuration, and managing equivalency to other products.

1.2.2 Problematic

The product department plays a key role in monitoring and introducing new products while ensuring their success. Currently, performance tracking relies on various metrics like net bookings, profit, and order count. However, the existing challenge lies in developing a comprehensive ranking system that considers multiple performance metrics. While these metrics do offer valuable insights when examined individually, they fail to provide a holistic view of a product's performance. For instance, it is common for a product to have high bookings but also attract a significant number of customer complaints, which can impact its overall performance. Hence, it is necessary to develop a ranking system that effectively combines these metrics to deliver a comprehensive evaluation of products.

1.2.3 Solution

One potential solution to address the problem of integrating multiple performance metrics and providing a comprehensive evaluation of products is by using a combined model that incorporates Principal Component Analysis (PCA) and Data Envelopment Analysis (DEA) techniques. PCA can be employed to reduce the dimensionality of the metrics by identifying the most significant components that capture the majority of the variation in the data. This helps in overcoming the issue of metric weighting and allows for a more unbiased assessment of product performance. DEA, on the other hand, is a non-parametric method widely used for Benchmarking and ranking decision-making units, in this case, products, based on their relative efficiency. By considering inputs and outputs of each product, DEA can determine the relative performance efficiency of different products. Therefore, the PCA-DEA combined model not only facilitates ranking of current products but also serves as a powerful tool for Benchmarking future products.

1.3 Conclusion

In this chapter, we have provided a comprehensive overview of the hosting company including its mission, vision, and values. We have also discussed the diverse range of products and services it provides and the industry it operates in. Additionally, we have highlighted the role presentation, job description, and the challenges faced. Finally, we described the proposed solution to address these challenges.

Chapter 2

Methodology

This chapter provides an overview of the methodology employed in the PCA-DEA approach. It starts by examining the existing techniques for measuring efficiency and their inherent limitations. To overcome these limitations, the PCA-DEA approach was chosen as the methodology for this project. Additionally, the chapter highlights previous studies that have successfully applied the PCA-DEA approach to measure efficiency in various industries. Furthermore, the chapter introduces the model and techniques utilized in this project, providing insights into their application and relevance.

2.1 Literature review

In many industries, evaluating the performance of products or systems depends heavily on efficiency measurement. Organizations can use it to pinpoint areas for improvement, allocate resources more effectively and reach wise decisions. Researchers have been looking into cutting-edge methodologies to improve efficiency measurement techniques in recent years. This review of the literature aims to examine the body of knowledge that already exists on the subject of measuring efficiency of product performance, with a focus on previous PCA-DEA applications.

2.1.1 Review of efficiency measurement techniques and limitations

Numerous techniques for measuring efficiency have been studied and applied, employing various methodologies and approaches. While some well-known methods have been successfully employed in real-world situations, it is important to acknowledge that these techniques also have certain limitations. In the following discussion, we provide few examples of these techniques and some limitations.

1. Review of efficiency measurement techniques:

- **Data Envelopment Analysis (DEA):** DEA is a widely used method for assessing efficiency that was first introduced in 1978. The term was coined by Edwardo Rhodes, George E. Cooper, and Abraham Charnes in a report based on Rhodes' PhD research. Instead of using a single output-to-input ratio, DEA offered a novel way to evaluate efficiency by comparing the input-output relationship of decision-making units (DMUs). This approach was a response to the need for a more comprehensive assessment of efficiency beyond simple ratios. Since then, DEA has been used by researchers in a variety of fields and industries. For instance, DEA was developed by Emmanuel D. Banker, Abraham Charnes, and William W. Cooper in 1984 to assess the relative efficiency of bank branches. Their work served as the basis for later studies and DEA applications in the banking industry.
- **Stochastic Frontier Analysis (SFA):** SFA is another common method for assessing efficiency. It takes into account both random error and estimation process inefficiency as shown in Figure 2.1. Charles Aigner, J. Dennis Lovell, and Robin C. Sickles in 1977 produced a

seminal piece of work in this area by introducing the idea of stochastic frontiers and creating the SFA model. Since then, SFA has been widely used to measure technical efficiency and pinpoint inefficiency-causing factors in a variety of industries, including manufacturing, agriculture, and the healthcare sector.

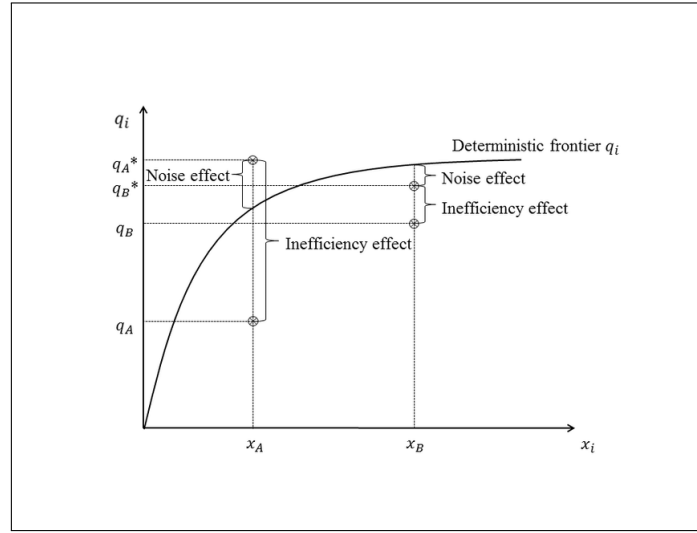


Figure 2.1: The Stochastic Frontier Model for Efficiency Analysis^{*}

- **Free Disposal Hulls (FDH):** Efficiency can also be measured using a non-parametric technique called the Free Disposal Hull (FDH) approach. The most effective frontier is chosen based on the distribution of the convex hull of the observed data points. To advance FDH models, researchers like Paul Simar and Thierry Post in 2007 have made a significant contribution. They suggested the FDH bootstrap approach, which makes the FDH approach more dependable and robust by allowing the estimation of confidence intervals for efficiency scores.

2. Limitations:

- **Input-Output proportionalities assumption:** Empirical evidence from study [7] shows that DEA displays significant bias and limited accuracy. The extent of bias and accuracy is influenced by the variability and correlation among inputs. To address the validity issues associated with the conventional method's estimates, a suggested alternative approach is a reverse two-stage procedure that provides unbiased and more precise estimates.
- **Measurement Error:** Prior study [20] has indicated that DEA and SFA models, when applied to cross-sectional data, are negatively influenced by measurement errors. This study suggested to use panel data models as they perform better by incorporating information from multiple time periods, thereby improving estimation. Specifically, a panel data DEA model utilizing averaged data has demonstrated effectiveness in reducing measurement error.
- **Discriminatory Power:** An article [1] findings showed that DEA may result in a high number of units being classified as efficient due to the flexibility in weight selection. In an analysis with multiple inputs and outputs, the potential for numerous efficient units exists. This study suggested that to ensure meaningful discrimination between units, the number of units in the analysis set should significantly exceed the expected number of efficient units, which is estimated to be the product of the number of outputs and inputs.
- **Lack of adaptability:** According to the research [23], the rigidity of conventional efficiency techniques has restricted their applicability in a variety of situations. They promote the

^{*}Source: [23]

use of adaptable functional forms, like non-parametric models or econometric procedures, to adequately represent the complexity and variety of real-world production processes.

2.1.2 Introduction of PCA to DEA

Karl Pearson, a British statistician and mathematician, developed principal component analysis (PCA) in 1901. The PCA concept was created by Pearson as a method for analyzing and streamlining complicated datasets by locating the most important patterns and relationships between variables. His contributions laid the groundwork for PCA to become a core method in multivariate data analysis. Harold Hotelling and Wold Herman, among others, have improved and expanded PCA over time, further solidifying its position as a potent tool for dimension reduction, feature extraction, and data visualization. Researchers acknowledged the need to reduce computational burdens and enhance efficiency measurements in DEA in the late 1990s and early 2000s. As Figure 2.2 from study [19] shows, the analysis involves two stages. In the first stage, the input and output variables are reduced in dimensionality using PCA. This step allows to identify the key factors influencing the variables and maintain the most information possible. In the second stage, DEA is applied to the transformed variables to obtain more realistic results. By providing a clear representation of the data, PCA's use in DEA not only improved computational efficiency but also made the analysis easier to understand. With the help of this integration, researchers were able to more successfully handle high-dimensional datasets and obtain more accurate efficiency measurements, which represented a significant advancement in the application of DEA.

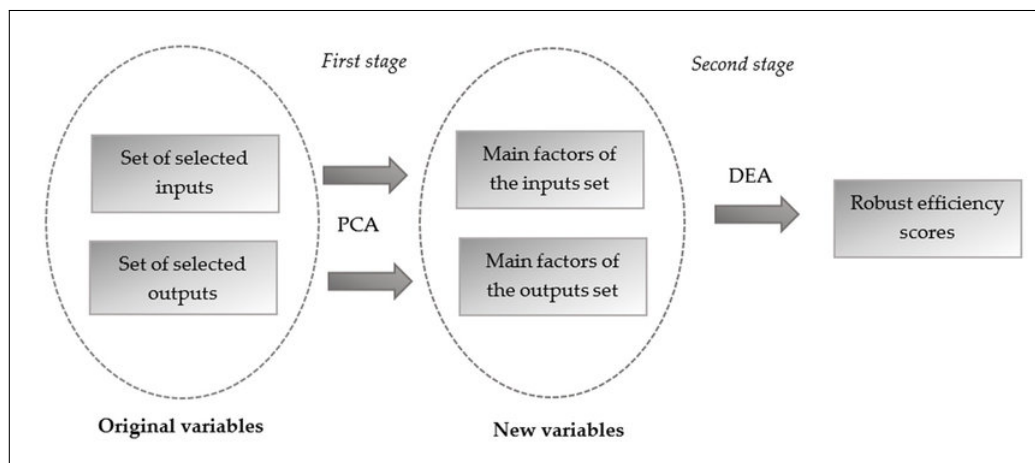


Figure 2.2: The PCA-DEA Combined Model Methodology *

2.1.3 PCA-DEA for Benchmarking

DEA serves as a powerful Benchmarking tool to assess the performance of a group of individuals, referred to as DMUs, in converting inputs into outputs. The primary objective is to identify the DMUs that achieve the most efficient transformation, represented by the units located on the efficiency frontier. On the other hand, DMUs falling behind the frontier are considered less-efficient.[11]

1. **DEA frontier:** Efficiency scores in DEA are indicative of a unit's performance. A score of 1 is assigned to DMUs located on the efficiency frontier, implying a perfect balance between input and output utilization. Inefficient points, positioned behind the frontier, receive scores lower than one, indicating a shortfall in output relative to input usage. The efficiency frontier comprises efficient

*Source: [19]

observations with scores of one and is bounded by the convex hull, representing the best practice or benchmark performance.

2. **Peers Analysis:** Each benchmark point can be expressed as a combination of its reference units or peers, typically represented by the corner points of the current frontier line. Peers play an important role in the analysis as they serve as role models from which an inefficient DMU can learn and improve its performance. They help identify the best practices and strategies that can be adopted to enhance efficiency.
3. **Excess Analysis:** For DMUs operating below the frontier, DEA calculates the potential additional output that could be achieved by adopting best practices, employing the Excess analysis. The Excess analysis involves projecting the inefficient unit towards the efficiency frontier while proportionally increasing all outputs while using the same inputs. This target value obtained from the Excess analysis serves as a guide for inefficient units, indicating the level of performance they could potentially achieve by aligning with the best practices represented by the frontier.

The DEA approach for Benchmarking is demonstrated in Figure 2.3 from study [26], where the efficient projections define targets for improving the efficiency of a specific DMU. In this example, DMUs A, B, C, D, and F serve as benchmarks due to their efficiency. They form the peers group for inefficient DMUs such as point E. The orientation of improvement is indicated by the arrows at DMU E, with the x-axis representing Input Oriented projections for potential improvements through input reduction (A and B as peers), and the y-axis representing Output Oriented projections for maximizing output (C and D as peers).

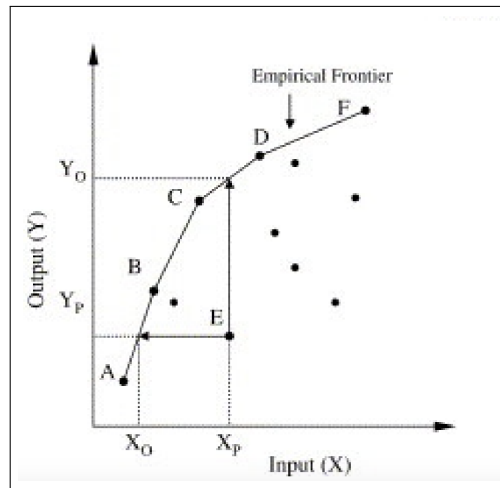


Figure 2.3: DEA approach example for Benchmarking *

2.1.4 Review of existing literature

The use of the PCA-DEA combined model approach for efficiency measurement in various domains has been investigated in a number of studies. For instance, an application of the PCA-DEA approach was utilized to assess the green growth of clean energy-environmental efficiency in China, in the article [9]. This study revealed that although China has experienced a notable upward trend in green growth in recent years, its overall level remains relatively low. Another example was the use PCA-DEA approach to improve the risk management efficiency of commercial banking system in China in the study [25]. The findings show that there is a significant issue with economies of scale in the risk management practices

*Source: [26]

of Chinese commercial banks. In Benchmarking and ranking, PCA-DEA approach was used in the study [18] to assess the quality of life and predict its scores and rankings of Estonian counties. It also provides a revised ranking using the combined model. Additionally, an assessment of the efficiency of transport companies over eight years was conducted in the study [27]. The researchers provided alternatives and ranking to determine the final efficiency of transport companies based on 10 input–output parameters. To summarize, these researchers have made significant contributions to their respective fields by advancing the application of the PCA-DEA combined model approach in efficiency assessment, Benchmarking and ranking. Their studies highlight the versatility and effectiveness of the PCA-DEA approach in measuring efficiency in various domains, ranging from environmental and social studies to risk management and transportation economics.

2.2 Data understanding and pre-processing

This section focuses on gaining insights into the variables used in the analysis and preparing the data for further investigation. This involves exploring the data to understand its characteristics, assessing data quality to ensure reliability, addressing missing values, and applying necessary transformations to variables for improved analysis. By undertaking these steps, the section ensures that the data is properly processed and ready for subsequent analysis and modeling.

2.2.1 Data structure

Selecting appropriate input and output variables is decisive in the context of DEA. The chosen variables should align closely with the goals and objectives of the firm being analyzed. In this section, a representation of the input and output variables is discussed and shown in Table 2.1 .

1. Inputs:

- **Order rejections per PG:** It is a useful indicator to track the performance of a products. The company keeps track of the number of orders that fail to reach the production process and aims to minimize it. Higher rejection rates could be a sign of problems with product setup or incorrect specifications.
- **Customer complaints per PG:** An essential component for determining the product efficiency is tracking their customer complaints. This indicator offer insightful feedback on the product's features, quality, and customer service.
- **Shipping cost per PG:** It constitutes a key factor in determining how effectively the product are operating. Effective shipping Cost Management supports cost reduction and streamlined processes.
- **Product cost per PG:** It serves as a decisive indicator of cost efficiency and resource allocation in relation to the output generated.

2. Outputs:

- **Gross profit per PG:** It is a key financial metric that shows the profitability of the products. It enables evaluation of the efficiency of resource management, pricing tactics, and cost control in relation to revenue production.
- **Order count per PG:** It is the count of items ordered for each product group. It is a crucial metric for determining how successful a product is.
- **Net promoter score per PG:** It is a well-known metric for assessing customer satisfaction and loyalty. It constitutes The likelihood that customers will recommend a product to others.

Table 2.1: Inputs and Outputs variables per Product Group

Inputs	Outputs
Order rejections	Gross profit
Customers complaints	Order count
Shipping cost	Net promoter score
Product cost	

2.2.2 Pre-processing of the data

Data preprocessing involves transforming raw data into a suitable format for analysis and modeling. It helps improving data quality, handling missing values, and preparing the data for specific analysis tasks.

1. Filling missing data:

The datasets used in this study are made up of weekly observations from week 30 in 2022 to week 13 in 2023. These dates fall between July 2022 and March 2023. It was identified that 5 datasets among a total number of 7 contained missing values. To address this issue, the interpolation technique was employed in Python. This technique allows for the estimation of missing values by leveraging the available data from similar time periods. The conversion into numeric type was performed to ensure that the data was in a suitable format for further analysis. The filled data, now in numeric format, was saved and prepared for the subsequent steps of the analysis.

2. Summarize the time periods:

To make sure that underlying trends could be understood more clearly, the average of this time period was calculated each time. By reducing the impact of outliers and extreme values, it improves stability. In addition to providing a summary of performance over the time period, it is also simpler to manipulate and interpret. A different strategy was used for the order rejections. The data was aggregated over the time periods. Similarly, the Net Promoter score data remained the same as we have decided to include only one month for data unavailability. These methods give a more accurate picture of the variables and make analysis and modeling more accurate.

3. Data Normalization:

Once all the datasets were prepared, the normalization step was performed using the Min Max Scaler from the Scikit-learn library. Normalization was necessary to ensure that all input and output data for each DMU were on the same scale, as it is essential for computing the ratio of outputs to inputs in DEA analysis. The MinMaxScaler rescaled the data to a common range, enabling a fair comparison and accurate efficiency calculations among the DMUs. The normalized data was then merged into a single file, ready for further DEA computations.

2.2.3 Packages and libraries

In order to effectively conduct the PCA-DEA analysis for the performance of products, several essential software libraries were used. These tools enabled us to analyze data, visualize key insights, and conduct Benchmarking. This presentation provides an overview of the packages and libraries used in this study.

- **pandas:** is a fundamental library for data manipulation and analysis in Python. It provides powerful data structures to read, clean, and pre-process data, facilitating subsequent analysis and modeling tasks.
- **os:** facilitates the interaction with the operating system. It is used to navigate directories, access files, and manage paths.

- **sklearn.preprocessing.MinMaxScaler:** The MinMaxScaler class from the scikit-learn (sklearn) library is used for feature scaling. It scales features to a specified range, typically between 0 and 1, by normalizing the data.
- **reticulate:** is an R package that allows the integration of Python code and functionality into the R environment.
- **Benchmarking :** allows for the comparison of performance of different products and identification of best practices using DEA application, peers and excess computation.
- **readxl:** is used to read Excel files directly into R. It simplifies data extraction and preprocessing.
- **writexl:** complements readxl by allowing to write dataframes and analysis results back to Excel files.
- **openxlsx:** provides functions for reading, writing, and manipulating Excel files. It imports and exports data from Excel, create of new sheets, and modify existing sheets.
- **ggplot2:** is a popular data visualization library in R. It allows an effective communication of the findings and visual representation of key performance metrics and trends.
- **ggpubr:** extends the capabilities of the ggplot2 package. It provides additional functions and features for creating publication-quality plots and graphics.
- **caret:** provides a unified interface for performing machine learning tasks in R. It offers algorithms and tools for predictive modeling and data preprocessing.
- **factoextra:** allows for exploratory data analysis and clustering techniques. It provides functions for extracting and visualizing information from multivariate analysis methods such as PCA.
- **reshape2:** reshapes and transforms data sets. It offers functions for converting data between different formats. It is used for data manipulation and restructuring tasks.
- **dplyr:** allows to filter, select, arrange, summarize, and modify data frames.

2.3 Data modeling

The selection of an appropriate model is important as it determines the way efficiency is measured and interpreted. In what follows, an explanation of the chosen model and the underlying rationale behind its selection is provided.

2.3.1 Input and output orientation

By comparing input and output ratios, DEA assesses the relative efficiency of the DMUs. Input-oriented and output-oriented DEA models are the two basic categories that can be distinguished.

- **Input-oriented model:** This model try to minimize the number of resources needed for an input to achieve a certain level of output. It presupposes that the DMUs are already achieving the optimal combination of outputs and that efficiency gains can be made by cutting back on the quantity of inputs used. When DMUs are faced with input limitations like constrained finances, raw resources, or manpower, input-oriented models might be helpful. The inputs serve as the benchmark for comparing the DMUs in an input-oriented DEA model. The goal of the model is to determine the optimal weights for each input that maximize the DMUs' effectiveness. The DEA model then identifies the most efficient DMUs as those that require the fewest inputs to produce a given level of output.

- **Output-oriented model:** The goal of output-oriented models is to produce a maximum of output from a fixed amount of inputs. These models presuppose that the DMUs are not yet utilizing the optimal combination of inputs and that efficiency gains can be made by increasing the amount of output generated with a given input level. When DMUs are faced with output restrictions like limited demand or market saturation, output-oriented models can be helpful. The outputs serve as the basis for comparison in an output-oriented DEA model. The goal of the model is to determine the optimal weights for each output that maximize the DMUs' efficiency. The DEA model then identifies the most efficient DMUs as those that produce the most output from a given level of inputs.

In conclusion, there are two primary types of DEA models: input-oriented and output-oriented. These two types of DEA models differ in their objective functions. Output-oriented models seek to maximize outputs for a given level of inputs rather than minimize inputs for a given level of output, as opposed to input-oriented models. The efficiency of DMUs has been assessed using both models in a variety of fields in order to pinpoint best practices and potential areas for development.

2.3.2 Model specifications

DEA introduces several models for measuring the relative efficiency of the DMUs, each with its own variations and assumptions to capture different aspects of efficiency and performance evaluation. The CCR and BCC models are two DEA models that are frequently employed.

- **The CCR model:** It was developed in 1978 by the researchers Rhodes, Cooper, and Charnes. It was named after the first initials of their last names. It is a non-radial DEA model that makes the Constant Return to Scale (CRS) assumption. It increases efficiency while maintaining constant input and allowing for variable output. The resulting efficiency scores show each decision-making unit's (DMU) capacity to minimize inputs while generating the level of outputs that is observed. Since its development, the CCR model has served as a foundation for the development of other DEA models
- **The BCC model:** It was developed by the researchers Banker, Charnes and Cooper, and was also named after the initials of their names. It was proposed as an extension to the CCR model to address the issue of variability. Unlike the CCR model, the BCC model makes the variable returns to scale (VRS) assumption. It considers both input reduction and output expansion to determine the efficiency scores of DMUs. It allows for varying levels of efficiency due to differences in scale efficiency among decision-making units (DMUs).

Figure 2.4 from study [22] illustrates the estimation of the efficiency frontier under both scale assumptions using four data points: A, B, C, and D. It is important to note that only fixed inputs are taken into account in this analysis. The Constant Returns to Scale (CRS) from CCR frontier is represented by point C. Along this frontier, all other points aside from point C are considered inefficient. On the other hand, the Variable Returns to Scale (VRS) from BCC frontier is described by points A, C, and D, while point B lies below the frontier. It is evident the capacity output of variable returns to scale is less than the capacity output of constant returns to scale. [21]

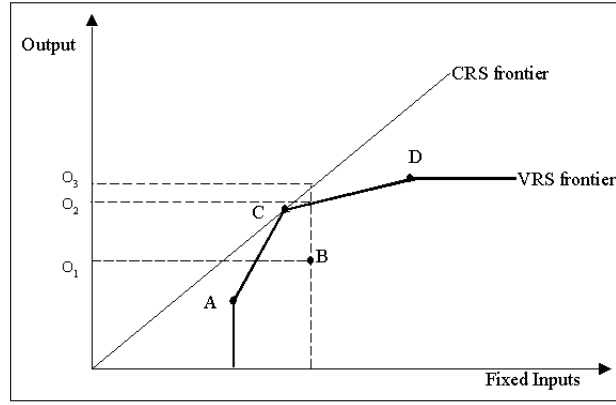


Figure 2.4: Difference between CRS and VRS in DEA frontier *

2.3.3 Model selection

The use of the input-oriented BCC model in this study is suitable for a variety of reasons. First, the objective of this study is to minimize the inputs, particularly costs associated with shipping and product costs, and inefficiency indicators such as the order rejections and customer complaints, while maintaining or maximizing the output. The input-oriented model is well suited for analyzing and optimizing the effective management of these inputs. Second, the BCC model allows for variable returns to scale (VRS). In this study, some product groups may operate at a scale that is different others. The BCC model hence provides a more realistic representation of the efficient DMUs. Finally, the BCC model can handle situations where there are multiple input variables involved. It provides a framework to evaluate the efficiency of DMUs based on their utilization of multiple inputs simultaneously, which is the case for this study. Figure 2.5 from study [24] shows the linear equation for the BCC input oriented model.

$$\begin{aligned}
 \text{Max } \theta_0 &= \sum_{j=1}^m u_j y_{j0} + u_0 \\
 \text{subject to } &\sum_{i=1}^s v_i x_{i0} = 1 \\
 &\sum_{j=1}^m u_j y_{jk} - \sum_{i=1}^s v_i x_{ik} + u_0 \leq 0 \\
 &v_i \geq 0, u_j \geq 0, u_0 \text{ free in sign}
 \end{aligned}$$

Figure 2.5: The Input-Oriented BCC Model *

Where :

θ_0 is the relative efficiency score for DMU_0 , x_{i0} is the vector of input at DMU_0 ,
 y_{j0} is the vector output at DMU_0 , x_{jk} is the actual value of input i used by DMU_k ,
 y_{jk} is the actual value of output j produced by DMU_k , u is the weights attached to inputs.
 v is the weights attached to outputs.
 A DMU is efficient if $\theta_0 = 1$.

*Source: [22]

*Source: [24]

Chapter 3

Implementation and Results

This chapter delves into the application of the PCA approach on the input and output variables. The retained principal components are thoroughly analyzed to capture the most significant information from the dataset. Second, DEA is applied on four different models, each capturing different dimensions of efficiency. Finally, a peers and excess analysis and ranking are conducted on the selected model to facilitate the Benchmarking , along with interpretations on the business findings.

3.1 PCA Application

In this PCA-DEA analysis, our objective is to reduce the dimensionality of the input and output variables by identifying the most representative principle components. To accomplish this, we initially examined the data to determine the number of significant components. Once the relevant components were identified, we proceeded with interpreting the variables that most contributed to each component. We then assigned labels to each component to facilitate the interpretation of next steps.

3.1.1 Selection of the Principle Components

Given the importance of this step as the foundation for further analysis, two techniques are examined to wisely select the principle components.

1. Scree Plot Analysis:

The scree plot is used to determine the optimal number of principal components to retain in subsequent analysis. It visually displays the eigenvalues and proportion of variance explained by each principal component. It also allows the identification of the elbow point, which indicates the number of principal components that should be retained. In the context of this study, two scree plots for input and output variables are separately needed to conduct the analysis.

- **For the input variables:** Figure 3.1 reveals a clear elbow point at the beginning of PC2, indicating a significant drop in the proportion of variance explained beyond this point. Furthermore, the eigenvalues associated with the principal components also decrease to values below 1. These findings suggest that the first two principal components capture the majority of the variability present in the data. Based on these results, it is reasonable to consider retaining only the first two principal components for further analysis. By doing so, we can represent the most significant patterns within the dataset while reducing dimensionality and complexity. Additionally, the scree plot provides insights into the proportion of variance explained by each principal component. In this study, the first principal component explains approximately 50% of the variance, the second principal component explains around 25%, the third around 22 %, and the fourth principal component has a very weak proportion of variance, accounting for

only 2%.

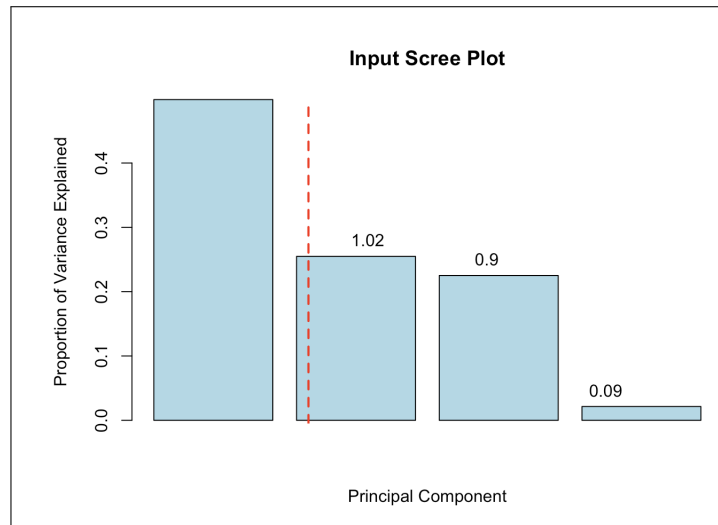


Figure 3.1: The Input Scree Plot *

- **For the output variables:** Figure 3.2 demonstrates a significant decline in the proportion of variance explained beyond the elbow point at the start of PC2. Moreover, the eigenvalues associated with the principal components decrease to values below 1. These observations suggest that the first two principal components capture the majority of the variability present in the data. Therefore, it is reasonable to retain only these two components for further analysis. Besides, the first principal component accounts for approximately 65% of the variance, the second principal component explains around 30%, and the third principal component has a very weak proportion of variance, contributing around 0% to the overall variability.

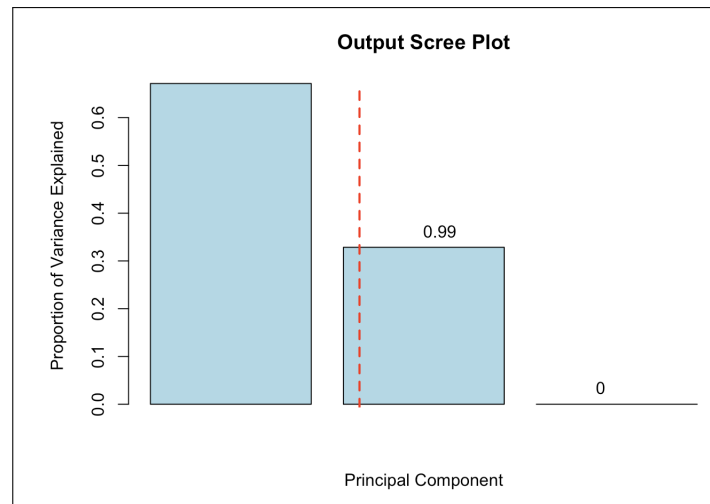


Figure 3.2: The Output Scree Plot *

*Source: by author - R Studio version (2023.03.0+386)

2. Cumulative Variance Explained:

In addition to analyzing the scree plot, another approach proposed by Kaiser in his seminal paper [10] is applied. He suggested to consider retaining components based on a threshold of cumulative variance. Specifically, he recommended retaining components that explain a cumulative variance of at least 70% or higher. This threshold was considered a rule of thumb to determine the number of components that capture a substantial amount of the total variance in the data. By retaining components that account for a substantial amount of variance, researchers can focus on the most meaningful dimensions while reducing the dimensionality of the dataset.

- **For the input variables:** In Figure 3.3, the cumulative proportion of variance explained reaches 75.34% with principle component 2. Adding the third principle component will increase the proportion to 97.85%. Although the third principal component demonstrates importance in capturing additional variance in the data, for the sake of simplicity and dimensionality reduction, we have decided to include only the first two principal components, since they already explain more than 75% of the total variance in the dataset. By retaining these components, we can effectively capture the majority of the underlying patterns and reduce the complexity of the analysis.

Importance of components:				
	PC1	PC2	PC3	PC4
Standard deviation	1.4122	1.0096	0.9489	0.29299
Proportion of Variance	0.4986	0.2548	0.2251	0.02146
Cumulative Proportion	0.4986	0.7534	0.9785	1.00000

Figure 3.3: The Importance of Input components *

- **For the output variables:** In Figure 3.4, the cumulative proportion of variance explained reached 100% by the second component. This means that these two principal components capture all the available information and effectively represent all the underlying patterns within the data. In this scenario, the third principal component becomes redundant for explaining variance since it does not contribute any additional information. Therefore, it is reasonable to limit the inclusion of principal components to just the first two, as they offer a comprehensive representation of the dataset while minimizing dimensionality and complexity.

Importance of components:			
	PC1	PC2	PC3
Standard deviation	1.4193	0.9928	1.301e-16
Proportion of Variance	0.6714	0.3286	0.000e+00
Cumulative Proportion	0.6714	1.0000	1.000e+00

Figure 3.4: The Importance of Output components *

*Source: by author - R Studio version (2023.03.0+386)

3.1.2 Interpretation of the retained Principle Components

In this section, we provide a detailed interpretation of the principal components that have been retained for further analysis. By examining the loadings associated with the variables, we uncover the underlying meaning and patterns captured by each component. Additionally, we analyze the variables that contribute significantly to each component and explore their influence on the formation of the principal components.

1. PCA circle of variables:

Also known as the variable correlation circle or variable factor map, the PCA graph provides insights into the relationships between variables and helps to identify the contributions of inputs and outputs to each principal component. The color of the variables is determined by the squared cosine (\cos^2) values, indicated by the `col.var = cos2` argument. The squared cosine represents the quality of variable representation on the principal components, with values closer to green indicating a better representation. The vectors parallel to a principal component axis contribute more to that component.

- **For the input variables:** In Figure 3.5, the x-axis represents the first principal component, explaining 49.86% of the variation, while the y-axis represents the second principal component, explaining 25.48% of the variation in the dataset. Analyzing the graph, we observe that Counted rejections and Items with complaints are the most strongly represented variables among all others. The Counted rejections and Items with complaints vectors are closest to the x-axis, indicating their strong influence on the first principal component, accounting for around 90% of the overall variation in this component. On the other hand, the Average shipping cost and Average product cost vectors are located near the y-axis, indicating their strong influence on the secondary source of variation. The Average shipping cost vector has a blue color, suggesting a moderate influence on the second component, while the Average product cost vector appears red, indicating a weaker influence on the dimension at hand. This plot also provides insights into the correlation between variables. We observe a small angle between Counted rejections and Items with complaints, indicating a positive correlation. In contrast, the approximate 90° angle between Average shipping cost and Counted rejections implies no correlation between these two variables.

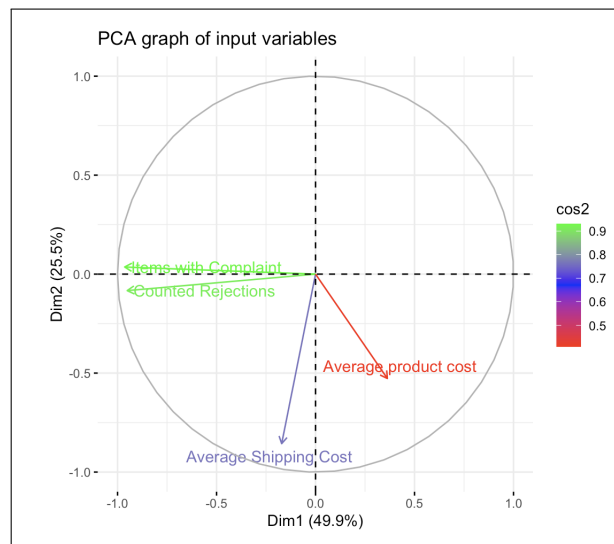


Figure 3.5: PCA graph of Input Variables *

*Source: by author - R Studio version (2023.03.0+386)

- **For the output variables:** In Figure 3.6, the x-axis represents the first principal component, explaining 67.14% of the variation, while the y-axis represents the second principal component, explaining 32.86% of the variation in the dataset. In this graph, the Average gross profit and the Monthly NPS are the closest to the x-axis with blue vectors suggesting a moderate influence on the first component. On the other hand, the Average order count is close the y-axis which indicates its significant influence on the secondary source of variation. Besides, this vector has a blue color, also suggesting an intermediate effect on the second component. We observe a small angle between Average gross profit and Monthly NPS indicating a positive correlation. In contrast, the right angle between Average order count and both other variables implies no correlation between them.

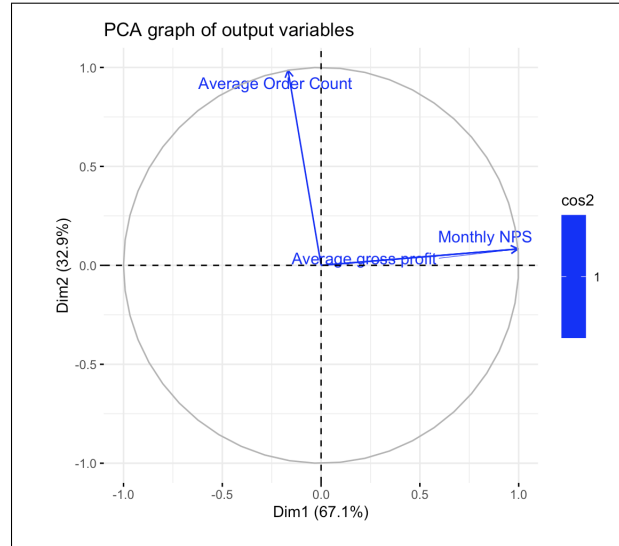


Figure 3.6: PCA graph of Output variables*

However, it is important to note that the correlation circle alone does not provide information about the actual contribution of each variable to the variation explained by each principal component. Therefore, further examination of the contribution plot is necessary to gain insights into variable contributions.

2. The contribution of variables to the principle components:

We further examined the contribution of each variable to the dimensions identified by the principal components. This analysis allows us to quantify the extent to which they contribute to the formation of the principal components.

- **For the input variables:** In Figure 3.7, the Items with complaints and Counted rejections have the highest positive loadings on the first principal component, indicating that they strongly contribute to the formation of the first dimension.

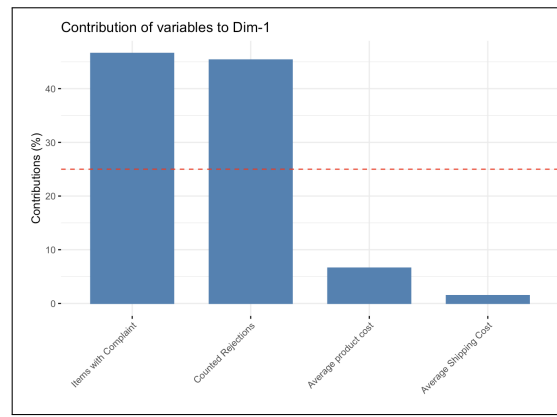


Figure 3.7: Contribution of Input Variables to PC 1 *

On the other hand, in Figure 3.8, the variables Average product cost and Average shipping cost have the highest positive loading on the component, suggesting that they contribute to the second dimension.

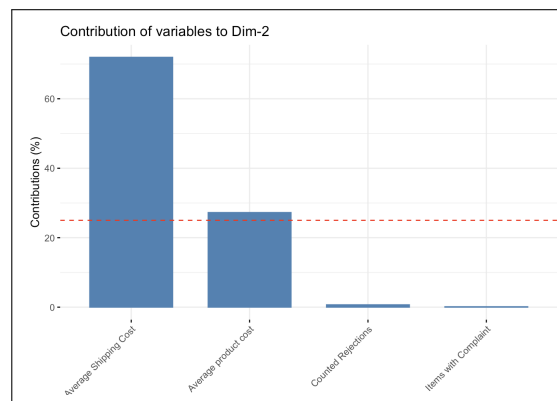


Figure 3.8: Contribution of Input variables to PC 2 *

- **For the output variables:** In Figure 3.9, the Average gross profit and Monthly NPS have the highest positive loadings on the first principal component, indicating that they strongly contribute to the formation of the first dimension.

* Source: by author - R Studio version (2023.03.0+386)

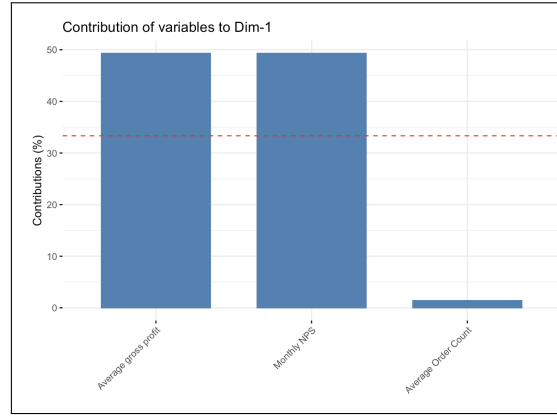


Figure 3.9: Contribution of Output variables to PC 1*

On the other hand, in Figure 3.10, the Average order count has the highest positive loading on the second principal component, indicating that it strongly contributes to the formation of the second dimension.

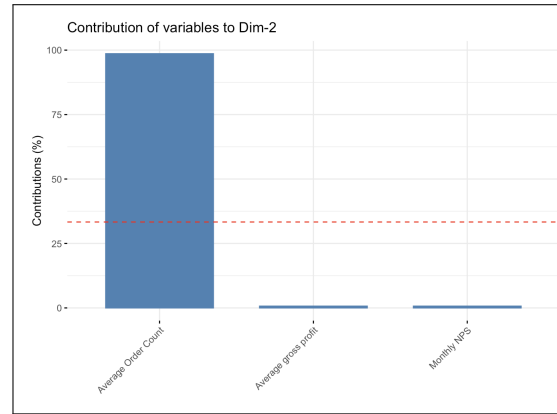


Figure 3.10: Contribution of Output Variables to PC 2*

3.1.3 Descriptive labeling of Principle Components

To enhance the interpretability of the derived components, we assigned meaningful labels that capture the essence of the captured variability in the data.

1. The input variables:

- The first principle component:** As discussed in the previous section, the first principal component is highly influenced by the variables Items with complaints and Counted rejections. Considering these variables as indicators of quality-related issues within the dataset, the label **Quality Control** becomes a suitable choice for this principal component. In fact, Quality Control measures are commonly implemented to ensure customer satisfaction by identifying and rectifying product or service-related issues. Therefore, by labeling the first principal component as **Quality Control**, we emphasize its significance in capturing variations related to quality aspects that directly impact customer satisfaction. This label resonates with stakeholders familiar with quality management practices and provides an intuitive understanding of the underlying patterns captured by the component.

Table 3.1: Label of PC 1 from Input variables

Variables name	Principle component name
Items with complaints Counted rejections	Quality Control

- **The second principle component:** This principal component, which is highly influenced by the variables Average shipping cost and Average product cost, can be properly named as the **Cost Management** component. This label reflects the underlying pattern captured by this component, where higher values indicate higher shipping and product costs. Thus, assigning the label **Cost Management** to this principal component will significantly represents the cost-related aspects within the dataset. This component captures the variations associated with optimizing costs related to shipping and product expenses and it identifies opportunities for cost optimization and efficiency improvements.

Table 3.2: Label of PC 2 from Input variables

Variables name	Principle component name
Average shipping cost Average product cost	Cost Management

2. The Output variables:

- **The first principle component:** The first principal component, which is majorly affected by the variables Average gross profit and Monthly NPS, can be appropriately labeled as the **Business Performance** component. In fact, the Average gross profit is a key indicator of the financial performance and profitability of a business. Including this variable in the component highlights its significance in capturing variations related to the overall financial success of the organization. In addition, Monthly NPS is a widely used metric to measure customer loyalty and satisfaction. Its inclusion in the component signifies the importance of customer perception and satisfaction in driving business performance. Therefore, assigning the label **Business Performance** to this principal component, we can effectively convey its significance in representing the holistic performance of the organization.

Table 3.3: Label of PC 1 from Output variables

Variables name	Principle component name
Average gross profit Monthly NPS	Business Performance

- **The second principle component:** The second principal component is only affected by the variable Average order count. It can be properly named as the **Demand Intensity** component. The variable Average order count directly represents the intensity of demand or the frequency at which customers place orders. Including this variable in the component emphasizes its significance in capturing variations related to the level of demand for products or services. By assigning the label **Demand Intensity** to this principal component, we can effectively understand the factors that drive demand and order frequency.

Table 3.4: Label of PC 2 from Output variables

Variable name	Principle component name
Average order count	Demand intensity

3. Summary of principle components names:

Table 3.5 presents a summary of labels assigned to the principle components along with the variables contributing to these components.

Table 3.5: Summary of Principle Components labels per Input and Output variables

Input/Output rank	Variable name	Principle component name
Input 1 Input 3	Counted rejections Items with complaints	Quality Control
Input 2 Input 4	Average shipping cost Average product cost	Cost Management
Output 2 Output 3	Average gross profit Monthly NPS	Business Performance
Output 1	Average order count	Demand intensity

3.2 PCA-DEA Application

In DEA, the efficient DMUs are identified based on their position relative to the efficiency frontier. The efficiency frontier represents the boundary of optimal performance, and DMUs located on this frontier are considered efficient. In his book [5], Cooper states that different DMUs may have an efficiency score of 1 but that does not necessarily mean the DMU is optimal. A score of 1 indicates that a DMU fully utilizes its inputs to generate outputs relative to the other DMUs in the analysis. It does not guarantee that the DMU operates at an optimal level. In fact, the optimal efficient DMUs are those that are located on the efficiency frontier, which represents the most efficient use of inputs to generate outputs. In this study, our objective is to compare between products that are efficient in terms of minimizing the inputs to generate outputs and other products that are less efficient. For this reason, we will consider all DMUs that have an efficiency score of 1, regardless of their placement on the DEA frontier.

3.2.1 Model 1: PCA-DEA Analysis with no variable reduction

In this model, we incorporated the normalized data from all variables examined in this study, namely Items with complaints, Counted rejections, Average shipping cost, Average product cost, Average gross profit, Average order count, and Monthly NPS. These variables were designated as inputs and outputs in the input-oriented BCC model, which serves for comparing other models that incorporate dimension reduction techniques. By including all variables initially, we establish a baseline for assessing the impact of dimension reduction on model performance and identifying the most influential variables in determining efficiency.

In Figure 3.11, we can visually observe that product groups 14, 21, 24, and 33 are identified as optimal efficient DMUs based on their positioning on the efficiency frontier. However, it is important to note that there may be other DMUs with a score of 1 that are not explicitly included on the frontier line but still hold significance in our analysis. These DMUs, while not located on the frontier line, exhibit a level of performance comparable to the optimal DMUs. Therefore, considering both the DMUs on the frontier and those with a score of 1 allows us to capture a comprehensive understanding of efficiency and holistically evaluate the performance of the DMUs in our analysis.

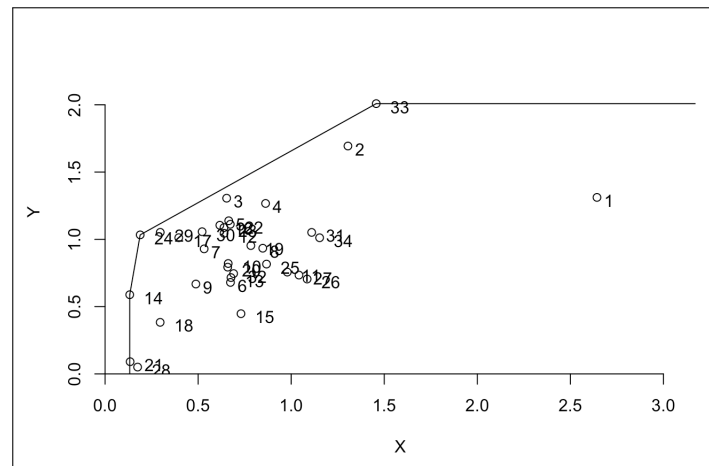


Figure 3.11: The DEA frontier with Model 1: the Basic DEA Model*

In the Table 3.6, the list of all DMUs with an efficiency score of 1 was generated. We can see that 23 out of the 34 product groups achieved an efficiency score of 1, indicating a relatively high number of efficient DMUs in the dataset.

Table 3.6: The DMUs with efficiency score of 1 - Model 1

Product ID (DMU)	Efficiency	Status
1	1	Efficient
2	1	Efficient
3	1	Efficient
8	1	Efficient
10	1	Efficient
12	1	Efficient
14	1	Efficient
15	1	Efficient
16	1	Efficient
17	1	Efficient
18	1	Efficient
20	1	Efficient
21	1	Efficient
22	1	Efficient
24	1	Efficient
26	1	Efficient
27	1	Efficient
28	1	Efficient
29	1	Efficient
30	1	Efficient
31	1	Efficient
33	1	Efficient
34	1	Efficient

Figure 3.12 shows the distribution plot of efficiency scores for Model 1. This plot indicates a right-skewed distribution, meaning that a smaller proportion of DMUs in the dataset are inefficient compared to the efficient ones. Upon closer inspection, it is clear the majority of DMUs are operating at relatively high levels of efficiency. This can be observed by the fact that there is only 1 DMU with a score below 0.5, 7 DMUs with scores between 0.6 and 0.9, 3 DMUs with scores between 0.9 and 1, and 23 DMUS with a

*Source: by author - R Studio version (2023.03.0+386)

score of 1. The clustering of DMUs towards the higher end of the efficiency scale suggests that Model 1 exhibits a high level of overall efficiency.



Figure 3.12: The distribution of efficiency scores of Model 1*

While this may initially suggest a high level of overall efficiency, it also raises the need for dimension reduction techniques to further refine our evaluation of efficiency and obtain more robust and accurate results.

3.2.2 Model 2: PCA-DEA Analysis with Input dimension reduction

In Model 2, we have implemented input dimension reduction to enhance the efficiency analysis. By replacing the original 4 input variables, Items with complaints, Counted rejections, Average shipping cost, and Average product cost, with 2 principal components, we aim to capture the underlying patterns and variability in the data in a concise manner. The decision to select these 2 principal components that were previously labeled Quality Control and Cost Management is based on their ability to account for more than 75% of the variance explained by the model, ensuring that we retain the most significant information while reducing the dimensionality of the input space.

In Figure 3.13, we can visually observe that product groups 1 and 33 are identified as optimal efficient DMUs based on their positioning on the efficiency frontier.

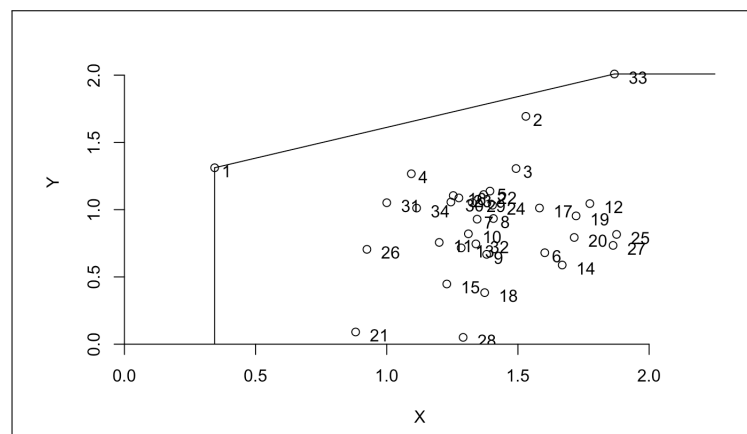


Figure 3.13: The DEA frontier with Model2: Input Dimension Reduction*

*Source: by author - R Studio version (2023.03.0+386)

In Table 3.7, we have generated a list of all DMUs that obtained an efficiency score of 1. We observe that out of the 34 product groups under analysis, 8 product groups are efficient. This result indicates a notable decrease in the number of efficient product groups compared to the previous model. In Model 1, a total of 23 product groups were identified as efficient, while in Model 2, this number reduced to 8. This decrease implies that there are 15 product groups that were previously considered efficient but are no longer classified as such in the current model. The change in the efficiency status of these product groups can be attributed to the input dimension reduction implemented in Model 2. By replacing the original 4 input variables with 2 principal components, we have simplified the input space and potentially altered the efficiency assessment. The dimension reduction process aims to capture the most significant information, specifically 75.34% of the variability, while disregarding certain nuances of the original variables. As a result, some product groups that were previously deemed efficient do no longer meet the efficiency criteria when evaluated based on the reduced input dimensions.

Table 3.7: The DMUs with efficiency score of 1 - Model 2

Product ID (DMU)	Efficiency	Status
1	1	Efficient
2	1	Efficient
16	1	Efficient
17	1	Efficient
21	1	Efficient
31	1	Efficient
33	1	Efficient
34	1	Efficient

Similar to Model 1, the distribution plot of efficiency scores for the second model in Figure 3.14, also exhibits a right-skewed distribution. However, there are notable differences when compared to Model 1. Figure 3.14 reveals that the second model has a larger number of DMUs with scores below 0.5 compared to Model 1. Specifically, there are 12 DMUs falling into this category. Additionally, there are 8 DMUs with scores ranging from 0.5 to 0.8, and 6 DMUs with scores between 0.9 and 1. The increased number of DMUs with scores below 0.5 and the larger variation on the left side of the distribution indicate that Model 2 contains a higher proportion of inefficient units compared to Model 1. This comparison highlights the differences in the overall efficiency levels. Model 1 demonstrates a higher concentration of efficient DMUs, with most of them scoring above 0.9, whereas the second model shows a relatively larger number of inefficient DMUs.

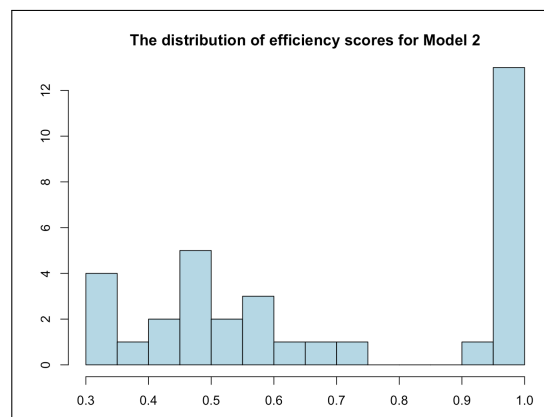


Figure 3.14: The distribution of efficiency scores of Model 2*

*Source: by author - R Studio version (2023.03.0+386)

Table 3.8: The DMUs with efficiency score of 1 - Model 3

Product ID (DMU)	Effeciency	Status
1	1	Efficient
4	1	Efficient
5	1	Efficient
14	1	Efficient
18	1	Efficient
21	1	Efficient
24	1	Efficient
27	1	Efficient
28	1	Efficient
29	1	Efficient
30	1	Efficient
31	1	Efficient
34	1	Efficient

Figure 3.16 shows the distribution of efficiency scores for Model 3 and allows for comparison with the previous models. Similar to the previous models, the distribution plot continues to exhibit a right-skewed pattern, indicating a higher concentration of efficient DMUs compared to inefficient ones. However, there are notable differences in the extent of variation and the number of DMUs falling into different efficiency score ranges. The distribution plot of Model 3 demonstrates more variation compared to Model 1 and Model 2. This implies greater heterogeneity in its performance. It reveals that 11 DMUs have efficiency scores below 0.5, indicating a relatively larger number of inefficient units compared to the first and second models. Additionally, there are 10 DMUs with scores ranging from 0.5 to 0.9, highlighting the presence of a substantial number of moderately efficient units in this range. Importantly, no DMUs have efficiency scores above 0.9, except for the 13 efficient DMUs with a score of 1.

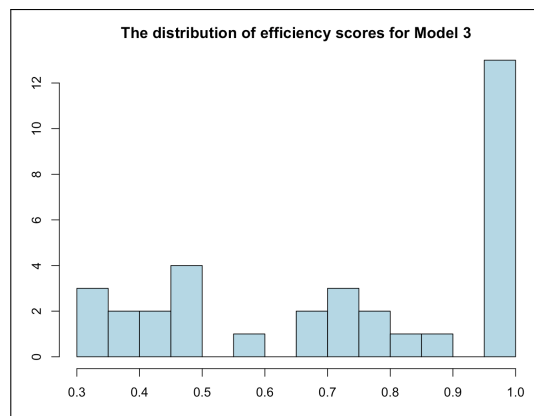


Figure 3.16: The distribution of efficiency scores of Model 3 *

This difference in efficiency between different models suggests that the choice of dimension reduction approach can significantly impact the identification of efficient product groups. Each model provides a unique perspective on efficiency and captures different aspects of the underlying patterns and relationships in the data.

*Source: by author - R Studio version (2023.03.0+386)

3.2.4 Model 4: PCA-DEA Analysis with joint Input-Output dimension reduction

In Model 4, we adopted a joint input-output dimension reduction approach, which involved replacing all original variables with the input and output principal components obtained from the previous analysis. This approach allows us to capture the essential information of both the input and output variables. Specifically, the input variables, namely Items with complaints, Counted rejections, Average shipping cost, and Average product cost, were replaced by the Quality Control and Cost Management variables. These new variables explain 75.34% of the variance in the model. Similarly, the output variables, including Average gross profit, Average order count and Monthly NPS were replaced by the Business performance and Demand intensity variables. These principal components summarize the main patterns and variability in the output space, explaining 100% of the variance in the model. In Figure 3.17, only product groups 1 and 31 are identified as optimal efficient DMUs based on their positioning on the efficiency frontier.

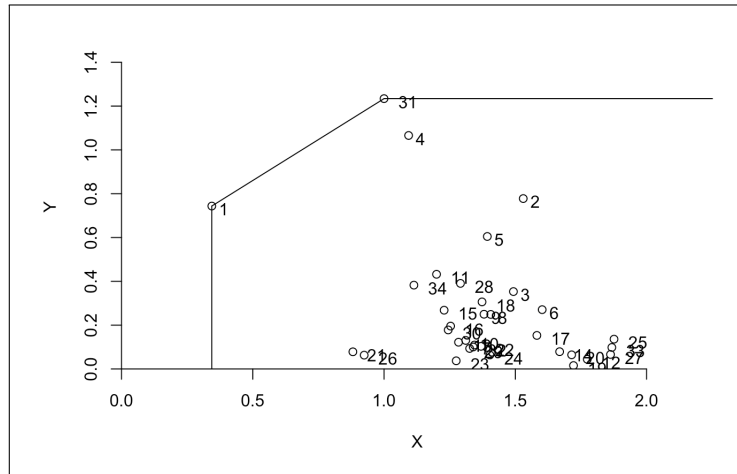


Figure 3.17: The DEA frontier with Model4: joint Input-Output Dimension Reduction*

In Table 3.9, the results reveal that, out of the 34 product groups analyzed, only 4 of them achieved an efficiency score of 1. In comparison across the different models, we observe some overlap. Specifically, products 1, 21 and 31 are common to both Model 2, 3 and 4, indicating their consistent high performance across different dimension reduction approaches. Additionally, product 4 is only common in Model 3 and 4, suggesting its good performance regardless of the additional dimension reduction applied.

Table 3.9: The DMUs with efficiency score of 1 - Model 4

Product ID (DMU)	Efficiency	Status
1	1	Efficient
4	1	Efficient
21	1	Efficient
31	1	Efficient

Figure 3.18 shows the distribution of efficiency scores in Model 4. To start with, it is visible that this plot is different from the previous plots. Model 4 has a left-skewed distribution, indicating that the majority of the DMUs are concentrated towards the lower end of the efficiency scale. In fact, the illustration shows 5 DMUs that have scores below 0.3, indicating a notable proportion of highly inefficient units. Additionally, there are 7 DMUs with scores ranging from 0.3 to 0.4, followed by 13 DMUs in the range of 0.4 to 0.5. This suggests a relatively larger number of moderately inefficient units in these ranges. Moreover, the distribution demonstrates a smaller group of DMUs with scores between 0.5 and 0.7 consisting of 4 DMUs signifying some improvement but still falling short of achieving high efficiency. Interestingly, there

*Source: by author - R Studio version (2023.03.0+386)

is only one DMU with a score above 0.9, indicating a single unit that has achieved exceptional efficiency. Comparing this distribution with the previous models, Model 4 stands out as having a larger number of highly inefficient DMUs and a smaller proportion of highly efficient ones. This suggests a considerable room for improvement among the majority of the DMUs in Model 4.



Figure 3.18: The distribution of efficiency scores of Model 4 *

Overall, the results of the joint input-output dimension reduction in Model 4 emphasize the importance of considering both input and output variables in assessing efficiency. The common efficient product groups across models demonstrate their consistent high performance and may serve as benchmarks for other product groups to strive for in terms of efficiency. Hence, this model is selected for further analysis.

3.3 Benchmarking and ranking

After selecting Model 4 as the Benchmarking model, the next step involves conducting a peers analysis and determining peers weights for each DMU. A comparative ranking is then conducted based on the efficiency scores and peers weights. Additionally, an excess analysis is performed to identify performance gaps between the DMUs and the benchmark.

3.3.1 Peers Analysis

Peers analysis is a widely used technique that compares the performance of entities within a similar context or industry. In this study, it is applied to compare efficient and inefficient product groups, more specifically to compare product groups with an efficiency score of 1 with other product groups that have a lower efficiency scores, providing insights for decision-making. The peers weights derived from the analysis serve as benchmarks for the inefficient DMUs, offering guidance on how to improve their performance. Among the DMUs assessed, four have been identified as efficient: DMUs 1, 4, 21, and 31. These efficient DMUs serve as reference points to the inefficient DMUs on how to enhance their performance. Table 3.10 shows the reference set and corresponding weights of the inefficient DMUs. Since only 4 product groups which are product 1, 4, 21 and 31 are efficient in the selected model, they act as peers set for remaining product groups. For instance, Product Group 3 is identified as an inefficient DMU. In the peers analysis, we have determined that Product Group 3 has three reference sets: Product Group 1, 21, and 31. These reference sets serve as benchmarks for Product Group 3 to enhance its performance.

*Source: by author - R Studio version (2023.03.0+386)

Table 3.10: The peers weights

Product ID (DMU)	1	4	21	31
1	1.000000000	0.0000000	0.0000000	0.000000000
2	0.601637009	0.2562301	0.0000000	0.142132937
3	0.703787374	0.0000000	0.2718952	0.024317443
4	0.000000000	1.0000000	0.0000000	0.000000000
5	0.716876791	0.0000000	0.2246581	0.058465134
6	0.710810859	0.0000000	0.1779452	0.111243908
7	0.574267011	0.0000000	0.4173566	0.008376376
8	0.652288771	0.0000000	0.3123519	0.035359334
9	0.613442379	0.0000000	0.2693476	0.117210038
10	0.560301987	0.0000000	0.4029443	0.036753747
11	0.494293162	0.0000000	0.2746216	0.231085261
12	0.704052971	0.0000000	0.2699967	0.025950378
13	0.530693728	0.0000000	0.4418229	0.027483350
14	0.687519654	0.0000000	0.2862151	0.026265272
15	0.584191351	0.0000000	0.3673955	0.048413184
16	0.501932174	0.0000000	0.4275716	0.070496193
17	0.691363153	0.0000000	0.2986555	0.009981371
18	0.617786463	0.0000000	0.3146983	0.067515188
19	0.680064076	0.0000000	0.3199359	0.000000000
20	0.677747496	0.0000000	0.2880177	0.034234838
21	0.000000000	0.0000000	1.0000000	0.000000000
22	0.561628490	0.0000000	0.3919963	0.046375207
23	0.373412001	0.0000000	0.6265880	0.000000000
24	0.583120268	0.0000000	0.4086506	0.008229104
25	0.745116797	0.0000000	0.1551831	0.099700097
26	0.006375141	0.0000000	0.9936249	0.000000000
27	0.716999595	0.0000000	0.2345113	0.048489116
28	0.563525913	0.0000000	0.2669031	0.169570941
29	0.531451228	0.0000000	0.4121438	0.056404999
30	0.468301195	0.0000000	0.4123791	0.119319706
31	0.000000000	0.0000000	0.0000000	1.000000000
32	0.549238042	0.0000000	0.4320308	0.018731137
33	0.720930104	0.0000000	0.2517079	0.027361961
34	0.266157353	0.0000000	0.4465007	0.287341965

The relationship of peers weights with the reference sets indicate the extent of influence a reference set has in affecting the performance improvement of the DMU and how likely will it serve as its benchmark. Table 3.11 reveals the summary of these relationships indicating the highest peers weight of each DMU. In the same example of PG 3, PG 1 has the highest peers weight of 0.703787374, suggesting that it plays a significant role in guiding and influencing the strategies adopted by PG 3 to improve its performance. Besides, from Table 3.11 we can compare the impact of each product group on the rest. We can see that among 34 products, 25 of them have the highest peers weight from Product Group 1. This suggests this Product Group is a prominent benchmark for the majority of the products in the analysis. The performance and strategies of product group 1 are considered highly influential and serve as a valuable reference for improving the performance of other products.

Table 3.11: Summary of peers relationship

Product ID (DMU)	Peers Count	Reference Set	Highest peers weight
1	1	1	1
2	3	1,4, 31	1
3	3	1, 21, 31	1
4	1	4	4
5	3	1,21,31	1
6	3	1, 21, 31	1
7	3	1, 21, 31	1
8	3	1, 21, 31	1
9	3	1, 21, 31	1
10	3	1, 21, 31	1
11	3	1, 21, 31	1
12	3	1, 21, 31,	1
13	3	1, 21, 31	1
14	3	1, 21, 31	1
15	3	1, 21, 31	1
16	3	1, 21, 31	1
17	3	1, 21, 31	1
18	3	1, 21, 31	1
19	3	1, 21, 31	1
20	3	1, 21, 31	1
21	1	21	21
22	3	1, 21, 31	1
23	3	1, 21, 31	21
24	3	1, 21, 31	1
25	3	1, 21, 31	1
26	3	1, 21	21
27	3	1, 21, 31	1
28	3	1, 21, 31	1
29	3	1, 21, 31	1
30	3	1, 21, 31	1
31	1	31	31
32	3	1, 21, 31	1
33	3	1, 21, 31	1
34	3	1, 21, 31	21

To summarize, in the context of peers analysis, we can refer to the assigned peers weights of each DMU illustrated in Table 3.10 to guide the decision-making process. Each DMU is assigned one or more peers weights, which represent the relative importance or influence of the corresponding peers. By examining these peers weights, we can identify the strategies and practices employed by the peers. In cases where there are multiple peers, decision makers may choose to look at the strategies implemented by the peers with the highest peers weights, considering them as potential sources of successful practices to be adopted. This approach leverages the knowledge and experiences of high-performing peers to enhance the strategies and performance of inefficient DMUs.

3.3.2 Ranking of the DMUs

After conducting the peers analysis and determining the peers weights for each DMU, we proceeded to rank the products based on their efficiency scores and peers counts.

1. Based on the efficiency scores:

In DEA, efficiency scores are the primary measure to evaluate the performance of individual DMUs by considering their output-to-input ratio. They assess how effectively a DMU utilizes its resources to generate outputs. While efficiency scores provide a measure of individual performance, they may not offer a comprehensive understanding of how DMUs compare to each other in terms of their relative performance. Looking at Table 3.12, we can observe the initial ranking of the DMUs. The ranking starts with the four efficient DMUs, namely 1, 4, 21, and 31, indicating their superior performance compared to others. In terms of relative performance, the closest DMU to the efficient group is PG 26, ranked fifth. This suggests that PG 26 exhibits a performance that is relatively closer to the efficient DMUs compared to others. Following closely is PG 34 at rank 6. On the bottom of 3.12, we find the least efficient DMU, which is PG 25, ranked 34th. This indicates that PG 25 demonstrates the lowest performance among all the DMUs in the dataset. It is followed by DMU 33 at rank 33. The rankings provide a hierarchical order of the DMUs based on their efficiency scores, offering insights into their relative performance within the dataset.

Table 3.12: Ranking of DMUs based on Efficiency scores

Product ID (DMU)	Efficiency score	Rank
1	1.0000000	1
4	1.0000000	2
21	1.0000000	3
31	1.0000000	4
26	0.9497194	5
34	0.6935117	6
11	0.5359343	7
23	0.5338813	8
30	0.5173772	9
16	0.4945957	10
13	0.4669387	11
15	0.4663935	12
28	0.4636416	13
29	0.4542843	14
10	0.4458532	15
32	0.4391914	16
22	0.4272199	17
7	0.4267272	18
2	0.4112271	19
9	0.4096064	20
18	0.4058591	21
24	0.4054987	22
8	0.3803588	23
5	0.3610169	24
3	0.3389601	25
17	0.3229626	26
6	0.3198785	27
14	0.3085913	28
20	0.3039690	29
19	0.2996400	30
12	0.2852426	31
27	0.2693887	32
33	0.2661858	33
25	0.2626518	34

2. Based on peers weight:

By considering the peers weight, the ranking of DMUs becomes more refined and meaningful. DMUs that have a higher peers weight are considered more influential in driving efficiency. In Table 3.13, we can observe that DMU 26 maintained its rank of 5, followed by PG 25 at rank 6. Interestingly, PG 25 was ranked as the least efficient based on the efficiency scores in Table 3.12. However, this apparent contradiction can be explained by the fact that the peers weight takes into account the performance of the DMU in comparison to its peers. In the case of DMU 25, despite having a low efficiency score of 0.2626518, it possesses a remarkably high peers weight of 0.745116797. This implies that while the DMU itself may not be highly efficient, it demonstrates relatively better performance when compared to its peers group. The peers weight captures the collective performance of the DMU and its peers, providing a more comprehensive assessment of relative efficiency.

Table 3.13: Ranking of DMUs based on Peers Weights

Product ID (DMU)	peers Weight	Rank
1	1.000000000	1
4	1.000000000	1
21	1.000000000	1
31	1.000000000	1
26	0.9936249	5
25	0.745116797	6
33	0.720930104	7
27	0.716999595	8
5	0.716876791	9
6	0.710810859	10
12	0.704052971	11
3	0.703787374	12
17	0.691363153	13
14	0.687519654	14
19	0.680064076	15
20	0.677747496	16
8	0.652288771	17
23	0.6265880	18
18	0.617786463	19
9	0.613442379	20
2	0.601637009	21
15	0.584191351	22
24	0.583120268	23
7	0.574267011	24
28	0.563525913	25
22	0.561628490	26
10	0.560301987	27
32	0.549238042	28
29	0.531451228	29
13	0.530693728	30
16	0.501932174	31
11	0.494293162	32
30	0.468301195	33
34	0.4465007	34

3.3.3 Excess analysis

The excess values obtained for each DMU reflect their level of relative efficiency in utilizing the input resources. A lower excess value indicates a higher level of efficiency, meaning that the DMU is utilizing its inputs more effectively compared to other DMUs. Conversely, a higher excess value suggests a greater degree of inefficiency, implying that the DMU could improve its input utilization to become more efficient. The analysis of the excess values in Table 3.14 provides insights into areas of improvement for each product group. Viewing the input orientation nature of the model, it only examines the excess values for our input variables. The mean excess for each principal component is calculated, revealing a higher value for PC1 (0.43287814) compared to PC2 (0.325182139). This indicates that, on average, the excess is primarily attributed to Quality Control rather than Cost Management. It suggests that Quality Control

is more likely to be the main factor contributing to inefficiencies in the DMUs, highlighting a greater potential for improvement in this area compared to Cost Management.

Table 3.14: Excess values per DMU

Product ID (DMU)	Excess from Quality Control (PC1)	Excess from Cost Management (PC2)
1	0.00000000	0.00000000
2	0.42797146	0.473001893
3	0.49297782	0.493844401
4	0.00000000	0.00000000
5	0.43760148	0.452737545
6	0.55443785	0.55443785
7	0.48237998	0.288365201
8	0.48517714	0.386381765
9	0.49516067	0.320054411
10	0.46650205	0.260081812
11	0.39990914	0.156974201
12	0.63351534	0.634568066
13	0.45519894	0.229084183
14	0.59769264	0.556145333
15	0.40859131	0.247143178
16	0.43916449	0.194346661
17	0.54700277	0.524194095
18	0.48594059	0.329972109
19	0.62835372	0.577416123
20	0.63255483	0.560849817
21	0.00000000	0.00000000
22	0.50378514	0.280561277
23	0.45967787	0.134589530
24	0.51549007	0.318639210
25	0.64595383	0.737348232
26	0.04420258	0.002275514
27	0.66593721	0.694943274
28	0.45561155	0.236897328
29	0.48377062	0.239904683
30	0.43453829	0.166042385
31	0.00000000	0.00000000
32	0.48746663	0.263866359
33	0.65849703	0.712061542
34	0.29279359	0.048556942
Mean value	0.43287814	0.325182139

3.4 Business Findings

The interpretation of the findings should take into account the peers weights, ranking, and excess values, as they provide important information about the relative performance and potential for improvement of the DMUs.

- By analyzing Table 3.10, decision makers can identify the peers with the highest weight for a specific DMU, indicating that it can majorly affect its efficiency improvement and serve as its benchmark.
- Table 3.13, which displays the peers ranking, helps decision makers understand how each DMU compares to its peers in terms of efficiency. DMUs with higher rankings are relatively more efficient, and require less efforts to enhance their performance, while those with lower rankings have more room for improvements.
- Table 3.14, on the other hand, provides insights into the excess values for specific variables, indicating areas where improvements can be made. DMUs with higher excess values, are less efficient, have greater performance gaps and therefore have more potential for improvement in those specific areas.

The inverse relationship between peers ranking and excess values underscores the importance of Benchmarking against more efficient peers and targeting areas with higher excess values to drive performance improvements and close the performance gaps. Let's take a closer look at DMU 26 as an example. In terms of Quality Control, DMU 26 exhibits an excess value of 0.04420258, indicating that there is room for improvement in reducing customer complaints and order rejections. This suggests that measures can be taken to enhance the overall Quality Control process and minimize issues related to customer satisfaction. Similarly, in Cost Management, DMU 26 has a small excess value of 0.002275514. This implies that actions can be implemented to optimize both product and shipping costs, leading to potential cost savings and improved efficiency in resource allocation. The relatively low excess values in both Quality Control and Cost Management contribute to the higher ranking of DMU 26 based on peers weight shown in Table 3.13. This suggests that, despite its current level of efficiency, DMU 26 has a great potential to further enhance its performance. In order to achieve these improvements, decision makers can look to the applied strategies of DMU 21, which holds the highest peers weight of 0.9936249 according to Table 3.10. The high peers weight assigned to DMU 21 signifies that its strategies have been successful in achieving efficiency and can serve as a valuable example for DMU 26.

A second example that can be examined is DMU 7. In Table 3.13, DMU 7 is ranked 24th based on peers weights, indicating its relative performance compared to other DMUs. By referring to Table 3.14, we can observe that DMU 7 has excess values of 0.48237998 in Quality Control and 0.288365201 in Cost Management. These excess values represent the gaps between DMU 7's actual performance and the benchmark established by the model. Turning our attention to Table 3.10, we find that DMU 1 has the highest peers weight of 0.574267011 for DMU 7. This indicates that DMU 1 is considered a significant reference point peers for DMU 7 in terms of performance evaluation. However, it is worth noting that DMU 21 also possesses a relatively close peers weight of 0.4173566 for DMU 7. This suggests that DMU 21 strategies and practices may also hold value and could be worth considering as potential improvement measures for DMU 7.

Overall, this analysis provides decision makers with insights to correctly prioritize improvement efforts and consider the strategies employed by the benchmarks as potential sources for enhancing the other products' performance in Quality Control and Cost Management.

Chapter 4

General Conclusion

4.1 Discussion of the findings

In this study, our primary objective was to assess the efficiency of product groups within Vistaprint using a variety of metrics all at once. To achieve this, an implementation of PCA-DEA was conducted. This combined model allows to establish a comprehensive ranking system that takes into account multiple performance metrics, providing a holistic assessment of product performance. The efficiency of the product groups was evaluated using the variable return to scale input orientation, employing both the traditional DEA model and the PCA-DEA model. The findings of the study are as follows [3] :

- The PCA allows for a reduction of the original 4 variables to 2 principal components, namely Quality Control and Cost Management, that capture more than 75% of the total variance in the input data. Outputs are reduced from 3 variables to 2 principle components, namely Business Performance and Demand Intensity that explain 100% of the variance in the output data.
- Four different models were considered to evaluate the efficiency of the product groups. Model 1 represented the basic DEA model, considering all inputs and outputs without dimension reduction. This model identified 23 efficient product groups out of the initial 34. Model 2 focused on reducing the dimension of the inputs to 2 principal components, resulting in a decrease in the number of efficient product groups to 8, providing a more detailed evaluation. Similarly, Model 3 aimed at reducing the dimension of the outputs to 2 principal components, leading to the identification of 13 efficient product groups. Finally, Model 4 incorporated the joint reduction of both inputs and outputs to 2 principal components each, resulting in the identification of 4 efficient product groups and further refining the efficiency evaluation process.
- In the traditional DEA model, out of the 34 product groups, 23 were found to be efficient. However, when utilizing the PCA-DEA Model with the joint inputs and outputs reduction, only 4 product groups were identified as efficient. This highlights the advantage of using the PCA-DEA model when dealing with a large number of variables, as it provides a more effective distinction between efficient and inefficient product groups.
- Three product groups, specifically DMU number 1, 21, and 31, were identified as efficient in both the traditional DEA model and the reduced models. Notably, product group 4 appeared to be efficient only after conducting output dimension reduction. Additionally, the 20 other product groups classified as efficient in the traditional DEA model were misclassified in the PCA-DEA model.
- DMU 1 was ranked as the top peers for 28 inefficient product groups, followed by DMU 21, which ranked second with 4 peers.
- Based on peers weight, DMU 34 was ranked last (rank 34) with a peers weight of 0.4465007, while DMU 30 preceded it with a peers weight of 0.468301195.

- The excess analysis reveals that the excess mean in Quality Control (0.43287814) is higher than the excess mean in Cost Management (0.325182139). This indicates that there is a greater scope for improvement in reducing customer complaints and order rejections compared to reducing shipping and product costs.
- The DMUs are relatively less efficient in managing and maintaining Quality Control processes. Efforts should be prioritized towards improving the Quality Control aspect by improving the customer service and order management strategies.

Overall, this study successfully applied PCA-DEA techniques to enhance the efficiency evaluation and Benchmarking of product groups within Vistaprint. The ranking of the product groups and excess analysis helps decision makers to identify strategies to optimize efficiency and performance. Furthermore, these findings can serve as benchmarks for future product introductions, aiding in the continuous improvement of product group efficiency.

4.2 Limitations

During the course of this research, several limitations were encountered that constrained the extent of the analysis and its potential outcomes.

- **Technical difficulties:** One of the limitations encountered in this study was related to the compatibility of DEA packages with newer versions of Python and R. Specifically, the study faced challenges in using certain libraries such as PyDEA in Python and DEA in R due to compatibility issues with the newer versions of these software. As a result, the implementation of advanced DEA techniques was not possible. This limitation restricted the scope of the study and may have affected the depth and breadth of the findings. This emphasizes the importance of regularly updating and maintaining DEA packages to ensure seamless integration with evolving programming languages and environments. Future studies should consider the compatibility aspect and seek alternative solutions or updated versions of DEA packages to overcome this limitation.
- **Sample Size:** The study focused on evaluating the efficiency of 34 product groups within the signage category at Vistaprint. These 34 DMUs were selected based on the availability of data and their relevance to the research objectives. It is important to note that due to confidentiality reasons, data from other categories could not be included, which resulted in a limited sample size. While a larger sample size would have provided a more comprehensive perspective on efficiency across a wider range of DMUs, efforts were made to ensure that the selected units represented the diversity within the signage category. However, caution should be exercised when generalizing the findings beyond the specific sample used in this study, as the results may not directly apply to other categories or organizations. The scope of this study was limited to the signage category at Vistaprint, and the findings should be interpreted within this context.
- **Simplification of analysis and temporal dynamics:** Averaging of the period in each variable was employed to streamline the analysis and facilitate the calculation of efficiency scores using DEA. However, this may result in overlooking the underlying variations and temporal dynamics within the data. This simplification assumes a stable efficiency throughout the specified period and may mask important temporal patterns or fluctuations. While this approach simplified the analysis, future studies should consider collecting data at more frequent intervals or exploring alternative methods to capture the temporal dynamics of variables, allowing for a more nuanced assessment of efficiency. Researchers should carefully consider the trade-off between simplicity and the potential loss of temporal information in DEA modeling.
- **Potential loss of interpretability:** When reducing the dimensions of the input-output data, the resulting principal components may not directly correspond to the original variables. This can make

it challenging to interpret the underlying meaning of the components and may limit the scope of interpretation. While the reduced dimensions can still provide insights into the factors driving efficiency, the interpretation becomes more abstract and detached from the original variables. Besides, the Benchmarking process based on dimension-reduced inputs and outputs may not fully capture the complexity of the original data. The exclusion of variables during dimension reduction can result in an incomplete Benchmarking analysis, as important variables that affect efficiency may not be considered. This limitation should be carefully considered when interpreting Benchmarking results and making decisions based on them.

4.3 Recommendations

Based on the findings and analysis conducted in this research, several recommendations can be made to enhance the efficiency and performance of the business:

- **Improve Quality Control:** Address the identified variables of interest, specifically the number of items with complaints and counted rejections. Given the higher excess value mean in this area, prioritize the improvement of this component in future decision-making. This can be achieved by conducting a thorough analysis of customer feedback and complaints to further understand the specific issues and concerns raised by customers. By identifying common patterns and recurring issues, appropriate actions can be taken to resolve these concerns and improve customer satisfaction. Besides, ensuring a reliable and efficient product setup process is essential to avoid order issues. This includes streamlining the order management system, improving coordination between different departments involved in the order fulfillment process, and implementing Quality Control measures at different stages.
- **Optimize Cost Management:** Evaluate both product and shipping costs. While these costs may not always be directly under the company's control, try to minimize expenses without compromising product quality through processes streamline and shipping logistic management. In case it is determined that significant cost reductions in product and shipping expenses are not feasible, it is advisable to focus on the first component mentioned in the recommendation. Place emphasis on improving internal processes and operational efficiency to mitigate any cost-related challenges.
- **Enhance peers Collaboration:** Leverage insights gained from the peers weight analysis to promote collaboration and knowledge-sharing among peers. Focus on Benchmarking from the highest peers weight, as this signifies the most influential and successful peer. In situations where there are multiple reference points available, it is essential to carefully evaluate and select the most appropriate one for Benchmarking . Consider factors such as similarity in cost and operational context.
- **Leverage peers rankings for product expansion:** Use the peers ranking to influence future product decisions and expand offerings within the efficient product groups. Encourage targeted expansion of product offerings in areas that have demonstrated efficiency, increasing the likelihood of success.

These recommendations aim to improve the business's efficiency based on the research findings. Implementing these recommendations can lead to performance enhancements and foster sustainable long-term growth.

Bibliography

- [1] What is dea? history of dea from its developer, william w cooper. <https://deazone.com/en/resources/tutorial/issues-in-dea>.
- [2] Annual report on form 10-k for fiscal year 2022, 2022.
- [3] D. Annapoorni¹ and V. Prakash². Measuring the performance efficiency of hospitals: Pca – dea combined model approach. 2016.
- [4] A. Charnes, W.W. Cooper, and E. Rhodes. *Measuring the efficiency of decision making units*. 2022.
- [5] William W. Cooper. *Introduction to Data Envelopment Analysis and Its Uses: With DEA-Solver Software and References*. 2006.
- [6] V. Prakash. D. Annapoorni¹. Measuring the performance efficiency of hospitals: Pca – dea combined model approach. *Indian Journal of Science and Technology*, 2016.
- [7] John M. Gleason Darold Barnum. Bias and precision in the dea two-stage method. 2008.
- [8] C.A. Knox Lovell Dennis J. Aigner and Peter Schmidt. *On the origins of Aigner, Lovell and Schmidt, 1977, and the development of stochastic frontier analysis*. 1977.
- [9] Joakim Widen Dennis van der Meern Dazhi Yang and Joakim Munkhammar. *Energy-environmental efficiency of clean energy in China: Integrated analysis with regional green growth*. 2020.
- [10] Kaiser H. F. *The Application of Electronic Computers to Factor Analysis. Educational and Psychological Measurement*. 1960.
- [11] María Margallo a Rubén Aldaco a Ian Vázquez-Rowe b Jara Laso a, Jorge Cristóbal a. *Assessing Progress Towards Sustainability*. Sciencedirect.com, 2022.
- [12] Alex Manzoni. A new approach to performance measurement using data envelopment analysis: Implications for organisation behaviour, corporate governance and supply chain management. *Victoria University*, 2007.
- [13] Ilker Murat and Kurtaranl Ahmet. *Evaluating the Relative Efficiency of Commercial Banks in Turkey: An Integrated AHP/DEA Approach*. Karadeniz Technical University, Trabzon, Turkey, 2022.
- [14] Trung Thanh Nguyen. Farm production efficiency and natural forest extraction in cambodia. *SSRN Electronic Journal*.
- [15] Trung Thanh Nguyen. Farm production efficiency and natural forest extraction in cambodia. *SSRN Electronic Journal*, 2018.
- [16] Daniel Owen. *Optimisation of memory reference patterns in dynamic binary translation systems*. The University of Manchester (United Kingdom), 2017.

-
- [17] Kirkley J.E. Gréboval D. Morrison-Paul C.J. Pascoe, S. Measuring and assessing capacity in fisheries 2. issues and methods. 2003.
 - [18] R. Põldaru and J. Roots. A pca–dea approach to measure the quality of life in estonian counties. 2014.
 - [19] Santiago Leguey-Galán Rocío Guede-Cid, Leticia Rodas-Alfaya. Innovation efficiency in the spanish service sectors, and open innovation. *Journal of Open Innovation Technology Market and Complexity*.
 - [20] DJohn Ruggiero. A comparison of dea and the stochastic frontier model using panel data. 2007.
 - [21] Ilhem Saadaoui. Performance analysis application using dea: case study of a coffee shop chain. 2021.
 - [22] Michael Schultz Petros Stratis Thomas Standfuß, Frank Fichert. Efficiency losses through fragmentation? scale effects in european ans provision. 2019.
 - [23] Christopher J. O'Donnell George E. Battese Timothy J. Coelli, D.S. Prasada Rao. *Data and Measurement Issues. In: An Introduction to Efficiency and Productivity Analysis*. Springer, Boston, MA, 2005.
 - [24] Wan Rosmanira Ismail Wan Malissa Wan Mohd Aminuddin. Integrated simulation and data envelopment analysis models in emergency department. 2016.
 - [25] Wenqi Yu. Xiao Shi. Advanced data analytics techniques for risk-based engineering problems”. 2021.
 - [26] Zijiang Yang. A two-stage dea model to evaluate the overall performance of canadian life and health insurance companies. 2006.
 - [27] Dragan Vojinović Eldina Huskanović Miomir Stanković andDragan Pamučar. Željko Stević, Smiljka Miškić. Development of a model for evaluating the efficiency of transport companies: Pca–dea–mcdm model. 2022.

Appendices

I - Data Cleaning and Pre-processing

1- Fill the missing values

- Input2: Shipping cost

```
import pandas as pd

Input2 = pd.read_excel('Shipping List Unit Price per product group.xlsx')
Input2 = Input2.fillna(method='ffill').fillna(method='bfill')

Input2.iloc[:, 1:] = Input2.iloc[:, 1:].apply(pd.to_numeric)
Input2 = Input2.interpolate(method='linear')
Input2.to_excel('Filled Input2.xlsx', index=False)
```

- Input4: Product cost

```
import pandas as pd

Input4 = pd.read_excel('Product cost by product group.xlsx')
Input4 = Input4.fillna(method='ffill').fillna(method='bfill')

Input4.iloc[:, 1:] = Input4.iloc[:, 1:].apply(pd.to_numeric)
Input4 = Input4.interpolate(method='linear')
Input4.to_excel('Filled Input4.xlsx', index=False)
```

- Output 1: Order count

```
import pandas as pd

Output1 = pd.read_excel('Order Count per Product group.xlsx')
Output1 = Output1.fillna(method='ffill').fillna(method='bfill')

Output1.iloc[:, 1:] = Output1.iloc[:, 1:].apply(pd.to_numeric)
Output1 = Output1.interpolate(method='linear')
Output1.to_excel('Filled Output1.xlsx', index=False)
```

- Output 2: Gross profit

```
import pandas as pd

Output2 = pd.read_excel('Gross profit per product group.xlsx')
Output2 = Output2.fillna(method='ffill').fillna(method='bfill')
```

```
Output2.iloc[:, 1:] = Output2.iloc[:, 1:].apply(pd.to_numeric)
Output2 = Output2.interpolate(method='linear')
Output2.to_excel('Filled Output2.xlsx', index=False)
```

- Output 3: Net promoter score

```
import pandas as pd

Output3 = pd.read_excel('NPS by Product Group.xlsx')
Output3 = Output3.fillna(method='ffill').fillna(method='bfill')

Output3.iloc[:, 1:] = Output3.iloc[:, 1:].apply(pd.to_numeric)
Output3 = Output3.interpolate(method='linear')
Output3.to_excel('Filled Output3.xlsx', index=False)
```

2- Summarize the time period

- Input1: Counted rejections

```
Input1 = pd.read_excel('Order rejections per product group.xlsx')
Input1 = Input1['Product ID'].value_counts()

Input1 = pd.DataFrame({'Product ID': Input1.index, 'Counted Rejections':
Input1.values})
Input1.to_excel('Input1.xlsx', engine='openpyxl', index=False)
```

```
print(Input1.head())
```

```
##      Product ID  Counted Rejections
## 0             1.0                1870
## 1             4.0                 360
## 2             2.0                 318
## 3             5.0                 158
## 4             3.0                 155
```

- Input2: Average shipping cost

```
Input2 = pd.read_excel('Filled Input2.xlsx')
Input2['Average Shipping Cost'] = Input2.iloc[:, 1:].mean(axis=1)

Input2 = Input2[['Product ID', 'Average Shipping Cost']].copy()
Input2.to_excel('Input2.xlsx', index=False)
print(Input2.head())
```

```
##      Product ID  Average Shipping Cost
## 0             1                4.068735
## 1             2                3.217055
## 2             3                1.762321
## 3             4                2.024766
## 4             5                2.036675
```

- Input3: Average Items with complaints

```
Input3 = pd.read_excel('Complaint Rate by Product Group Weekly.xlsx')
Input3 = Input3.loc[:, ~Input3.columns.str.contains('Complaint Rate|Complaint
```

```
Action Amount|Complaint Items VGP']])
Input3 = Input3.dropna(subset=['Items with Complaint'])
Input3 = Input3.groupby('Product ID')['Items with Complaint'].mean().reset_index()
Input3.to_excel('Input3.xlsx', index=False)
```

```
print(Input3.head())
```

```
##      Product ID  Items with Complaint
## 0           1.0             160.0
## 1           2.0             80.0
## 2           3.0             50.0
## 3           4.0             49.0
## 4           5.0             42.0
```

- Input4: Average product cost

```
Input4 = pd.read_excel('Filled Input4.xlsx')
Input4['Average product cost'] = Input4.iloc[:, 1:].mean(axis=1)
```

```
Input4 = Input4[['Product ID', 'Average product cost']].copy()
Input4.to_excel('Input4.xlsx', index=False)
print(Input4.head())
```

```
##      Product ID  Average product cost
## 0             1             6.315177
## 1             2            35.384452
## 2             3             7.708251
## 3             4            20.678609
## 4             5            10.616473
```

- Output1: Average order count

```
Output1 = pd.read_excel('Filled Output1.xlsx')
Output1['Average Order Count'] = round(Output1.iloc[:, 1:].mean(axis=1), 0)
```

```
Output1 = Output1[['Product ID', 'Average Order Count']].copy()
Output1.to_excel('Output1.xlsx', index=False)
```

```
print(Output1.head())
```

```
##      Product ID  Average Order Count
## 0             1            1054.0
## 1             2            1389.0
## 2             3            1031.0
## 3             4             765.0
## 4             5             437.0
```

- Output2: Average gross profit

```
Output2 = pd.read_excel('Filled Output2.xlsx')
Output2['Average gross profit'] = round(Output2.iloc[:, 1:].mean(axis=1), 0)
Output2 = Output2[['Product ID', 'Average gross profit']].copy()
Output2.to_excel('Output2.xlsx', index=False)
print(Output2.head())
```

```
##      Product ID  Average gross profit
## 0             1             2.0
```



```
## 1      2      26.0
## 2      3      14.0
## 3      4      15.0
## 4      5       7.0
```

- Output3: Net promoter score

```
Output3 = pd.read_excel('Filled Output3.xlsx')
Output3['Monthly NPS'] = round(Output3.iloc[:, 1])

Output3 = Output3[['Product ID', 'Monthly NPS']].copy()
Output3.to_excel('Output3.xlsx', index=False)
```

```
print(Output3.head())
```

```
##      Product ID  Monthly NPS
## 0             1          68.0
## 1             2          72.0
## 2             3          66.0
## 3             4          76.0
## 4             5          85.0
```

3- Data normalization

- Merging the variables

```
Input1 = pd.read_excel('Input1.xlsx')
Input2 = pd.read_excel('Input2.xlsx')
Input3 = pd.read_excel('Input3.xlsx')
Input4 = pd.read_excel('Input4.xlsx')
Output1 = pd.read_excel('Output1.xlsx')
Output2 = pd.read_excel('Output2.xlsx')
Output3 = pd.read_excel('Output3.xlsx')
```

```
Dataset = pd.merge(Input1, Input2, on='Product ID')
Dataset = pd.merge(Dataset, Input3, on='Product ID')
Dataset = pd.merge(Dataset, Input4, on='Product ID')
Dataset = pd.merge(Dataset, Output1, on='Product ID')
Dataset = pd.merge(Dataset, Output2, on='Product ID')
Dataset = pd.merge(Dataset, Output3, on='Product ID')
Dataset.to_excel('Dataset.xlsx', index=False)
print(Dataset.head())
```

```
##      Product ID  Counted Rejections  ...  Average gross profit  Monthly NPS
## 0             1          1870  ...             2             68
## 1             4           360  ...            15             76
## 2             2           318  ...            26             72
## 3             5           158  ...             7             85
## 4             3           155  ...            14             66
##
## [5 rows x 8 columns]
```

- Normalization

```
from sklearn.preprocessing import MinMaxScaler

data = pd.read_excel('Dataset.xlsx')
scaler = MinMaxScaler()
Variables = ['Counted Rejections', 'Average Shipping Cost', 'Items with Complaint', 'Average product cost', 'Average Order Count', 'Average gross profit', 'Monthly NPS']
normalized_data = scaler.fit_transform(data[Variables])

data[Variables] = normalized_data

data = data.sort_values(by=data.columns[0])
data.to_excel('Normalized Dataset.xlsx', index=False)
```

II- PCA Application

1- Optimal Principal Components

- Input Matrix creation

```
data <- read_excel("Normalized Dataset.xlsx")

input.matrix <- matrix(nrow = nrow(data), ncol = 4)
colnames(input.matrix) <- c("Counted Rejections", "Average Shipping Cost", "Items with Complaint", "Average product cost")
rownames(input.matrix) <- paste(1:nrow(data), sep = "")

input.matrix[, "Counted Rejections"] <- data$`Counted Rejections`
input.matrix[, "Average Shipping Cost"] <- data$`Average Shipping Cost`
input.matrix[, "Items with Complaint"] <- data$`Items with Complaint`
input.matrix[, "Average product cost"] <- data$`Average product cost`

input.matrix <- apply(input.matrix, 2, as.numeric)
head(input.matrix)

##      Counted Rejections Average Shipping Cost Items with Complaint
## [1,]      1.00000000      0.6181396      1.0000000
## [2,]      0.16960942      0.4752585      0.4968553
## [3,]      0.08239700      0.2312064      0.3081761
## [4,]      0.19208133      0.2752352      0.3018868
## [5,]      0.08400214      0.2772332      0.2578616
## [6,]      0.02140182      0.4222182      0.2075472
##      Average product cost
## [1,]      0.02478593
## [2,]      0.16322217
## [3,]      0.03142015
## [4,]      0.09318872
## [5,]      0.04526994
## [6,]      0.02268905
```

- Output Matrix creation

```
data <- read_excel("Normalized Dataset.xlsx")

output.matrix <- matrix(nrow = nrow(data), ncol = 3)
colnames(output.matrix) <- c("Average Order Count", "Average gross profit",
"Monthly NPS")
rownames(output.matrix) <- paste(1:nrow(data), sep = "")

output.matrix[, "Average Order Count"] <- data$`Average Order Count`
output.matrix[, "Average gross profit"] <- data$`Average gross profit`
output.matrix[, "Monthly NPS"] <- data$`Average gross profit`

output.matrix <- apply(output.matrix, 2, as.numeric)
head(output.matrix)

##      Average Order Count Average gross profit Monthly NPS
## [1,]          0.7588193          0.03040541  0.03040541
## [2,]          1.0000000          0.11148649  0.11148649
## [3,]          0.7422606          0.07094595  0.07094595
## [4,]          0.5507559          0.07432432  0.07432432
## [5,]          0.3146148          0.04729730  0.04729730
## [6,]          0.1929446          0.05405405  0.05405405
```

- Summary of PCA application

```
pca.output <- prcomp(output.matrix, scale = TRUE)
summary(pca.output)

## Importance of components:
##              PC1      PC2      PC3
## Standard deviation    1.4193 0.9928 1.301e-16
## Proportion of Variance 0.6714 0.3286 0.000e+00
## Cumulative Proportion 0.6714 1.0000 1.000e+00

pca.input <- prcomp(input.matrix, scale = TRUE)
summary(pca.input)

## Importance of components:
##              PC1      PC2      PC3      PC4
## Standard deviation    1.4122 1.0096 0.9489 0.29299
## Proportion of Variance 0.4986 0.2548 0.2251 0.02146
## Cumulative Proportion 0.4986 0.7534 0.9785 1.00000
```

- Scree Plot

```
variance <- pca.input$sdev^2 / sum(pca.input$sdev^2)
eigenvalues <- pca.input$sdev^2

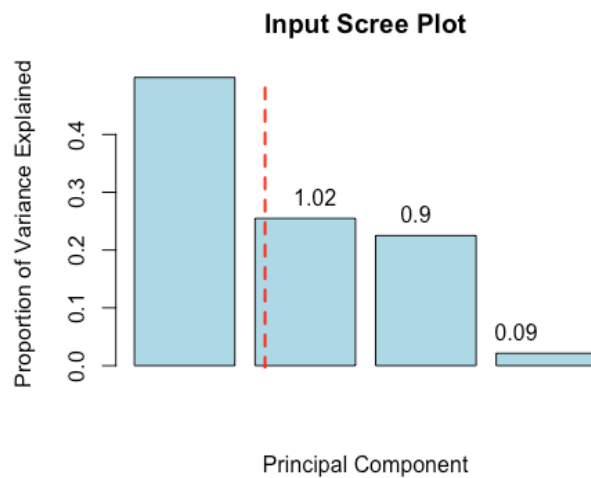
barplot(variance, xlab = "Principal Component", ylab = "Proportion of Variance
Explained",
        main = "Input Scree Plot", col = "lightblue")
text(x = 1:length(eigenvalues), y = variance, labels = round(eigenvalues, 2), pos
= 3)

elbow_point <- 0
for (i in 2:length(eigenvalues)) {
```

```

if (variance[i] - variance[i-1] < 0.05) {
  elbow_point <- i - 1
  break}}
abline(v = elbow_point + 0.5, col = "red", lwd = 2, lty = 2)

```



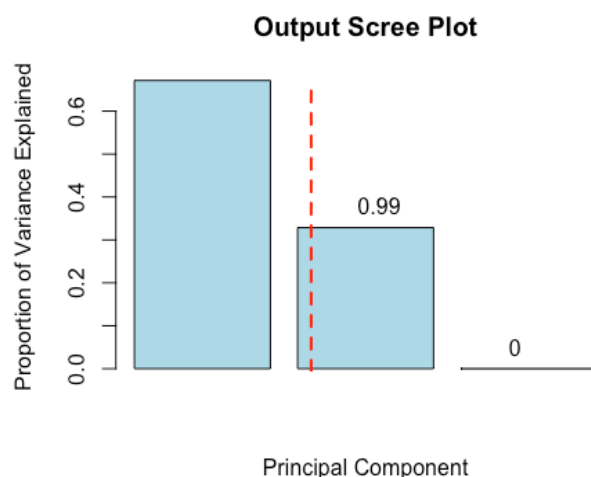
```

variance <- pca.output$sdev^2 / sum(pca.output$sdev^2)
eigenvalues <- pca.output$sdev^2

barplot(variance, xlab = "Principal Component", ylab = "Proportion of Variance Explained",
        main = "Output Scree Plot", col = "lightblue")
text(x = 1:length(eigenvalues), y = variance, labels = round(eigenvalues, 2), pos = 3)

elbow_point <- 0
for (i in 2:length(eigenvalues)) {
  if (variance[i] - variance[i-1] < 0.05) {
    elbow_point <- i - 1
    break
  }
}
abline(v = elbow_point + 0.5, col = "red", lwd = 2, lty = 2)

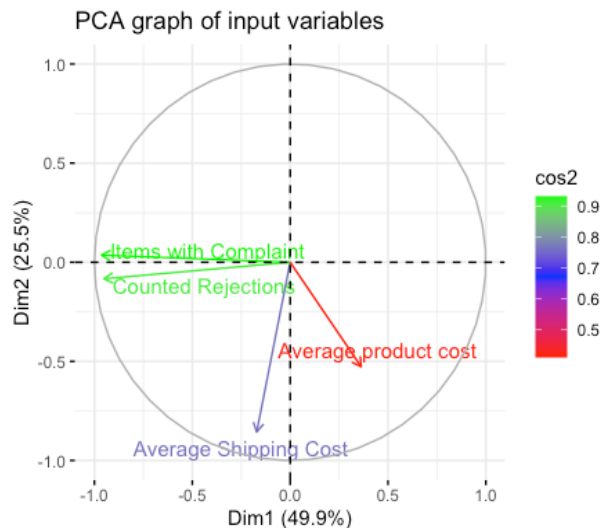
```



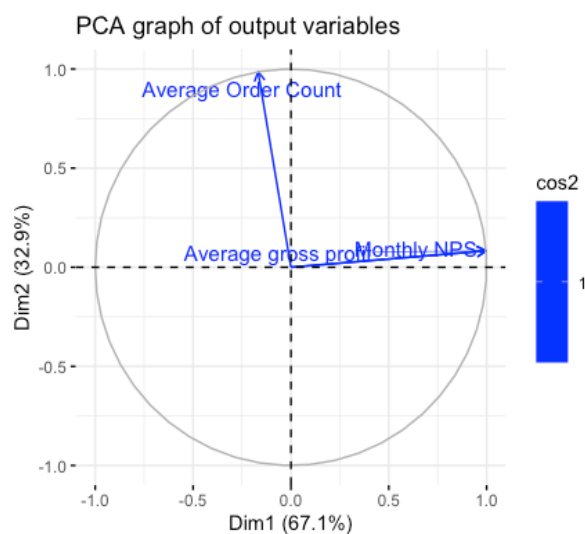
2- PCA Visualization

- The PCA graph of variables

```
input.variables <- fviz_pca_var(pca.input,  
  col.var = "cos2",  
  gradient.cols = c("red", "blue", "green"),  
  repel = TRUE)  
input.variables <- input.variables + ggtitle("PCA graph of input variables")  
input.variables
```

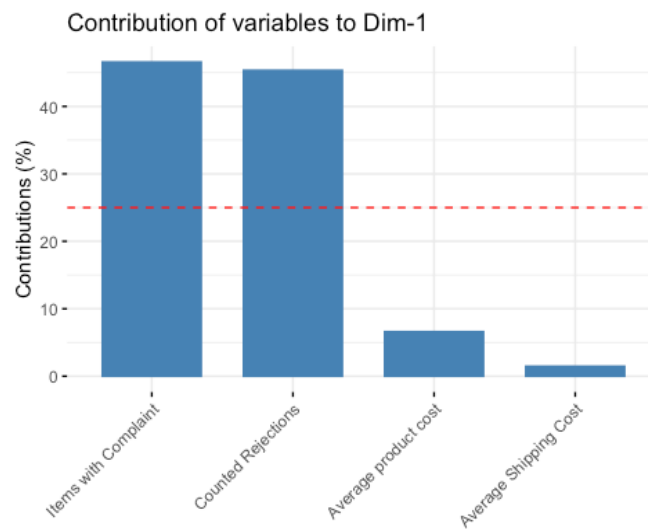


```
output.variables <- fviz_pca_var(pca.output,  
  col.var = "cos2",  
  gradient.cols = c("red", "blue", "green"),  
  repel = TRUE)  
output.variables <- output.variables + ggtitle("PCA graph of output variables")  
output.variables
```

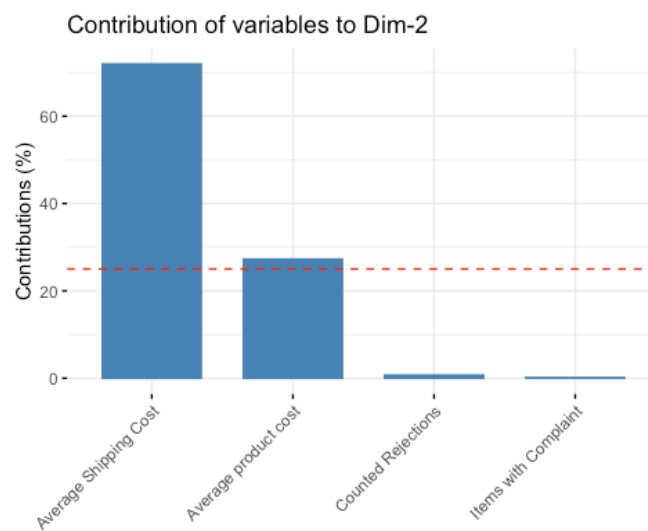


- The Contribution Plot

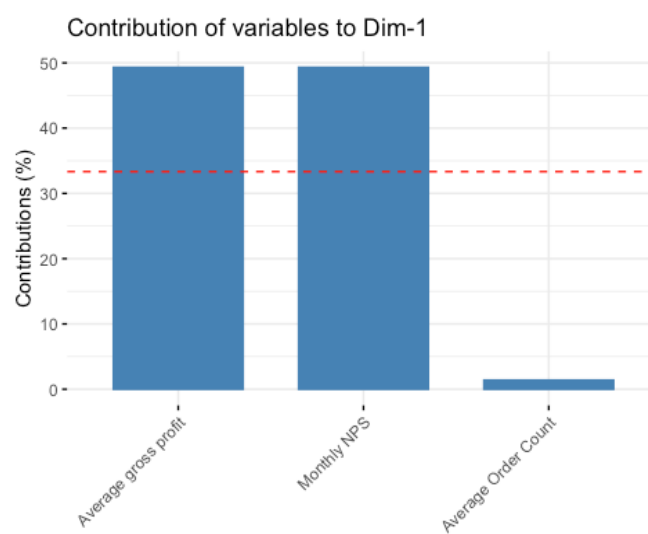
```
fviz_contrib(pca.input, choice = 'var')
```



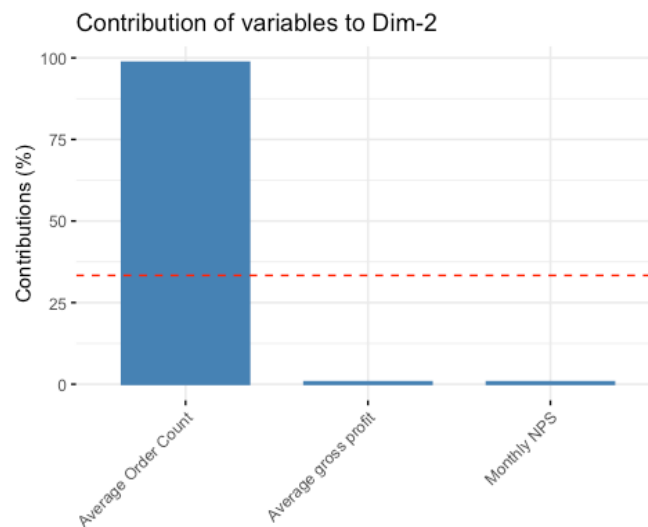
```
fviz_contrib(pca.input,choice = 'var',axes = 2)
```



```
fviz_contrib(pca.output,choice = 'var')
```



```
fviz_contrib(pca.output,choice = 'var',axes = 2)
```



- The PC scores

```
input.scores <- as.data.frame(predict(pca.input, newdata = input.matrix))
colnames(input.scores) <- paste("Input", colnames(input.scores), sep = " ")
```

```
write_xlsx(input.scores, "Input scores.xlsx")
head(input.scores)
```

```
##      Input PC1  Input PC2  Input PC3  Input PC4
## 1 -7.0533346 -0.7502417  1.0689154 -0.59431286
## 2 -1.8165282 -0.2303109  0.3421541  0.94428083
## 3 -0.8322619  0.9833476  0.1433049  0.51023487
## 4 -1.1921195  0.6254730  0.4017478  0.07412179
## 5 -0.6707294  0.7709152  0.0466701  0.33482523
## 6 -0.3520460  0.3048418 -0.4856597  0.42842211
```

```
output.scores <- as.data.frame(predict(pca.output, newdata = output.matrix))
colnames(output.scores) <- paste("Output", colnames(output.scores), sep = " ")
```

```
write_xlsx(output.scores, "Output scores.xlsx")
head(output.scores)
```

```
##      Output PC1 Output PC2      Output PC3
## 1 -0.8847830  2.3909767  1.665335e-16
## 2 -0.3370888  3.4517808 -2.081668e-17
## 3 -0.5445279  2.3630913  8.326673e-17
## 4 -0.4241143  1.5868337  5.551115e-17
## 5 -0.5312655  0.5992367  1.110223e-16
## 6 -0.4169795  0.1105447  8.326673e-17
```

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
```

```
PC.input = pd.read_excel('Input scores.xlsx')
PC.output = pd.read_excel('Output scores.xlsx')
PCA.scores = pd.concat([PC.input, PC.output], axis=1)
scaler = MinMaxScaler()
normalized.scores = scaler.fit_transform(PCA.scores)
normalized.dataset = pd.DataFrame(normalized.scores, columns=PCA.scores.columns)

dataset = pd.read_excel('Normalized Dataset.xlsx')
```

```
merged = pd.concat([dataset, normalized.dataset], axis=1)

merged_data.to_excel('PCA-DEA Dataset.xlsx', index=False)
print(merged.head())

##      Product ID  Counted Rejections  ...  Output PC2  Output PC3
## 0           1           1.000000  ...      0.743642  1.498801e-15
## 1           2           0.169609  ...      1.000000  1.311451e-15
## 2           3           0.082397  ...      0.736903  1.415534e-15
## 3           4           0.192081  ...      0.549310  1.387779e-15
## 4           5           0.084002  ...      0.310643  1.443290e-15
##
## [5 rows x 15 columns]
```

III- DEA- PCA Application

1 - Model 1: Basic DEA Model

```
data <- read_excel("Normalized Dataset.xlsx")
x <- data.matrix(data[, colnames(data) %in% c("Counted Rejections", "Average
Shipping Cost", "Items with Complaint", "Average product cost")])
y <- data.matrix(data[, colnames(data) %in% c("Average Order Count", "Average
gross profit", "Monthly NPS")])

deam1results <- dea(x, y, RTS = "vrs")
efficiencies <- efficiencies(deam1results)
status <- ifelse(efficiencies == 1, "Efficient", "Inefficient")
m1results <- data.frame(Efficiency = efficiencies, Status = status, row.names =
row.names(x))

subset <- subset(data, select = c("Product ID", colnames(x), colnames(y)))
m1data <- merge(subset, m1results, all = TRUE, sort = FALSE)
head(m1data)

##      Product ID  Counted Rejections  Average Shipping Cost  Items with Complaint
## 1           1           1.000000000           0.6181396           1.0000000
## 2           2           0.16960942           0.4752585           0.4968553
## 3           3           0.08239700           0.2312064           0.3081761
## 4           4           0.19208133           0.2752352           0.3018868
## 5           5           0.08400214           0.2772332           0.2578616
## 6           6           0.02140182           0.4222182           0.2075472
##      Average product cost  Average Order Count  Average gross profit  Monthly NPS
## 1           0.02478593           0.7588193           0.03040541           0.5223881
## 2           0.16322217           1.0000000           0.11148649           0.5820896
## 3           0.03142015           0.7422606           0.07094595           0.4925373
## 4           0.09318872           0.5507559           0.07432432           0.6417910
## 5           0.04526994           0.3146148           0.04729730           0.7761194
## 6           0.02268905           0.1929446           0.05405405           0.4328358
##      Efficiency  Status
## 1           1  Efficient
## 2           1  Efficient
## 3           1  Efficient
## 4           1  Efficient
```



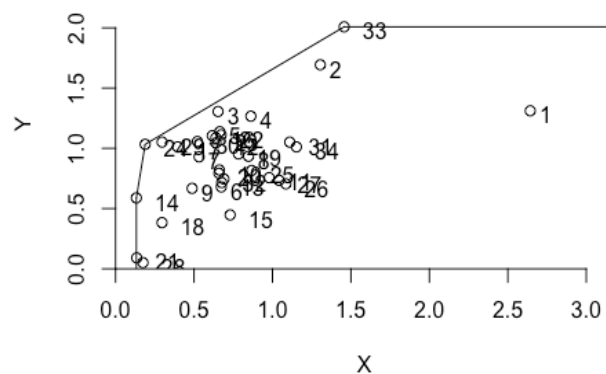
```
## 5      1 Efficient
## 6      1 Efficient
```

```
m1results
```

##	Efficiency	Status
## 1	1.000000	Efficient
## 2	1.000000	Efficient
## 3	1.000000	Efficient
## 4	0.8005266	Inefficient
## 5	0.6472558	Inefficient
## 6	0.9521556	Inefficient
## 7	0.8490536	Inefficient
## 8	1.000000	Efficient
## 9	0.9983325	Inefficient
## 10	1.000000	Efficient
## 11	0.6952274	Inefficient
## 12	1.000000	Efficient
## 13	0.6409660	Inefficient
## 14	1.000000	Efficient
## 15	1.000000	Efficient
## 16	1.000000	Efficient
## 17	1.000000	Efficient
## 18	1.000000	Efficient
## 19	0.7604851	Inefficient
## 20	1.000000	Efficient
## 21	1.000000	Efficient
## 22	1.000000	Efficient
## 23	0.9304941	Inefficient
## 24	1.000000	Efficient
## 25	0.4901398	Inefficient
## 26	1.000000	Efficient
## 27	1.000000	Efficient
## 28	1.000000	Efficient
## 29	1.000000	Efficient
## 30	1.000000	Efficient
## 31	1.000000	Efficient
## 32	0.6481731	Inefficient
## 33	1.000000	Efficient
## 34	1.000000	Efficient

- DEA frontier of Model1

```
dea.plot.frontier(x,y,RTS = "vrs" , txt=TRUE)
```



- Efficient DMUs of Model1

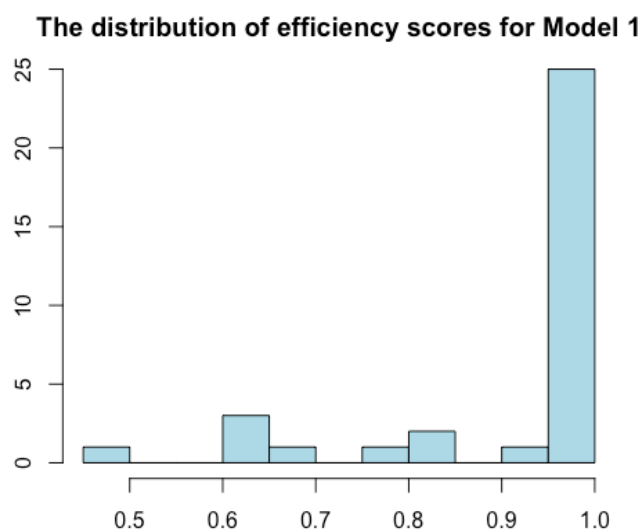
```
m1efficientDMUs <- subset(m1results, Status == "Efficient")
m1efficientDMUs
```

```
##      Efficiency    Status
## 1           1 Efficient
## 2           1 Efficient
## 3           1 Efficient
## 8           1 Efficient
## 10          1 Efficient
## 12          1 Efficient
## 14          1 Efficient
## 15          1 Efficient
## 16          1 Efficient
## 17          1 Efficient
## 18          1 Efficient
## 20          1 Efficient
## 21          1 Efficient
## 22          1 Efficient
## 24          1 Efficient
## 26          1 Efficient
## 27          1 Efficient
## 28          1 Efficient
## 29          1 Efficient
## 30          1 Efficient
## 31          1 Efficient
## 33          1 Efficient
## 34          1 Efficient
```

- Distribution of efficiency scores of Model1

To visualize the distribution of efficiency scores, a histogram was created. It will provide an overview of the efficiency scores, allowing the identification of the distribution and range of efficiencies among the DMUs.

```
par(mar = c(2, 2, 2, 2) + 0.1)
hist(efficiencies, breaks = 10, col = "lightblue", main = "The distribution of
efficiency scores for Model 1", xlab = "Efficiency")
```



2 - Model 2: Input dimension reduction

```
data <- read_excel("PCA-DEA Normalized Dataset.xlsx")
x <- data.matrix(data[, colnames(data) %in% c("Input PC1", "Input PC2")])
y <- data.matrix(data[, colnames(data) %in% c("Average Order Count", "Average
gross profit", "Monthly NPS")])

deam2results <- dea(x, y, RTS = "vrs")

m2efficiencies <- efficiencies(deam2results)
status <- ifelse(efficiencies == 1, "Efficient", "Inefficient")
m2results <- data.frame(Efficiency = efficiencies, Status = status, row.names =
row.names(x))
subset <- subset(data, select = c("Product ID", colnames(x), colnames(y)))
m2data <- merge(subset, m2results, all = TRUE, sort = FALSE)

head(m2data)
```

##	Product ID	Input PC1	Input PC2	Average Order Count	Average gross profit
## 1	1	0.0000000	0.3439481	0.7588193	0.03040541
## 2	2	0.7268872	0.8033690	1.0000000	0.11148649
## 3	3	0.7457611	0.7470720	0.7422606	0.07094595
## 4	4	0.6118828	0.4817357	0.5507559	0.07432432
## 5	5	0.6848404	0.7085281	0.3146148	0.04729730
## 6	6	0.8152041	0.7871323	0.1929446	0.05405405

```
## Monthly NPS Efficiency Status
## 1 0.5223881 1 Efficient
## 2 0.5820896 1 Efficient
## 3 0.4925373 1 Efficient
## 4 0.6417910 1 Efficient
## 5 0.7761194 1 Efficient
## 6 0.4328358 1 Efficient

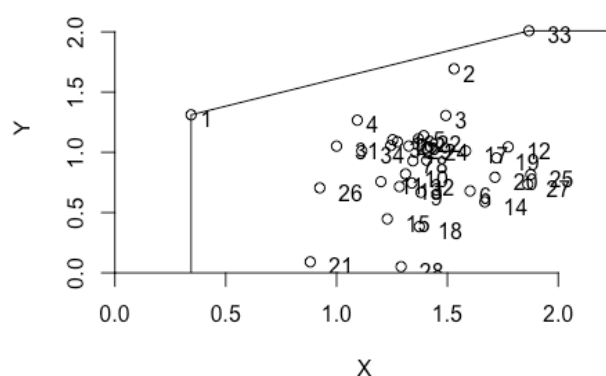
m2results
```

##	Efficiency	Status
## 1	1.0000000	Efficient
## 2	1.0000000	Efficient
## 3	1.0000000	Efficient
## 4	0.8005266	Inefficient
## 5	0.6472558	Inefficient
## 6	0.9521556	Inefficient
## 7	0.8490536	Inefficient
## 8	1.0000000	Efficient
## 9	0.9983325	Inefficient
## 10	1.0000000	Efficient
## 11	0.6952274	Inefficient
## 12	1.0000000	Efficient
## 13	0.6409660	Inefficient
## 14	1.0000000	Efficient
## 15	1.0000000	Efficient
## 16	1.0000000	Efficient
## 17	1.0000000	Efficient
## 18	1.0000000	Efficient
## 19	0.7604851	Inefficient
## 20	1.0000000	Efficient

```
## 21  1.0000000  Efficient
## 22  1.0000000  Efficient
## 23  0.9304941  Inefficient
## 24  1.0000000  Efficient
## 25  0.4901398  Inefficient
## 26  1.0000000  Efficient
## 27  1.0000000  Efficient
## 28  1.0000000  Efficient
## 29  1.0000000  Efficient
## 30  1.0000000  Efficient
## 31  1.0000000  Efficient
## 32  0.6481731  Inefficient
## 33  1.0000000  Efficient
## 34  1.0000000  Efficient
```

- DEA frontier of Model2

```
dea.plot.frontier(x,y,RTS = "vrs" , txt=TRUE)
```



- Efficient DMUs of Model2

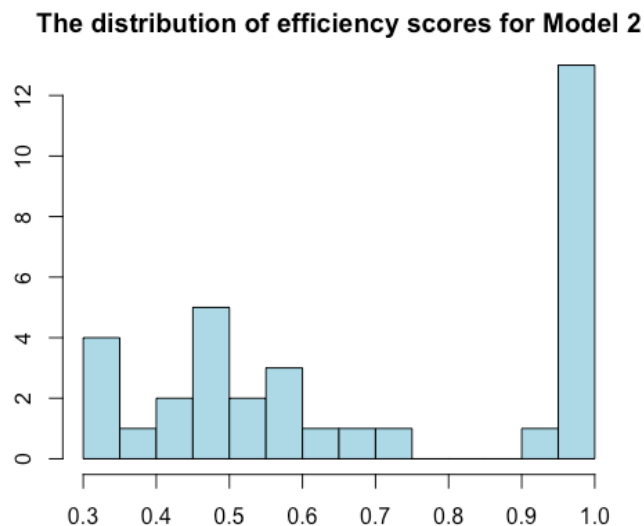
```
m2efficientDMUs <- subset(m2results, Status == "Efficient")
m2efficientDMUs
```

```
##      Efficiency    Status
## 1             1 Efficient
## 2             1 Efficient
## 3             1 Efficient
## 8             1 Efficient
## 10            1 Efficient
## 12            1 Efficient
## 14            1 Efficient
## 15            1 Efficient
## 16            1 Efficient
## 17            1 Efficient
## 18            1 Efficient
## 20            1 Efficient
## 21            1 Efficient
## 22            1 Efficient
## 24            1 Efficient
## 26            1 Efficient
## 27            1 Efficient
## 28            1 Efficient
## 29            1 Efficient
```

```
## 30      1 Efficient
## 31      1 Efficient
## 33      1 Efficient
## 34      1 Efficient
```

- Distribution of efficiency scores of Model2

```
par(mar = c(2, 2, 2, 2) + 0.1)
hist(m2efficiencies, breaks = 10, col = "lightblue", main = "The distribution of
efficiency scores for Model 2", xlab = "Efficiency")
```



3 - Model 3: Output dimension reduction

```
data <- read_excel("PCA-DEA Normalized Dataset.xlsx")
x <- data.matrix(data[, colnames(data) %in% c("Counted Rejections", "Average
Shipping Cost", "Items with Complaint", "Average product cost")])
y <- data.matrix(data[, colnames(data) %in% c("Output PC1" , "Output PC2" )])

deam3results <- dea(x, y, RTS = "vrs")
m3efficiencies <- efficiencies(deam3results)
```

```
status <- ifelse(efficiencies == 1, "Efficient", "Inefficient")
m3results <- data.frame(Efficiency = efficiencies, Status = status, row.names =
row.names(x))
subset <- subset(data, select = c("Product ID", colnames(x), colnames(y)))
m3data <- merge(subset, m3results, all = TRUE, sort = FALSE)
```

```
m3results
```

```
##      Efficiency      Status
## 1      1.0000000 Efficient
## 2      1.0000000 Efficient
## 3      1.0000000 Efficient
## 4      0.8005266 Inefficient
## 5      0.6472558 Inefficient
## 6      0.9521556 Inefficient
## 7      0.8490536 Inefficient
```

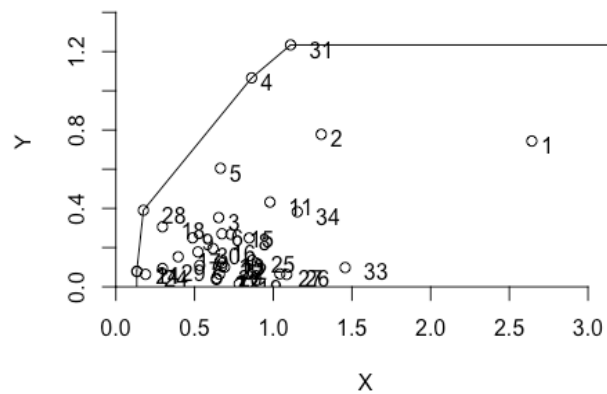
```
## 8 1.0000000 Efficient
## 9 0.9983325 Inefficient
## 10 1.0000000 Efficient
## 11 0.6952274 Inefficient
## 12 1.0000000 Efficient
## 13 0.6409660 Inefficient
## 14 1.0000000 Efficient
## 15 1.0000000 Efficient
## 16 1.0000000 Efficient
## 17 1.0000000 Efficient
## 18 1.0000000 Efficient
## 19 0.7604851 Inefficient
## 20 1.0000000 Efficient
## 21 1.0000000 Efficient
## 22 1.0000000 Efficient
## 23 0.9304941 Inefficient
## 24 1.0000000 Efficient
## 25 0.4901398 Inefficient
## 26 1.0000000 Efficient
## 27 1.0000000 Efficient
## 28 1.0000000 Efficient
## 29 1.0000000 Efficient
## 30 1.0000000 Efficient
## 31 1.0000000 Efficient
## 32 0.6481731 Inefficient
## 33 1.0000000 Efficient
## 34 1.0000000 Efficient
```

```
head(m3data)
```

```
## Product ID Counted Rejections Average Shipping Cost Items with Complaint
## 1 1 1.00000000 0.6181396 1.0000000
## 2 2 0.16960942 0.4752585 0.4968553
## 3 3 0.08239700 0.2312064 0.3081761
## 4 4 0.19208133 0.2752352 0.3018868
## 5 5 0.08400214 0.2772332 0.2578616
## 6 6 0.02140182 0.4222182 0.2075472
## Average product cost Output PC1 Output PC2 Efficiency Status
## 1 0.02478593 0.00000000 0.7436420 1 Efficient
## 2 0.16322217 0.04094896 0.7369031 1 Efficient
## 3 0.03142015 0.04254505 0.3106431 1 Efficient
## 4 0.09318872 0.06591380 1.0000000 1 Efficient
## 5 0.04526994 0.05544048 0.5493097 1 Efficient
## 6 0.02268905 0.12317320 0.1474658 1 Efficient
```

- DEA frontier of Model3

```
dea.plot.frontier(x,y,RTS = "vrs" , txt=TRUE)
```



- Efficient DMUs of Model3

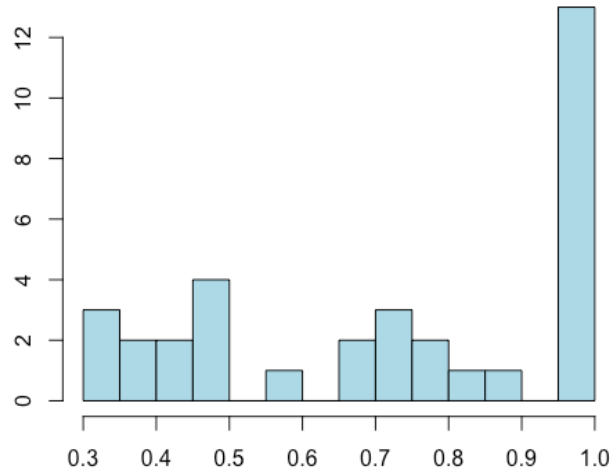
```
m3efficientDMUs <- subset(m3results, Status == "Efficient")
m3efficientDMUs
```

```
##      Efficiency      Status
## 1             1 Efficient
## 2             1 Efficient
## 3             1 Efficient
## 8             1 Efficient
## 10            1 Efficient
## 12            1 Efficient
## 14            1 Efficient
## 15            1 Efficient
## 16            1 Efficient
## 17            1 Efficient
## 18            1 Efficient
## 20            1 Efficient
## 21            1 Efficient
## 22            1 Efficient
## 24            1 Efficient
## 26            1 Efficient
## 27            1 Efficient
## 28            1 Efficient
## 29            1 Efficient
## 30            1 Efficient
## 31            1 Efficient
## 33            1 Efficient
## 34            1 Efficient
```

- Distribution of efficiency scores of Model3

```
par(mar = c(2, 2, 2, 2) + 0.1)
hist(m3efficiencies, breaks = 10, col = "lightblue", main = "The distribution of
efficiency scores for Model 3", xlab = "Efficiency")
```

The distribution of efficiency scores for Model 3



4 - Model 4: Joint Input-Output dimension reduction

```
data <- read_excel("PCA-DEA Normalized Dataset.xlsx")
x <- data.matrix(data[, colnames(data) %in% c("Input PC1", "Input PC2" )])
y <- data.matrix(data[, colnames(data) %in% c("Output PC1" , "Output PC2" )])

deam4results <- dea(x, y, RTS = "vrs")
```

```
efficiencies <- efficiencies(deam4results)
status <- ifelse(efficiencies == 1, "Efficient", "Inefficient")
m4results <- data.frame(Efficiency = efficiencies, Status = status, row.names =
row.names(x))
subset <- subset(data, select = c("Product ID", colnames(x), colnames(y)))
```

```
m4data <- merge(subset, m4results, all = TRUE, sort = FALSE)
head(m4data)
```

##	Product ID	Input PC1	Input PC2	Output PC1	Output PC2	Efficiency	Status
## 1	1	0.0000000	0.3439481	0.0000000	0.7436420	1	Efficient
## 2	2	0.7268872	0.8033690	0.04094896	0.7369031	1	Efficient
## 3	3	0.7457611	0.7470720	0.04254505	0.3106431	1	Efficient
## 4	4	0.6118828	0.4817357	0.06591380	1.0000000	1	Efficient
## 5	5	0.6848404	0.7085281	0.05544048	0.5493097	1	Efficient
## 6	6	0.8152041	0.7871323	0.12317320	0.1474658	1	Efficient

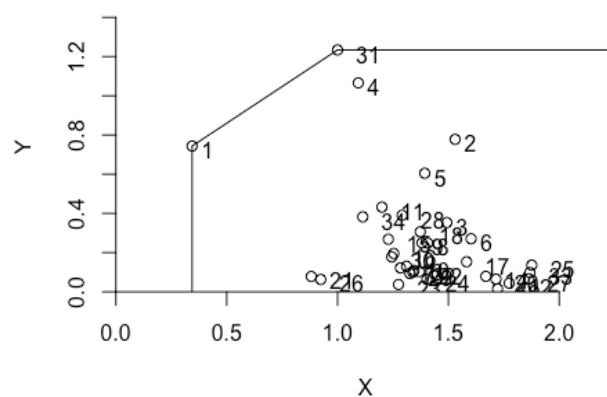
```
m4results
```

##	Efficiency	Status
## 1	1.0000000	Efficient
## 2	0.4112271	Inefficient
## 3	0.3389601	Inefficient
## 4	1.0000000	Efficient
## 5	0.3610169	Inefficient
## 6	0.3198785	Inefficient
## 7	0.4267272	Inefficient
## 8	0.3803588	Inefficient
## 9	0.4096064	Inefficient
## 10	0.4458532	Inefficient
## 11	0.5359343	Inefficient


```
## 12 0.2852426 Inefficient
## 13 0.4669387 Inefficient
## 14 0.3085913 Inefficient
## 15 0.4663935 Inefficient
## 16 0.4945957 Inefficient
## 17 0.3229626 Inefficient
## 18 0.4058591 Inefficient
## 19 0.2996400 Inefficient
## 20 0.3039690 Inefficient
## 21 1.0000000 Efficient
## 22 0.4272199 Inefficient
## 23 0.5338813 Inefficient
## 24 0.4054987 Inefficient
## 25 0.2626518 Inefficient
## 26 0.9497194 Inefficient
## 27 0.2693887 Inefficient
## 28 0.4636416 Inefficient
## 29 0.4542843 Inefficient
## 30 0.5173772 Inefficient
## 31 1.0000000 Efficient
## 32 0.4391914 Inefficient
## 33 0.2661858 Inefficient
## 34 0.6935117 Inefficient
```

- DEA frontier of Model4

```
dea.plot.frontier(x,y,RTS = "vrs" , txt=TRUE)
```



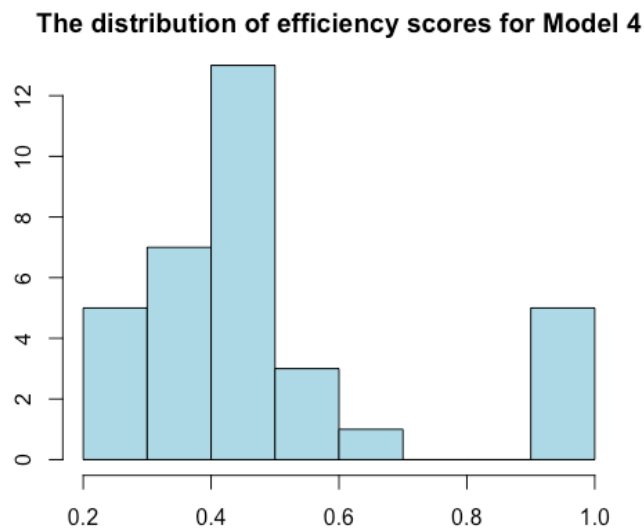
- Efficient DMUs of Model4

```
m4efficientDMUs <- subset(m4results, Status == "Efficient")
m4efficientDMUs
```

```
##      Efficiency      Status
## 1             1 Efficient
## 4             1 Efficient
## 21            1 Efficient
## 31            1 Efficient
```

- Distribution of efficiency scores of Model4

```
par(mar = c(2, 2, 2, 2) + 0.1)
hist(efficiencies, breaks = 10, col = "lightblue", main = "The distribution of
efficiency scores for Model 4", xlab = "Efficiency")
```



VI- Interpretation

- Peers Analysis

```
peers <- data.frame(peers(deam4results))
lambda <- data.frame(lambda(deam4results))
lambda.matrix <- as.matrix(lambda(deam4results))
```

peers

##	peer1	peer2	peer3
## 1	1	NA	NA
## 2	1	4	31
## 3	1	21	31
## 4	4	NA	NA
## 5	1	21	31
## 6	1	21	31
## 7	1	21	31
## 8	1	21	31
## 9	1	21	31
## 10	1	21	31
## 11	1	21	31
## 12	1	21	31
## 13	1	21	31
## 14	1	21	31
## 15	1	21	31
## 16	1	21	31
## 17	1	21	31
## 18	1	21	31
## 19	1	21	NA
## 20	1	21	31
## 21	21	NA	NA
## 22	1	21	31
## 23	1	21	NA

```
## 24      1      21      31
## 25      1      21      31
## 26      1      21      NA
## 27      1      21      31
## 28      1      21      31
## 29      1      21      31
## 30      1      21      31
## 31     31      NA      NA
## 32      1      21      31
## 33      1      21      31
## 34      1      21      31
```

lambda

```
##          L1          L4          L21          L31
## 1  1.000000000 0.0000000 0.0000000 0.000000000
## 2  0.601637009 0.2562301 0.0000000 0.142132937
## 3  0.703787374 0.0000000 0.2718952 0.024317443
## 4  0.000000000 1.0000000 0.0000000 0.000000000
## 5  0.716876791 0.0000000 0.2246581 0.058465134
## 6  0.710810859 0.0000000 0.1779452 0.111243908
## 7  0.574267011 0.0000000 0.4173566 0.008376376
## 8  0.652288771 0.0000000 0.3123519 0.035359334
## 9  0.613442379 0.0000000 0.2693476 0.117210038
## 10 0.560301987 0.0000000 0.4029443 0.036753747
## 11 0.494293162 0.0000000 0.2746216 0.231085261
## 12 0.704052971 0.0000000 0.2699967 0.025950378
## 13 0.530693728 0.0000000 0.4418229 0.027483350
## 14 0.687519654 0.0000000 0.2862151 0.026265272
## 15 0.584191351 0.0000000 0.3673955 0.048413184
## 16 0.501932174 0.0000000 0.4275716 0.070496193
## 17 0.691363153 0.0000000 0.2986555 0.009981371
## 18 0.617786463 0.0000000 0.3146983 0.067515188
## 19 0.680064076 0.0000000 0.3199359 0.000000000
## 20 0.677747496 0.0000000 0.2880177 0.034234838
## 21 0.000000000 0.0000000 1.0000000 0.000000000
## 22 0.561628490 0.0000000 0.3919963 0.046375207
## 23 0.373412001 0.0000000 0.6265880 0.000000000
## 24 0.583120268 0.0000000 0.4086506 0.008229104
## 25 0.745116797 0.0000000 0.1551831 0.099700097
## 26 0.006375141 0.0000000 0.9936249 0.000000000
## 27 0.716999595 0.0000000 0.2345113 0.048489116
## 28 0.563525913 0.0000000 0.2669031 0.169570941
## 29 0.531451228 0.0000000 0.4121438 0.056404999
## 30 0.468301195 0.0000000 0.4123791 0.119319706
## 31 0.000000000 0.0000000 0.0000000 1.000000000
## 32 0.549238042 0.0000000 0.4320308 0.018731137
## 33 0.720930104 0.0000000 0.2517079 0.027361961
## 34 0.266157353 0.0000000 0.4465007 0.287341965
```

- Ranking of DMUs Based on peers weight

```
rankedDMUs <- rank(-lambda, ties.method = "min")
rankedDMUs <- data.frame(DMU = rownames(lambda), Rank = rankedDMUs)
rankedDMUs <- rankedDMUs[order(rankedDMUs$Rank), ]
```

```
##      DMU Rank
## 1      1      1
```

```
## 2    4    1
## 3   21    1
## 4   31    1
## 5   26    5
## 6   25    6
## 7   33    7
## 8   27    8
## 9    5    9
## 10   6   10
## 11  12   11
## 12   3   12
## 13  17   13
## 14  14   14
## 15  19   15
## 16  20   16
## 17   8   17
## 18  23   18
## 19  18   19
## 20   9   20
## 21   2   21
## 22  15   22
## 23  24   23
## 24   7   24
## 25  28   25
## 26  22   26
## 27  10   27
## 28  32   28
## 29  29   29
## 30  13   30
## 31  16   31
## 32  11   32
## 33  30   33
## 34  34   34
```

- Ranking of DMUs Based on efficiency scores

```
m4results <- m4results[order(-m4results$Efficiency), ]
m4results$Rank <- seq_len(nrow(m4results))
```

```
subset <- data.frame(Rank = m4results$Rank)
row.names(subset) <- row.names(m4results)
```

```
subset
```

```
##      Rank
## 1      1
## 4      2
## 21     3
## 31     4
## 26     5
## 34     6
## 11     7
## 23     8
## 30     9
## 16    10
## 13    11
## 15    12
```

```
## 28 13
## 29 14
## 10 15
## 32 16
## 22 17
## 7 18
## 2 19
## 9 20
## 18 21
## 24 22
## 8 23
## 5 24
## 3 25
## 17 26
## 6 27
## 14 28
## 20 29
## 19 30
## 12 31
## 27 32
## 33 33
## 25 34
```

- Excess Analysis

```
excess <- data.frame(excess(deam4results,x,y))
mean <- colMeans(excess)
excess <- rbind(excess, mean)
```

```
excess
```

```
##      Input.PC1  Input.PC2
## 1  0.00000000 0.00000000
## 2  0.42797146 0.473001893
## 3  0.49297782 0.493844401
## 4  0.00000000 0.00000000
## 5  0.43760148 0.452737545
## 6  0.55443785 0.535345656
## 7  0.48237998 0.288365201
## 8  0.48517714 0.386381765
## 9  0.49516067 0.320054411
## 10 0.46650205 0.260081812
## 11 0.39990914 0.156974201
## 12 0.63351534 0.634568066
## 13 0.45519894 0.229084183
## 14 0.59769264 0.556145333
## 15 0.40859131 0.247143178
## 16 0.43916449 0.194346661
## 17 0.54700277 0.524194095
## 18 0.48594059 0.329972109
## 19 0.62835372 0.577416123
## 20 0.63255483 0.560849817
## 21 0.00000000 0.00000000
## 22 0.50378514 0.280561277
## 23 0.45967787 0.134589530
## 24 0.51549007 0.318639210
## 25 0.64595383 0.737348232
```

```
## 26 0.04420258 0.002275514
## 27 0.66593721 0.694943274
## 28 0.45561155 0.236897328
## 29 0.48377062 0.239904683
## 30 0.43453829 0.166042385
## 31 0.00000000 0.000000000
## 32 0.48746663 0.263866359
## 33 0.65849703 0.712061542
## 34 0.29279359 0.048556942
## 35 0.43287814 0.325182139
```