Ministry of Education and Training

Ho Chi Minh University of Economics and Finance

SCIENTIFIC RESEARCH

TOPIC

SUPPLY CHAIN MANAGEMENT ANALYSIS

Research area	Business Intelligence	
Instructor	Master Tran Thanh Cong	
Members	Nguyen Long Vu	215122225
	Nguyen Pham Anh Thu	215122400
	Ha Hai An	205121026
	Nguyen Tran Hong An	215121737
	Nguyen Quoc Huy	205121493

Ho Chi Minh city, March 2024

Ministry of Education and Training

Ho Chi Minh University of Economics and Finance

SCIENTIFIC RESEARCH

TOPIC

SUPPLY CHAIN MANAGEMENT ANALYSIS

Research area	Business Intelligence	
Instructor	Master Tran Thanh Cong	
Members	Nguyen Long Vu	215122225
	Nguyen Pham Anh Thu	215122400
	Ha Hai An	205121026
	Nguyen Tran Hong An	215121737
	Nguyen Quoc Huy	205121493

Ho Chi Minh city, March 2024

REVIEW AND EVALUATION

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ho Chi Minh city, March 2024

Instructor

Master Tran Thanh Cong

WORK COMPLETION LEVEL EVALUATION TABLE

FULL NAME	CODE	CONTRIBUTION (%)
Nguyen Long Vu	215122225	100%
Nguyen Pham Anh Thu	215122400	100%
Nguyen Quoc Huy	205121493	100%
Nguyen Tran Hong An	215121737	100%
Ha Hai An	205121026	100%

PREFACE

In today's business context, effective supply chain management is considered the foundation of success. The complex network of processes, resources, and entities involved in getting a product or service from production to consumer requires data analytics. Supply chain management analysis plays an important and indispensable role in a business.

Through supply chain data analysis, we will address the complex network of supply chain components, shedding light on how data is collected, processed, and interpreted. This serves as the foundation for informed decision making. From demand forecasting to inventory optimization, from supplier relationship management to distribution channel analysis, every aspect of supply chain operations is scrutinized through the Business Intelligence (BI) subject.

As we begin this journey, we must recognize the evolving nature of both supply chain management and data analytics. Rapid advances in technology, proliferation of data sources, and ever-changing global market dynamics require a continuous process of adaptation and innovation. Therefore, this analysis is one of the best practices in the field of Business Intelligence. By demystifying supply chain data analysis, we aim to equip readers with the knowledge and tools needed to navigate the complexities of modern supply chains with precision and proficiency.

When we demystify the complexities of data-driven supply chains, it can inspire innovation, drive strategic thinking and success in the ever-evolving landscape of business modernity.

Therefore, within the scope of the Business Intelligence course, under the guidance of master Tran Thanh Cong along with exploration, research and discussion, our team took advantage of data from Kaggle and used the Python language on the platform. Google Collab and Anaconda platforms to conduct data analysis on supply chain management.

ACKNOWLEDGEMENTS

First and foremost, we would like to express our gratitude to master Tran Thanh Cong, our "Business Intelligence" lecturer and instructor, for his unwavering guidance and assistance over the last several months. He gives us a lot of inspiration and drive to work hard and passionately on the analytical report. His thorough understanding of the subject, his thought-provoking analytical activities, and the course materials he gave will surely be invaluable resources in helping us to enhance our talents for the next semesters and our future careers.

We readily acknowledge that there are several gaps in our expertise and data analysis skills. Even if our research is frequently flawed, what matters most is that we constantly work very hard and take great care to produce the finest report we can. We appreciated the constructive feedback and recommendations from educators and other students who were receptive to new ideas and eager to learn. These will be really helpful to us in finishing the thesis and raising the caliber of our work.

Lastly, we would like to express our sincere gratitude to everyone that supported and encouraged our team. This solidarity and assistance have greatly motivated us to overcome obstacles and advance in our academic careers.

We are grateful for that!

Best Regards,

THEORETICAL BASE PREMISES

Supply chain is all about logistics and transportation of goods, within a country or between countries around the world. Rapid changes in economic trends require better quality of goods and faster shipping times.

Therefore, companies need to serve consumers with the products they ordered in the shortest time frame. Because Supply Chain Management is time-bound, data plays a huge role in supply chain success. Finding the right set of tools that can effectively collate and match data from consumer and supplier systems and create a suitable information roadmap is no small challenge.

On that theoretical basis, data is an indispensable part. With the amount of data properly collected, processed and cleaned, overall management efficiency will be enhanced. Not only does it increase the level of similarity of structures in the data series. It also increases the ability to stick to set goals, thereby promoting sustainable and solid strategic activities.

The idea of the methodology is to use the integration of algorithmic structures, code and specific functions to convert and process raw data into effective data, meaning that each data will be considered as a point that needs to be analyzed and used to combine and solve a common problem. In other words, data is the component that makes up the entire data diagram of an entire supply chain management system.

For that reason, we need to closely align these two issues so that they always go hand in hand and comply with the scientific standards of the data analysis industry.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1. Overview

In modern businesses, supply chain management (SCM) plays a key role in organizational success. Supply chain management involves coordinating the various processes, resources, and stakeholders involved in the flow of goods and services from production to consumption. In today's competitive global market, effectively optimizing supply chain operations has become imperative for businesses to maintain a competitive advantage and meet growing customer demands.

In this context, the combination of supply chain management with data analytics has emerged as a transformative force, allowing organizations to harness unprecedented insights, improve operational efficiency Motivate and promote strategic decision making. Business Intelligence (BI) involves the use of data-driven methods, advanced analytics, and data visualization techniques to extract useful information from vast and disparate data sources.

Supply chain management represents a paradigm shift in the way organizations approach optimizing and analyzing supply chain networks. By exploiting the wealth of data generated at many points across the supply chain continuum, businesses can gain a comprehensive understanding of their operations, identify inefficiencies, reduce risks and take advantage of improvement opportunities.

The purpose of the topic is to shed light on how integrating data-driven approaches enhances the visibility, flexibility and resilience of supply chain operations. Through a comprehensive examination of concepts and methods, this topic equips readers with the knowledge and tools necessary to leverage the subject of Business Intelligence for effective supply chain management analysis.

Throughout this topic, we will delve into different aspects of supply chain management analytics in the Business Intelligence subject, including:

- *Data-driven decision making:* how organizations can leverage tools and techniques to transform raw data into actionable insights, enabling informed decision making at all levels of the supply chain.

- *Demand forecasting and predictive analytics:* The role of forecasting algorithms and predictive modeling in predicting customer demand, optimizing inventory levels, and minimizing out-of-stocks or overstocks redundant.

- *Supplier relationship management:* Strategies for leveraging data analytics to evaluate supplier performance, minimize supply chain risk, and optimize sourcing strategies for cost savings and operational efficiency.

- *Inventory Optimization:* Techniques for analyzing inventory data, identifying demand patterns, and implementing inventory optimization strategies to minimize carrying costs while ensuring availability of product.

- *Distribution channel analytics:* The use of analytics tools to evaluate the performance of distribution channels, optimize shipping routes, and improve the efficiency of last-mile delivery operations.

By exploring these topics and more, this report aims to provide a comprehensive framework for leveraging Business Intelligence in analyzing and optimizing supply chain operations. To become supply chain experts, business leaders seeking to gain a deeper understanding of modern supply chain management practices will provide valuable insights into the potential to transform Business Intelligence innovation in driving supply chain excellence.

1.2. Kaggle

According to Kaggle's website, Kaggle is a crowd-sourced platform to attract, nurture, train and challenge data scientists from all around the world to solve data science, machine learning and predictive analytics problems. It has over 536,000 active members from 194 countries and it receives close to 150,000 submissions per month. Started from Melbourne, Australia Kaggle moved to Silicon Valley in 2011, raised some 11 million dollars from the likes of Hal Varian (Chief Economist at Google), Max Levchin (Paypal), Index and Khosla Ventures

and then ultimately was acquired by Google in March of 2017. Kaggle is the number one stop for data science enthusiasts all around the world who compete for prizes and boost their Kaggle rankings. There are only 94 Kaggle Grandmasters in the world to this date.

Most data scientists are only theorists and rarely get a chance to practice before being employed in the real-world. Kaggle solves this problem by giving data science enthusiasts a platform to interact and compete in solving real-life problems. The experience you get on Kaggle is invaluable in preparing you to understand what goes into finding feasible solutions for big data.

Kaggle enables data scientists and other developers to engage in running machine learning contests, write and share code, and to host datasets. The types of data science problems posted on Kaggle can be anything from attempting to predict cancer occurrence by examining patient records to analyzing sentiment to evoke by movie reviews and how this affects audience reaction.

While some are just for educational purposes and fun brain exercises, others are genuine issues that companies are trying to solve. Kaggle makes the environment competitive by awarding prizes and rankings for winners and participants. The prizes are not only monetary but can also include attractive rewards such as jobs or free products from the company hosting the competition.

Monetary prices are exciting to most Kagglers. For instance, Home Depot was offering a winning prize of a whopping $40,000 in search of an algorithm to improve search results on homedepot.com. For most data science enthusiasts, this innovative website is not only a monetary resource, but it is also an indispensable learning tool that helps improve the experience, gain knowledge, elevate and enhance the skills, and learn from mistakes by submitting the code. It is the perfect platform to practice consistently.

The Kaggle community is growing fast. There are currently over one million Kaggle members (Kagglers). This data community has submitted above four million learning models to different competitions. Kaggle users have shared over one thousand datasets, more than 170,000 forum posts and over 250 kernels.

According to the founder, this incredibly fast growth can be attributed to high-quality content, data, and code shared by Kagglers.

Most Kaggle users are committed and active hence the 4,000 forum posts per month and more than 3,500 competition submissions on a daily basis. This platform is the place to be for data scientists and machine learning engineers worldwide (*What Is Kaggle, Why I Participate, What Is the Impact? | Kaggle*, n.d.).

1.3 Google Collab platform

According to the website of Mat Bao company, Google Colab, or Google Colaboratory, has become one of the important and indispensable tools in the modern world of programming and research.

With the power of servers from Google and flexible access from anywhere in the world, Colab has opened a wide door for developers, researchers and practitioners to carry out projects. complex projects, crunch big data, and test machine learning models.

Google Colab, short for Google Colaboratory, is a service that provides a completely online Jupyter Notebook environment. It allows users to create, share and edit notebook files easily without installing any software.

Some application areas of Google Colab include:

- *Machine learning and data science:* Google Colab is a powerful tool for machine learning and data science research. With free access to GPUs and TPUs, users can efficiently build, train, and validate machine learning models.

- *Develop AI and ML applications*: With the power of Colab's GPU and TPU, developers can leverage it to develop artificial intelligence (AI) and machine learning (ML) applications. This includes building and evaluating machine learning models, creating predictive applications, natural language processing, image recognition, and many other AI applications.

- *Research and analyze data:* Researchers and data analysts often use Google Colab to perform data analysis, process big data, and conduct scientific experiments. Colab provides the ability to integrate data from Google Drive or

other sources, allowing them to perform complex analyzes without downloading data to a local machine.

- *Education and training:* Google Colab is also widely used in the field of education and training. It provides an online Python programming environment with no installation required, making it easy for students and teachers to access and share study notebooks. This increases interactivity and community learning in sharing knowledge and projects.

1.4. Anaconda

Besides using the Google Colab platform, we also use Anaconda software to develop supply chain data processing commands on it.

About Anaconda software, this is a platform that distributes Python and R programming languages, serving scientific computing including data science, machine learning, AI, big data processing, analysis and prediction, etc. The purpose is to simplify package management and deployment. This platform is available on Windows, MacOS and Linux.

Anaconda contains all the most popular packages data scientists need. Packages in Anaconda are managed by the Anaconda platform's own manager. We usually use Anaconda to create isolated environments for our projects, use different Python versions or different package versions like Jupyter Lab, Jupyter Notebook etc. and use Anaconda for installation, uninstall and update packages. separately for each project.

1.5. Python language

To carry out this project, our team used the Python programming language. Python is a high-level programming language for general-purpose programming, created by Guido Van Rossum and first released in 1991. Python is designed and widely used because of its strong advantages of being easy to read and understand. learn and remember easily. Widely used in artificial intelligence development and data analysis. Python's structure also allows users to write code with a minimum number of keystrokes. Python is fully dynamically typed and uses

automatic memory allocation, so it is similar to Perl, Ruby, Schema, Smalltalk, and Tcl.

Initially, Python was developed to run on the Unix platform. But over time, Python gradually expanded to all operating systems from Windows to Mac OS, Linux and other operating systems of the Unix family.

As part of the Business Intelligence report, our team used the Python language as the main tool for data analysis.

CHAPTER 2: DATA ANALYSIS

2.1. Definition of data analysis

According to Shamoo and Resnik (2003) Data analysis is the systematic application of statistical and/or logical techniques to understand, summarize, and evaluate data. It involves using various methods to draw meaningful insights, distinguish relevant information from noise, and identify patterns or trends.

In qualitative research, data analysis can involve statistical procedures, but it often takes the form of an ongoing iterative process. Researchers continuously collect and analyze data simultaneously, searching for patterns and themes within observations. The specific qualitative approach and data format (such as field notes, documents, or recordings) shape the analysis methods employed. (Savenye, Robinson, 2004)

Accurate and appropriate data analysis is crucial for maintaining research integrity. Improper statistical analyses can distort research findings, mislead readers, and negatively impact public perception. It is essential to ensure integrity in analyzing both statistical and non-statistical data to uphold the validity and reliability of research outcomes.

2.2. The importance of data analysis in Supply Chain Management

Data analysis plays a crucial role in the field of logistics, offering significant benefits and driving informed decision-making. Here are some key reasons why data analysis is important in logistics

- *Improving operational efficiency:* Data analysis helps identify inefficiencies and bottlenecks in logistics operations. By analyzing data on transportation routes, delivery times, inventory levels, and order processing, companies can optimize their supply chain processes, reduce costs, and improve overall efficiency.

- *Enhancing demand forecasting:* Accurate demand forecasting is essential in the supply chain to ensure optimal inventory levels and minimize stock outs or excess inventory. Data analysis enables companies to analyze historical sales data,

customer behavior, market trends, and external factors to make more accurate demand forecasts, leading to improved inventory management and reduced costs.

- *Optimizing transportation and routing:* Logistic, which is a part of Supply chain, involves managing the movement of goods from one location to another. Data analysis helps identify the most cost-effective transportation routes, optimal delivery schedules, and efficient routing options. This analysis considers factors such as distance, fuel consumption, traffic patterns, and delivery constraints, enabling companies to streamline their transportation operations.

- *Mitigating risks and improving supply chain resilience:* Data analysis helps identify potential risks and vulnerabilities in the supply chain. By analyzing data related to supplier performance, quality control, demand fluctuations, and external factors like natural disasters or geopolitical events, companies can proactively identify and mitigate risks. Data analysis also aids in building a resilient supply chain by enabling companies to develop contingency plans and alternative sourcing strategies.

- *Continuous improvement and decision-making:* Data analysis provides valuable insights for continuous improvement initiatives in logistics. By analyzing key performance indicators (KPIs) and metrics, companies can identify areas for improvement, monitor progress, and make data-driven decisions. This leads to better resource allocation, process optimization, and overall performance enhancement in logistics operations.

2.3. Cleaning Supply Chain Management data using Python language

In fact, the supply chain management data source we collected had quite a few errors. We found quite a lot of data defects as well as some positions in the spreadsheet that were corrupted, leading to data errors. Wrong data display leads to loss of data visualization.

We realized that limitation so we cleaned the data using the Python programming language. Processing data with Python is productive but requires

patience, care, and expertise. We had to use a lot of data cleaning syntax from filtering input data, grouping data into groups and streaming data.

To get the visualization results for tables, figures, and objects on charts, we used batch scanning tricks, but it also included our approach to the subject as the data object. This will be consistent with our data collection and analysis process.

CHAPTER 3: BUSINESS PROBLEM

3.1. Identifying problem

3.1.1. Data operational mechanism problems

Supply chain management analysis is a complex subject. It requires the ability to collect large amounts of data and accurately analyze that data in real time. Not only that, whenever we identify a problem that needs to be analyzed, the data also encounters problems with its operating mechanism.

We need to clearly understand who is the subject of the data, the path of the data, the nodes and statistics that need to be achieved for different types of data.

Supply chain data needs many inputs due to the nature of the operation itself, and each input is a primary key for us to handle data problems. We must balance the processing of input data and data generated during the process of maintaining and operating the supply chain.

3.1.2. Data analysis problems

In today's increasingly complex and interconnected business landscape, organizations face many challenges in effectively managing their supply chains. A pressing issue that continues to plague supply chain managers is the lack of visibility and transparency across the entire supply chain network. Without accurate and timely insights into key performance metrics, such as inventory levels, demand fluctuations, supplier performance and shipping efficiency, businesses will struggle. in making informed decisions and responding effectively to market dynamics.

This lack of visibility often leads to inefficiencies, such as overstocking or out-of-stocks, which can lead to increased shipping costs, missed sales opportunities, and ultimately reduced customer's satisfaction.

Furthermore, without strong data analytics capabilities, organizations will struggle to proactively identify and mitigate supply chain risks, such as disruptions in the flow of raw materials, geopolitical instability or unexpected changes in consumer preferences.

In addition, the global nature of modern supply chains creates additional complexity, including the need to manage diverse suppliers in different regions, navigate regulatory compliance requirements, and maximize Optimize shipping routes to minimize costs and reduce environmental impact. Without a comprehensive understanding of these multifaceted challenges and the ability to harness data-driven insights, organizations will struggle to adapt and thrive in a fast-paced market environment. fast and volatile today.

Thus, the current business problem revolves around the need to analyze effective supply chain management by leveraging data analysis solutions in Business Intelligence. Organizations require analytical tools and methods to extract useful information from vast and disparate supply chain data sources. They need real-time visibility into supply chain operations, predictive capabilities to predict demand fluctuations, and prescriptive analytics to optimize inventory levels, supplier relationships, and customer satisfaction. supply and distribution channels.

Solving this business problem requires a strategic approach that combines technological innovation, data-driven decision making and cross-functional collaboration. Organizations must invest in robust systems capable of integrating data from multiple internal and external sources, deploying advanced analytics algorithms to derive actionable insights and empower supply chain stakeholders with the tools and knowledge needed to drive continuous improvement and resilience.

By effectively addressing the challenge of Supply Chain Management Analytics through the Business Intelligence subject, organizations can improve operational efficiency, reduce costs, mitigate risks and ultimately gain profits. competitive edge in the dynamic and connected world of modern business.

3.2. Solving problem

It can be seen that after our team receives a large amount of data from the Kaggle platform, the data will have many defects, missing, or inconsistent areas. This makes it a bit difficult for us to make the right judgments about the data,

visualize them and represent them smoothly on charts. What's more, the raw data won't give us useful information in which we can come up with a strategy. It shows us the difficulty of solving the problem and when the data is not being well visualized.

For the scope of data analysis, data is the key. Only with data can we build flexible charts and solve outstanding business problems.

In this situation, we build a mechanism to analyze and visualize input data and we should be able to answer questions like Who is the data for? How does this data help the supply chain? Does data impact the management behavior of supply chain management enterprises? Supply chain chain elements and supply chain dynamics?

In other words, we can always plan on how to solve a problem when we have a specific goal of analyzing data in the supply chain. It helps us capture important milestones and moreover the sharp points of the data and the operational structure of the data.

3.3. The need to solve the problem

Business data is the backbone of every business. Whenever business data has problems, they also cause significant disruptions to business operations.

Therefore, we need to solve the problem of enterprise using business data appropriately. Especially for supply chain management and logistics businesses.

Supply chain data is linked and chained. In other words, the current data of the first point will be the theoretical basis for solving problems for the next point of the activity. The more data, the more subjective errors and the more difficult it is to process.

With such a large amount of data, if we manage with Excel, it will cause difficult problems that will hinder the business's ability to manage and operate. A correct data source and properly visualized will help businesses make the right plans. With the raw data we collect, processing them and subsequent steps are

necessary. Not to mention it must be done as quickly as possible to serve as a basis for data processing for subsequent actions.

In supply chain management, each data chain will allow users to implement actions related to it. From there we will have a more accurate view of data planning.

That is why we need to process data in business.

CHAPTER 4: DATA DESCRIPTION (COLLECT DATA)

4.1. Data understanding

To analyze a company's supply chain, we need data on the different stages of the supply chain, like data about sourcing, manufacturing, transportation, inventory management, sales and customer demographics. Nextly, we conduct to analyze this data using Python.

Let's get started with the task of Supply Chain Analysis by importing the necessary Python libraries and the dataset.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Figure 4-1: Import libraries

The code consists of four lines importing different Python libraries:

- The first code line imports the pandas library and assigns it the alias pd. Pandas is a powerful library used for data analysis and manipulation in Python

- The second code line imports the numpy library and assigns it the alias np. NumPy is another fundamental library for scientific computing in Python.

- The third code line imports the matplotlib.pyplot library and assigns it the alias plt. Matplotlib is a popular library for creating static, animated, and interactive visualizations in Python.

- The fourth code line imports the seaborn library and assigns it the alias sns. Seaborn is built on top of matplotlib and helps make statistical data visualizations more attractive.

In essence, this code imports the necessary libraries to perform data analysis, data manipulation, and visualization tasks in Python.

```
data = pd.read_csv(r"C:\Users\loves\Downloads\supply_chain_data2.csv")
```

Figure 4-2: Reading dataset

This code is written in Python and reads data from a CSV file from the panda library.

```
#DATA DESCRIPTION
#Display information about the data types and non-null values of each column
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 24 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Product type             100 non-null    object
 1   SKU                      100 non-null    object
 2   Price                    100 non-null    float64
 3   Availability             100 non-null    int64
 4   Number of products sold  100 non-null    int64
 5   Revenue generated        100 non-null    float64
 6   Customer demographics    69 non-null     object
 7   Stock levels             96 non-null     float64
 8   Lead times               96 non-null     float64
 9   Order quantities         97 non-null     float64
 10  Shipping times           97 non-null     float64
 11  Shipping carriers        100 non-null    object
 12  Shipping costs           100 non-null    float64
 13  Supplier name            100 non-null    object
 14  Location                 100 non-null    object
 15  Lead time                97 non-null     float64
 16  Production volumes       99 non-null     float64
 17  Manufacturing lead time  96 non-null     float64
 18  Manufacturing costs      100 non-null    float64
 19  Inspection results       100 non-null    object
 20  Defect rates             100 non-null    float64
```

Figure 4-3: Display information about the data types and non-null values of each column

The code uses the data.info() method to display a brief summary of the DataFrame. Here is the output breakdown:

- *Data class:* The first line shows the data stored in the pandas DataFrame object.

- *RangeIndex:* This indicates that the DataFrame uses a 0-based integer index to label rows. In this DataFrame, there are 100 rows, labeled from 0 to 99

- *Data Columns:* The output shows a DataFrame with 24 columns. Each column name is listed and information about the data types and non-null values for each column is provided.

For example, the first column has a data type object, which in pandas usually refers to string data. It also shows that there are 100 non-null values, meaning all 100 rows in this column have data.

On the other hand, the "Customer Demographics" column has 69 non-null values, indicating missing (null) values in its 31 rows.

```
data.columns
```

```
Index(['Product type', 'SKU', 'Price', 'Availability',
       'Number of products sold', 'Revenue generated', 'Customer demographics',
       'Stock levels', 'Lead times', 'Order quantities', 'Shipping times',
       'Shipping carriers', 'Shipping costs', 'Supplier name', 'Location',
       'Lead time', 'Production volumes', 'Manufacturing lead time',
       'Manufacturing costs', 'Inspection results', 'Defect rates',
       'Transportation modes', 'Routes', 'Costs'],
      dtype='object')
```

Figure 4-4: Display all the columns of dataset

- Product type: Classification of products.
- SKU: Code of each product type: Unique code assigned to each product for inventory management.
- Price: Selling price of the product.
- Availability: The current status of the product, indicating whether the product is available in stock or not.
- Number of products sold: Number of products sold in a certain period of time.
- Revenue generated: Total revenue generated from product sales.
- Customer demographics: Information about the characteristics of customers who buy products.
- Stock levels: Quantity of product remaining in stock.
- Lead times: The time a customer has to wait from placing an order until receiving the product.
- Order quantities: The number of products ordered in each order.

- Shipping times: Time required to deliver goods from origin to delivery address.
- Shipping carriers: A service company that transports products from origin to destination.
- Shipping costs: Costs associated with shipping the product.
- Supplier name: Name of the partner providing the product.
- Location: The physical location of the product or production site or warehouse.
- Lead time: The time required to complete a specific process from start to finish.
- Production volumes: Number of products produced in a certain period of time.
- Manufacturing lead time: The time required to produce a product from the time raw materials are prepared until the product is finished.
- Manufacturing costs: Costs associated with producing a product.
- Inspection results: Results of the product quality inspection process.
- Defect rates: Rate of defective products during quality inspection.
- Transportation modes: Vehicles used to transport products.
- Routes: The specific route by which a product is transported from origin to destination.
- Costs: Various costs related to managing and operating the product and business.

By looking at the column names, we can infer that this DataFrame is capable of storing data about products, manufacturing processes, shipping logistics, and potential sales information.

```
#Display the top 5 observations of the dataset
data.head()
```

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times	Order quantities	...	Location	Lead time	Production volumes	Manufa lea
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58.0	7.0	96.0	...	Mumbai	29.0	215.0	
1	skincare	SKU1	14.843523	95	736	7460.900065	Female	53.0	30.0	37.0	...	Mumbai	23.0	517.0	
2	haircare	SKU2	11.319683	34	8	9577.749626	NaN	1.0	10.0	88.0	...	Mumbai	12.0	971.0	
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23.0	NaN	59.0	...	Kolkata	24.0	937.0	
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5.0	3.0	NaN	...	Delhi	5.0	414.0	

5 rows × 24 columns

Figure 4-5: Display the top 5 observations of the dataset

```
#Display the last 5 observations of the dataset
data.tail()
```

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times	Order quantities	...	Location	Lead time	Production volumes	Man
95	haircare	SKU95	77.903927	65	672	7386.363944	NaN	15.0	14.0	26.0	...	Mumbai	18.0	450.0	
96	cosmetics	SKU96	24.423131	29	324	7698.424766	Non-binary	NaN	2.0	32.0	...	Mumbai	NaN	648.0	
97	haircare	SKU97	3.526111	56	62	4370.916580	Male	46.0	19.0	4.0	...	Mumbai	10.0	535.0	
98	skincare	SKU98	19.754605	43	913	8525.952560	Female	53.0	1.0	27.0	...	Chennai	28.0	581.0	
99	haircare	SKU99	68.517833	17	627	9185.185829	NaN	55.0	8.0	59.0	...	Chennai	29.0	921.0	

5 rows × 24 columns

Figure 4-6: Display the last 5 observations of the dataset

```
data['Product type'].unique()
```

```
array(['haircare', 'skincare', 'cosmetics'], dtype=object)
```

```
data['Transportation modes'].unique()
```

```
array(['Road', 'Air', 'Rail', 'Sea'], dtype=object)
```

```
data['Routes'].unique()
```

```
array(['Route B', 'Route C', 'Route A'], dtype=object)
```

```
data['Customer demographics'].unique()
```

```
array(['Non-binary', 'Female', 'Male'], dtype=object)
```

```
data['Location'].unique()
```

```
array(['Mumbai', 'Kolkata', 'Delhi', 'Bangalore', 'Chennai'], dtype=object)
```

Figure 4-7: Unique values of each Categorical Variable

This code displays the unique categories contained in several columns of pandas DataFrame.

Lines 1-5: Each line uses the .unique() method on a separate column of DataFrame data.

- *Line 1:* data['Product type'].unique() finds unique categories in the "Product type" column and stores the results in an array.

- *Line 2:* data['Transportation models'].unique() finds unique categories in the "Transportation mode" column and stores the results in an array.

- *Line 3:* data['Routes'].unique() finds unique categories in the "Routes" column and stores the results in an array.

- *Line 4:* data['Customerdemographics'].unique() finds unique categories in the "Customer Demographics" column and stores the results in an array.

- *Line 5:* data['Location'].unique() finds unique categories in the "Location" column and stores the results in an array.

4.2. Statistics summary

```
#STATISTICS SUMMARY
#Provide a statistics summary of data belonging to numerical datatype such as int, float
data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Price	100.0	49.462461	31.168193	1.699976	19.597823	51.239831	77.198228	99.171329
Availability	100.0	48.400000	30.743317	1.000000	22.750000	43.500000	75.000000	100.000000
Number of products sold	100.0	460.990000	303.780074	8.000000	184.250000	392.500000	704.250000	996.000000
Revenue generated	100.0	5776.048187	2732.841744	1061.618523	2812.847151	6006.352023	8253.976921	9866.465458
Stock levels	100.0	47.890000	30.740489	0.000000	21.000000	47.500000	71.500000	100.000000
Order quantities	100.0	48.958763	26.532586	1.000000	26.000000	51.500000	69.500000	96.000000
Shipping times	100.0	5.760000	2.689749	1.000000	4.000000	6.000000	8.000000	10.000000
Shipping costs	100.0	5.548149	2.651376	1.013487	3.540248	5.320534	7.601695	9.929816
Lead time	100.0	17.120000	8.642484	1.000000	10.000000	18.000000	25.000000	30.000000
Production volumes	100.0	561.540000	263.571063	104.000000	332.750000	566.000000	793.250000	985.000000
Manufacturing lead time	100.0	14.530000	8.979725	1.000000	7.000000	13.000000	23.000000	30.000000
Manufacturing costs	100.0	47.266693	28.982841	1.085069	22.983299	45.905622	68.621026	99.466109
Defect rates	100.0	2.277158	1.461366	0.018608	1.009650	2.141863	3.563995	4.939255
Costs	100.0	529.245782	258.301696	103.916248	318.778455	520.430444	763.078231	997.413450
Profit Margin	100.0	86.066707	11.083219	46.051103	82.222943	89.540644	93.196058	98.219015

Figure 4-8: Provide a statistics summary of data belonging to numerical data type such as int, float

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Product type	100	3	skincare	40	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SKU	100	100	SKU0	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Price	100.0	NaN	NaN	NaN	49.462461	31.168193	1.699976	19.597823	51.239831	77.198228	99.171329
Availability	100.0	NaN	NaN	NaN	48.4	30.743317	1.0	22.75	43.5	75.0	100.0
Number of products sold	100.0	NaN	NaN	NaN	460.99	303.780074	8.0	184.25	392.5	704.25	996.0
Revenue generated	100.0	NaN	NaN	NaN	5776.048187	2732.841744	1061.618523	2812.847151	6006.352023	8253.976921	9866.465458
Customer demographics	100	3	Female	40	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Stock levels	100.0	NaN	NaN	NaN	47.89	30.740489	0.0	21.0	47.5	71.5	100.0
Order quantities	100.0	NaN	NaN	NaN	48.958763	26.532586	1.0	26.0	51.5	69.5	96.0
Shipping times	100.0	NaN	NaN	NaN	5.76	2.689749	1.0	4.0	6.0	8.0	10.0
Shipping carriers	100	3	Carrier B	43	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Shipping costs	100.0	NaN	NaN	NaN	5.548149	2.651376	1.013487	3.540248	5.320534	7.601695	9.929816
Supplier name	100	5	Supplier 1	27	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Location	100	5	Kolkata	25	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Lead time	100.0	NaN	NaN	NaN	17.12	8.642484	1.0	10.0	18.0	25.0	30.0
Production volumes	100.0	NaN	NaN	NaN	561.54	263.571063	104.0	332.75	566.0	793.25	985.0
Manufacturing lead time	100.0	NaN	NaN	NaN	14.53	8.979725	1.0	7.0	13.0	23.0	30.0
Manufacturing costs	100.0	NaN	NaN	NaN	47.266693	28.982841	1.085069	22.983299	45.905622	68.621026	99.466109
Inspection results	100	3	Pending	41	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Defect rates	100.0	NaN	NaN	NaN	2.277158	1.461366	0.018608	1.00965	2.141863	3.563995	4.939255
Transportation modes	100	4	Road	29	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Routes	100	3	Route A	43	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Costs	100.0	NaN	NaN	NaN	529.245782	258.301696	103.916248	318.778455	520.430444	763.078231	997.41345
Profit Margin	100.0	NaN	NaN	NaN	86.066707	11.083219	46.051103	82.222943	89.540644	93.196058	98.219015

Figure 4-9: Provide a statistics summary of all data, include object, category, etc

```
import statistics
```

```
#Calculate descriptive statistics (Mean)
mean_ = statistics.mean(data['Revenue generated'])
mean_
```

```
5776.048187380086
```

```
#Calculate descriptive statistics (Median)
median_ = statistics.median(data['Price'])
median_
```

```
51.239830500537565
```

```
#Calculate descriptive statistics (Mode)
mode_ = statistics.mode(data['Product type'])
mode_
```

```
'skincare'
```

Figure 4-10: Calculate Descriptive Statistics

CHAPTER 5: DATA CLEANING/WRANGLING

5.1. Check for duplication

```
#DATA CLEANING
#Check for Duplication
data.nunique()
```

```
Product type                3
SKU                       100
Price                     100
Availability               63
Number of products sold    96
Revenue generated         100
Customer demographics       3
Stock levels               63
Lead times                 29
Order quantities           59
Shipping times             10
Shipping carriers           3
Shipping costs            100
Supplier name               5
Location                    5
Lead time                  29
Production volumes         95
Manufacturing lead time    29
Manufacturing costs       100
Inspection results          3
Defect rates              100
Transportation modes        4
Routes                      3
Costs                     100
dtype: int64
```

Figure 5-1: Check for Duplication

This code indicates that the data may be inconsistent and needs to be cleaned before further analysis. The data cleaning process can include techniques such as removing duplicates, correcting missing items, and standardizing formats.

Overall, the code points to potential data quality issues in the dataset. Further cleaning steps will likely be required to ensure data accuracy and reliability for any subsequent analysis.

5.2. Missing value

```
#Missing Values Calculation
data.isnull().sum()
```

```
Product type                0
SKU                         0
Price                       0
Availability                0
Number of products sold     0
Revenue generated           0
Customer demographics      31
Stock levels                4
Lead times                  4
Order quantities            3
Shipping times              3
Shipping carriers           0
Shipping costs              0
Supplier name               0
Location                    0
Lead time                   3
Production volumes          1
Manufacturing lead time     4
Manufacturing costs         0
Inspection results          0
Defect rates                0
Transportation modes        0
Routes                      0
Costs                       0
dtype: int64
```

Figure 5-2: Missing Values Calculation

The code appears to be written in Python and it looks like a function to calculate missing values in a data set

This code calculates the number of missing values present in each column of the Pandas DataFrame.

This information can be useful for data cleaning tasks such as deciding how to handle missing data (e.g., truncate, delete rows/columns).

```
#Calculate the percentage of missing values in each column
(data.isnull().sum()/(len(data)))*100
```

```
Product type                0.0
SKU                         0.0
Price                       0.0
Availability                0.0
Number of products sold     0.0
Revenue generated           0.0
Customer demographics      31.0
Stock levels                4.0
Lead times                  4.0
Order quantities            3.0
Shipping times              3.0
Shipping carriers           0.0
Shipping costs              0.0
Supplier name               0.0
Location                    0.0
Lead time                   3.0
Production volumes          1.0
Manufacturing lead time     4.0
Manufacturing costs         0.0
Inspection results          0.0
Defect rates                0.0
Transportation modes        0.0
Routes                      0.0
Costs                       0.0
dtype: float64
```

Figure 5-3: Calculate the percentage of missing values in each column

```
#Filling null values with Forward fill
forward_fill = data['Customer demographics'].fillna(method='ffill')
print("\nForward Fill:")
print(forward_fill)
```

```
Forward Fill:
0     Non-binary
1         Female
2         Female
3     Non-binary
4     Non-binary
         ...
95          Male
96    Non-binary
97          Male
98        Female
99        Female
Name: Customer demographics, Length: 100, dtype: object
```

Figure 5-4: Filling null values with Forward fill

```
data["Customer demographics"].fillna(forward_fill, inplace = True)
data
```

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times	Order quantities	...	Location	Lead time	Production volumes	Man
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58.0	7.0	96.0	...	Mumbai	29.0	215.0	
1	skincare	SKU1	14.843523	95	736	7460.900065	Female	53.0	30.0	37.0	...	Mumbai	23.0	517.0	
2	haircare	SKU2	11.319683	34	8	9577.749626	Female	1.0	10.0	88.0	...	Mumbai	12.0	971.0	
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23.0	NaN	59.0	...	Kolkata	24.0	937.0	
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5.0	3.0	NaN	...	Delhi	5.0	414.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	

Figure 5-5: Filling null values with Forward fill data "Customer demographics"

```
#Filling null values with Linear interpolation
linear_interpolation = data['Stock levels'].interpolate(method='linear')
data["Stock levels"].fillna(linear_interpolation, inplace = True)
data
```

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Order quantities	Shipping times	...	Location	Lead time	Production volumes	I
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58.0	96.0	4.0	...	Mumbai	29.0	215.0	
1	skincare	SKU1	14.843523	95	736	7460.900065	Female	53.0	37.0	2.0	...	Mumbai	23.0	517.0	
2	haircare	SKU2	11.319683	34	8	9577.749626	Female	1.0	88.0	2.0	...	Mumbai	12.0	971.0	
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23.0	59.0	6.0	...	Kolkata	24.0	937.0	
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5.0	NaN	8.0	...	Delhi	5.0	414.0	

Figure 5-6: Filling null values with Linear interpolation

```
#Filling null values with Mean imputation
mean_imputation = data['Order quantities'].fillna(data['Order quantities'].mean())
data["Order quantities"].fillna(mean_imputation, inplace = True)
data
```

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Order quantities	Shipping times	...	Location	Lead time	Production volumes
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58.0	96.000000	4.0	...	Mumbai	29.0	215.0
1	skincare	SKU1	14.843523	95	736	7460.900065	Female	53.0	37.000000	2.0	...	Mumbai	23.0	517.0
2	haircare	SKU2	11.319683	34	8	9577.749626	Female	1.0	88.000000	2.0	...	Mumbai	12.0	971.0
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23.0	59.000000	6.0	...	Kolkata	24.0	937.0
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5.0	48.958763	8.0	...	Delhi	5.0	414.0

Figure 5-7: Filling null values with Mean imputation

```
#Filling null values with Median imputaion
median_imputation = data['Shipping times'].fillna(data['Shipping times'].median())
data["Shipping times"].fillna(median_imputation, inplace = True)
data
```

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Order quantities	Shipping times	...	Location	Lead time	Production volumes	I
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58.0	96.000000	4.0	...	Mumbai	29.0	215.0	
1	skincare	SKU1	14.843523	95	736	7460.900065	Female	53.0	37.000000	2.0	...	Mumbai	23.0	517.0	
2	haircare	SKU2	11.319683	34	8	9577.749626	Female	1.0	88.000000	2.0	...	Mumbai	12.0	971.0	
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23.0	59.000000	6.0	...	Kolkata	24.0	937.0	
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5.0	48.958763	8.0	...	Delhi	5.0	414.0	

Figure 5-8: Filling null values with Median imputation

```
#Filling null values with Backward Fill
backward_fill = data['Production volumes'].fillna(method='bfill')
data["Production volumes"].fillna(backward_fill, inplace = True)
data
```

```
C:\Users\loves\AppData\Local\Temp\ipykernel_13104\3915649346.py:2: FutureWarning: Series.fillna with 'method' is deprecated and will r
aise in a future version. Use obj.ffill() or obj.bfill() instead.
  backward_fill = data['Production volumes'].fillna(method='bfill')
```

er of :ts ld	Revenue generated	Customer demographics	Stock levels	Order quantities	Shipping times	...	Location	Lead time	Production volumes	Manufacturing lead time	Manufacturing costs	Inspection results	Defect rates	T
02	8661.996792	Non-binary	58.0	96.000000	4.0	...	Mumbai	29.0	215.0	29.0	46.279879	Pending	0.226410	
36	7460.900065	Female	53.0	37.000000	2.0	...	Mumbai	23.0	517.0	30.0	33.616769	Pending	4.854068	
8	9577.749626	Female	1.0	88.000000	2.0	...	Mumbai	12.0	971.0	27.0	30.688019	Pending	4.580593	
83	7766.836426	Non-binary	23.0	59.000000	6.0	...	Kolkata	24.0	937.0	18.0	35.624741	Fail	4.746649	
71	2686.505152	Non-binary	5.0	48.958763	8.0	...	Delhi	5.0	414.0	3.0	92.065161	Fail	3.145580	

Figure 5-9: Filling null values with Backward Fill

```
#Filling null values with Mode imputation
mode_imputation = data['Manufacturing lead time'].fillna(data['Manufacturing lead time'].mode().iloc[0])
data["Manufacturing lead time"].fillna(mode_imputation, inplace = True)
data
```

Figure 5-10: Filling null values with Mode imputation

5.3. Data reduction

```
#DATA REDUCTION
#Remove 'Lead times' column from data
data = data.drop(['Lead times'], axis = 1)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 23 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Product type             100 non-null    object
 1   SKU                      100 non-null    object
 2   Price                    100 non-null    float64
 3   Availability             100 non-null    int64
 4   Number of products sold  100 non-null    int64
 5   Revenue generated        100 non-null    float64
 6   Customer demographics    100 non-null    object
 7   Stock levels             96 non-null     float64
 8   Order quantities         97 non-null     float64
 9   Shipping times           97 non-null     float64
 10  Shipping carriers        100 non-null    object
 11  Shipping costs           100 non-null    float64
 12  Supplier name            100 non-null    object
 13  Location                 100 non-null    object
 14  Lead time                97 non-null     float64
 15  Production volumes       99 non-null     float64
 16  Manufacturing lead time  96 non-null     float64
 17  Manufacturing costs      100 non-null    float64
 18  Inspection results       100 non-null    object
 19  Defect rates             100 non-null    float64
```

Figure 5-11: Remove 'Lead times' column from data

This code is a simple function designed to reduce the number of columns in a Pandas DataFrame.

This function is useful for dimensionality reduction tasks or for selecting a specific subset of features from a dataset.

```
data.isnull().sum()
```

```
Product type              0
SKU                       0
Price                     0
Availability              0
Number of products sold   0
Revenue generated         0
Customer demographics     0
Stock levels              0
Order quantities          0
Shipping times            0
Shipping carriers         0
Shipping costs            0
Supplier name             0
Location                  0
Lead time                 0
Production volumes        0
Manufacturing lead time   0
Manufacturing costs       0
Inspection results        0
Defect rates              0
Transportation modes      0
Routes                    0
Costs                     0
dtype: int64
```

Figure 5-12: Checking null values after cleaning the dataset

This code may be part of a larger data cleaning or preprocessing process.

The is_null_check function helps determine if there are missing values in specific columns of the DataFrame.

The result of this function (number of missing values in each column) can be used to decide how to handle missing data (e.g., imputation, row/column deletion).

5.4. Feature Engineering

```
#FEATURE ENGINEERING
#Create new feature 'Profit Margin'
data['Profit Margin'] = ((data['Revenue generated'] - (data['Manufacturing costs'] + data['Shipping costs'] + data['Costs']))/data['Re
data.head()
```

ck els	Order quantities	Shipping times	...	Lead time	Production volumes	Manufacturing lead time	Manufacturing costs	Inspection results	Defect rates	Transportation modes	Routes	Costs	Profit Margin
3.0	96.000000	4.0	...	29.0	215.0	29.0	46.279879	Pending	0.226410	Road	Route B	187.752075	97.264043
3.0	37.000000	2.0	...	23.0	517.0	30.0	33.616769	Pending	4.854068	Road	Route B	503.065579	92.676501
1.0	88.000000	2.0	...	12.0	971.0	27.0	30.688019	Pending	4.580593	Air	Route C	141.920282	98.113724
3.0	59.000000	6.0	...	24.0	937.0	18.0	35.624741	Fail	4.746649	Rail	Route A	254.776159	96.238746
5.0	48.958763	8.0	...	5.0	414.0	3.0	92.065161	Fail	3.145580	Air	Route A	923.440632	62.054927

Figure 5-13: Create new feature "Profit Margin"

The table shows the features used to create the new feature called "Profit Margin". The new feature is the calculation of several other features in the dataset including Revenue generated, Manufacturing costs, Shipping costs, Costs.

The new feature "Profit Margin" can support data analysis tasks such as classification or linear and regression modeling. It can be used to understand how much profit is generated per unit of revenue after accounting for various costs.

CHAPTER 6: EXPLORATORY DATA ANALYSIS (EDA)

6.1. EDA univariate analysis

```
#UNIVARIATE ANALYSIS
#Visualize Numerical variables using a histogram and a box plot
for col in num_cols:
    print(col)
    print('Skew :', round(data[col].skew(), 2))
    plt.figure(figsize = (15, 4))
    plt.subplot(1, 2, 1)
    data[col].hist(grid=False)
    plt.ylabel('count')
    plt.subplot(1, 2, 2)
    sns.boxplot(x=data[col])
    plt.show()
```

Figure 6-1: Visualize Numerical variables using a histogram and a box plot (code)

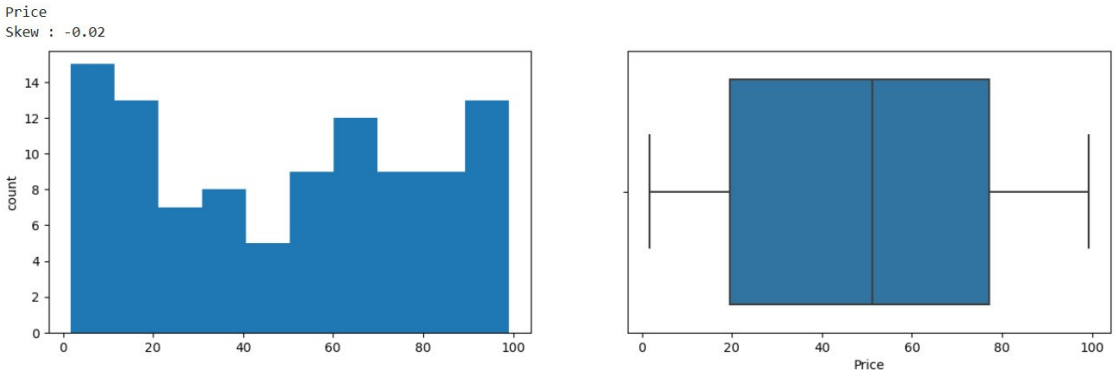

Figure 6-2: Visualize Numerical variables using a histogram and a box plot (Price)

The chart shows a visualization of the count chart with the price chart. As we can interpret from the count chart, the figure indicates that the number of samples that was counted started off at 14 which was the peak. It then fell dramatically to around 6 when it was at the 20th count. It levels out until 40 which experienced a small dip then rose up to 12 when the figure passed 60. From around 70 to 90, the figure kept the count of around 8 then ended at 13 when it reached 100.

The price chart shows a straightforward display of information. After 100 samples were given, it is clear that the price of the vehicle can range from 20 to a little less than 80.

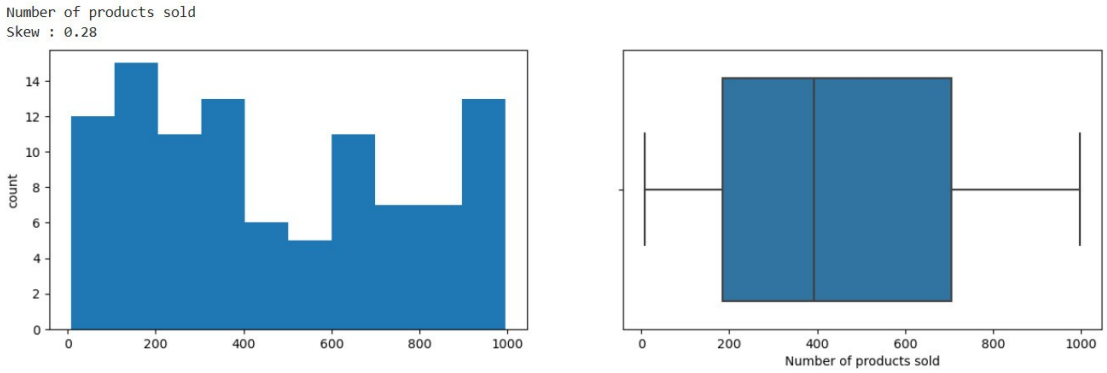

Figure 6-3: Visualize Numerical variables using a histogram and a box plot (Number of products sold)

The 2 charts indicate the sample that was recorded and the figure for the number of products sold. Overall, there is an overwhelming amount of samples that was counted in these charts.

On the count chart, the count started off at 12 samples then experienced a peak at 14 which then back to around 12 at the 200 mark. The figure then rose slightly to 13 at about 350 just to dipped heavily to 4 at 500 samples. It increased back to 11 but then decreased and leveled out at 6 from 500 to 900 samples. The figure ended at 12 for the final 1000 samples.

The number of products sold that was recorded from the second chart shows that companies sell around about 200 up to slightly above 700 cars.

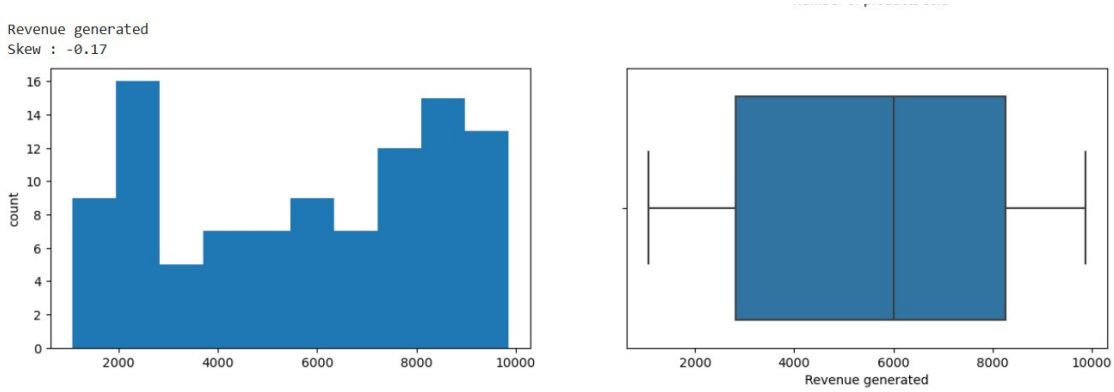

Figure 6-4:Visualize Numerical variables using a histogram and a box plot (Revenue generated)

Similar to the last chart, this chart shows the number that was recorded in 10000 count and the second chart shows the revenue that was generated from these samples.

In the first chart, the counted figure started off at 9 then peaked immediately to 16 at 2000 samples. After the peak, it fell drastically down to its lowest point, which is about 4, then stabilized at 6 until about slightly less than 6000. It did experience a slight increase to 8 but quickly went back to 6. it rose significantly to 14 at the approximately 9000 samples then ended at 13 on the 10000 samples.

The revenue that was shown indicates that companies can generate about 3000 averagely at the lowest rate up to 8000 at the highest rate.

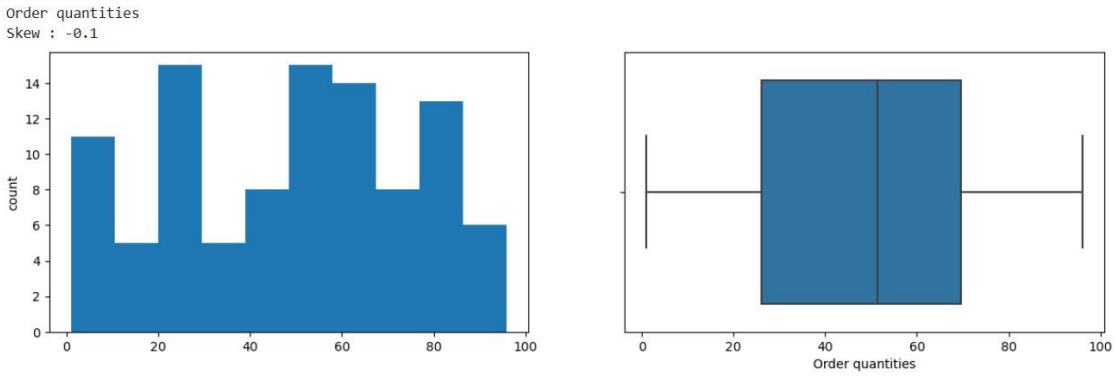

Figure 6-5: Visualize Numerical variables using a histogram and a box plot (Order quantities)

Moving on to the order charts, as you can see from the count chart, it quite fluctuated in terms of samples recorded. Overall there are 100 times counted and a sizable amount of ups and downs.

The count charge began at around 11 samples recorded but then fell down quite hard to nearly 4. After that, the figure immediately ascended and peaked at more than 14 recorded samples after passing the 20 count. It experienced a downfall back to 4 after the peak, however, it quickly rose back to more than 14 after a slight increase to 8. It then slightly dropped to exactly 14 which then plummeted down to 8 after the 60 counts mark. At 80 count, the amount of samples recorded jumped to roughly about 13, ending at a descent of 6 at the 100 count.

The second chart shows the average order quantities of recorded samples, the range of which was around a little less than 30 to almost 70 orders.

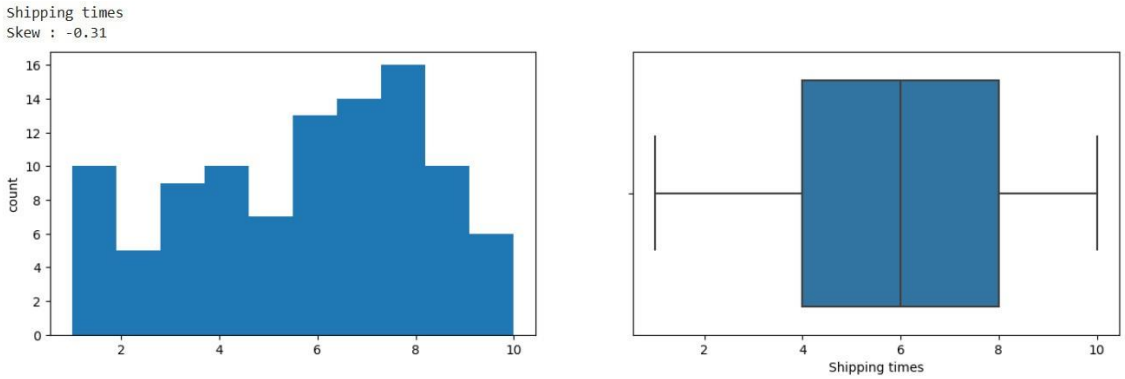

Figure 6-6: Visualize Numerical variables using a histogram and a box plot (Shipping times)

Moving on to the order charts, as you can see from the count chart, it quite fluctuated in terms of samples recorded. Overall there are 100 times counted and a sizable amount of ups and downs.

The count charge began at around 11 samples recorded but then fell down quite hard to nearly 4. After that, the figure immediately ascended and peaked at more than 14 recorded samples after passing the 20 count. It experienced a downfall back to 4 after the peak, however, it quickly rose back to more than 14 after a slight increase to 8. It then slightly dropped to exactly 14 which then plummeted down to 8 after the 60 counts mark. At 80 count, the amount of samples recorded jumped to roughly about 13, ending at a descent of 6 at the 100 count.

The second chart shows the average order quantities of recorded samples, the range of which was around a little less than 30 to almost 70 orders.

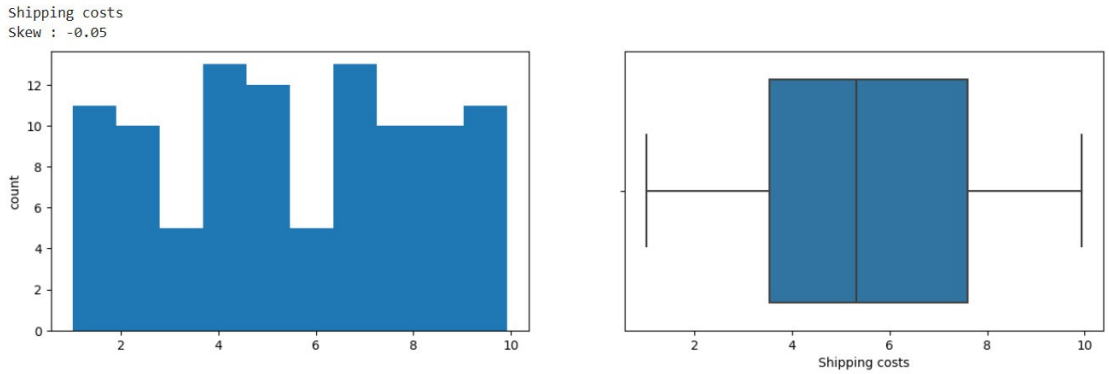

Figure 6-7: Visualize Numerical variables using a histogram and a box plot (Shipping costs)

Like the previous chart, there were a limited number of samples that were recorded for the shipping cost which was only 10 counts.

The number of recorded samples across the chart was quite high across the chart, starting off at approximately 11. Right at the 3rd count, the figure experienced its first downward fall to about 5 but immediately hit its first peak at more than 12 at the 4th counts. It fell off slightly before hitting its second lowest point at 5 samples on the 6th count. The figure rose to its second peak at more than 12 then leveled out at slightly less than 10 samples to the 9th count, ending with 10 samples on the 10th count.

The second chart illustrates the shipping cost of the documented samples. The chart shows that the lowest average cost is a bit more than 3 while the highest is just under 8.

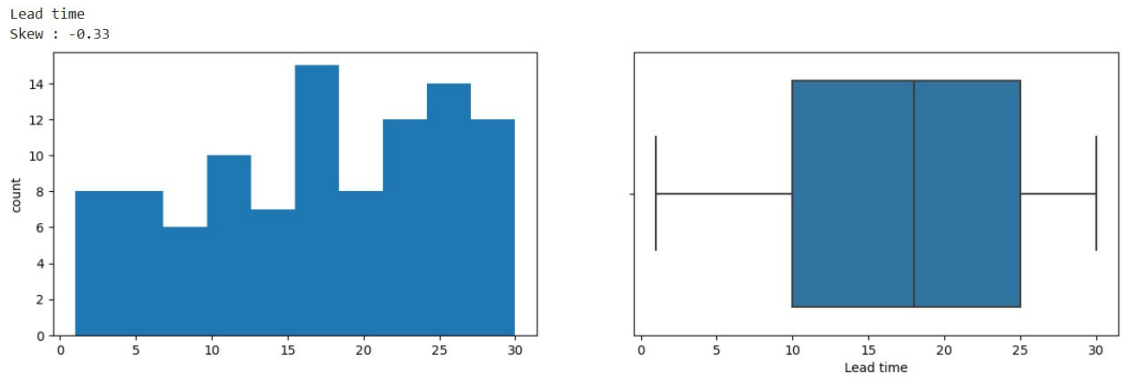

Figure 6-8: Visualize Numerical variables using a histogram and a box plot (Lead time)

The next 2 charts illustrate the lead time of the recorded companies. From the count hart, it can be interpreted that there was an upward trend in the recorded samples

The count chart's figure first started off at 8 on the first few counts then it would keep roughly the same figure until the 15th count. It only experienced a decrease of 2 and an increase to 10 on the 9th and 11th count respectively. It reached its peak at more than 14 samples recorded then plummeted back to 8 at the 20th count. The number of samples surged to 14 passing the 25th count mark then ended at 30 documented samples at the 30th count.

The second presents the average lead time of those companies. The lead time is average at 10 to 25.

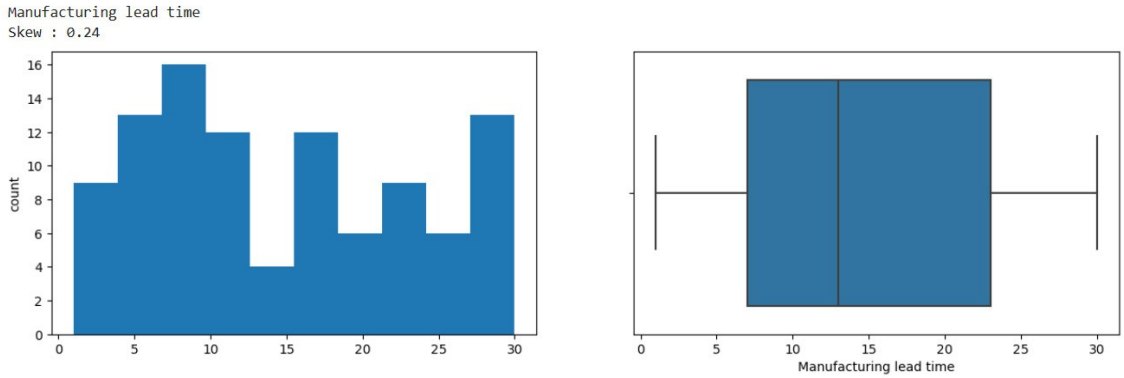

Figure 6-9: Visualize Numerical variables using a histogram and a box plot (Manufacturing lead time)

The next 2 charts have the information of manufacturing time that was cataloged from the sampled companies. Overall, the count chart provided the same amount of counts as the previous chart.

On the first chart, there were about 8 samples that were collected on the first count and the sample number increased to 13 on the 5th count. It heightened to its peak of 16 on the 7th count just before plummeting down significantly to 4 passing the 10th count. On the 15th count, the sample's figure rose back to 12 right before meeting a downward trend to 6 from 20 to 25 counts. At the end of the 30th count, the figure has a sample size of 12.

The second chart shows us details about the approximate manufacturing lead time. The chart suggests that the average time can be from about 6 up to 24.

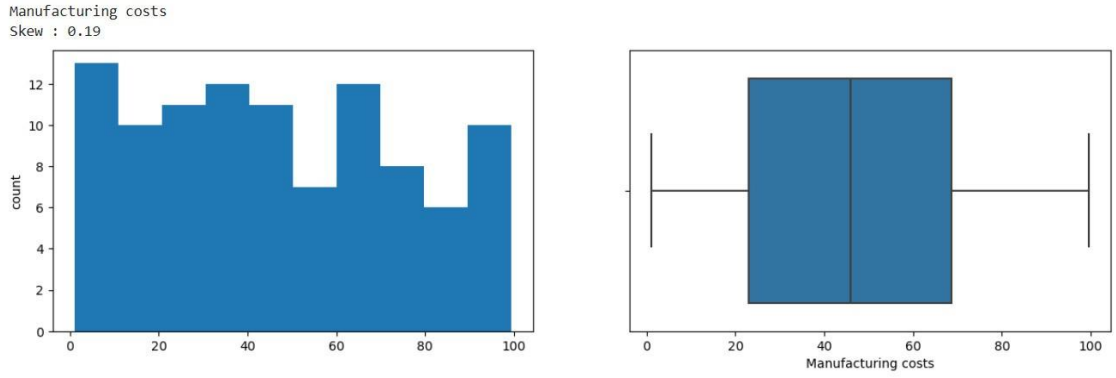

Figure 6-10: Visualize Numerical variables using a histogram and a box plot (Manufacturing costs)

The next 2 charts describe the manufacturing cost of several sample companies. Even though there were 100 counts of samples, each count contains less samples than other previous charts.

The amount of samples started at its peak of 12 samples on the first count then only decreased slightly to about 10 on the 20th count. The figure stabilized at that number until the 50th count which experienced a decline to 6 samples. A significant rise to 12 was soon followed at the 60th count but quickly decreased down to 8 before the 80th count and to 6 afterwards. At the end of the chart, the figure has a number of 9 on the 100th count.

It can be seen from the second chart that the average manufacturing cost can be around slightly above 20 up to about 70.

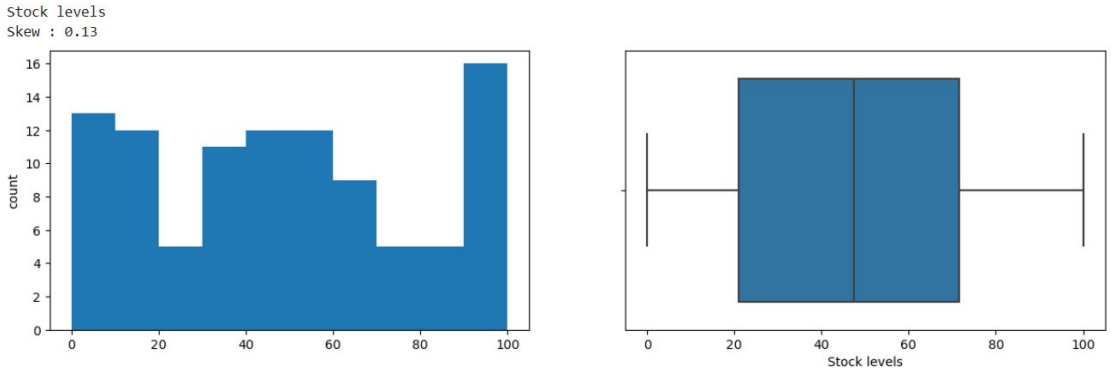

Figure 6-11: Visualize Numerical variables using a histogram and a box plot (Stock levels)

Stock levels charts are the next on the list and as we can see from the chart, the sample size is much less from the previous chart.

The count figure started off at about 13 then maintained about the same to 12 before dipped heavily to 4 at the 20 mark. The count rose back to less than 12 samples and stabilized at 12 from 40 to 60 count. The sample decreased slightly to 10 after passing the 60 count and then moved down back to 4 from about 50 to 90 count. At the 100 count, the figure was at its highest in the chart which is 16, ending the count.

The next chart indicates clearly about the stock level of 100 companies, it ranges from 20 to about 70 stock levels.

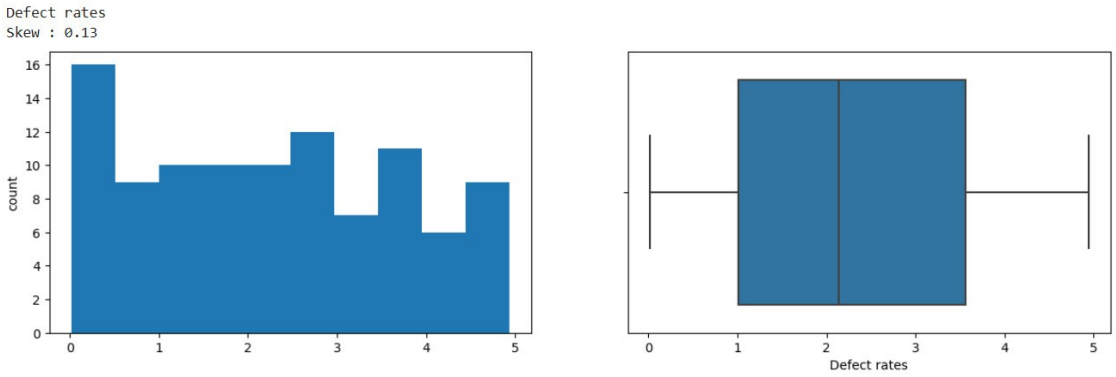

Figure 6-12: Visualize Numerical variables using a histogram and a box plot (Defect rates)

The chart is a combination of a histogram and a boxplot, both of which are used to visualize the distribution of data.

The chart shows the frequency of the error rate. The x-axis represents the error rate, and the y-axis represents the number of items with that error rate. In this case, the error rate ranges from 0 to 5. The most common error rate is 1, with 21 items having that error rate.

Overall, these are some of the key observations from the chart:

- There are more items with a defect rate of 1 than any other defect rate.
- Error rates are clustered around 1, with some exceptions at 0 and 3.
- The average error rate is 1.
- There are more errors with a ratio lower than 1 than higher than 1 (because the cell is skewed to the right).

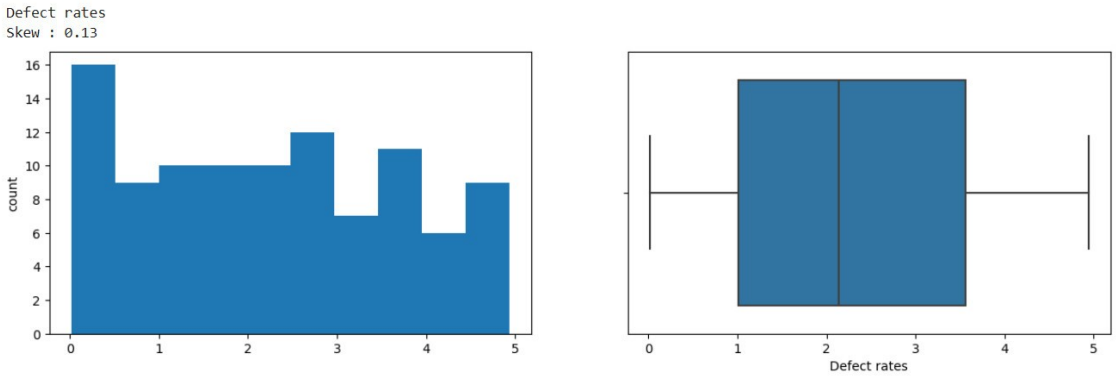

Figure 6-13: Visualize Numerical variables using a histogram and a box plot (Defect rates)

The next charts describe the defect rates of multiple sampled companies. As we can see from the chart, there were not a significant amount of samples recorded, only about 5 counts that are present in the chart.

On the first count the figure started at its highest which is 16 samples and moved down heavily to 8. From 1st count to almost 3, it remained the same at about 9 and increased slightly to- 12. The figure fluctuated with a downward trend of about 6 to 12 from the 3rd count to the final count which ended at 10.

The boxplot summarizes the distribution of the defect rates. The box in the center of the boxplot represents the middle 50% of the data. The line in the middle of the box is the median, which is the defect rate that separates the lower half of the data from the upper half. In this case, the median defect rate is 1. The lines extending from the box (whiskers) represent the rest of the data, up to 1.5 times the interquartile range (IQR) from the top and bottom of the box. The IQR is the difference between the 75th percentile and the 25th percentile. In this case, the whiskers extend from 0 to 3.

To summarize, it is clear that there were not a lot of defects that were produced. On average, there were only about 1 to about 4 defective products.

```
In [72]: sns.countplot(data=slc_df, x='Shipping carriers')
Out[72]: <Axes: xlabel='Shipping carriers', ylabel='count'>
```

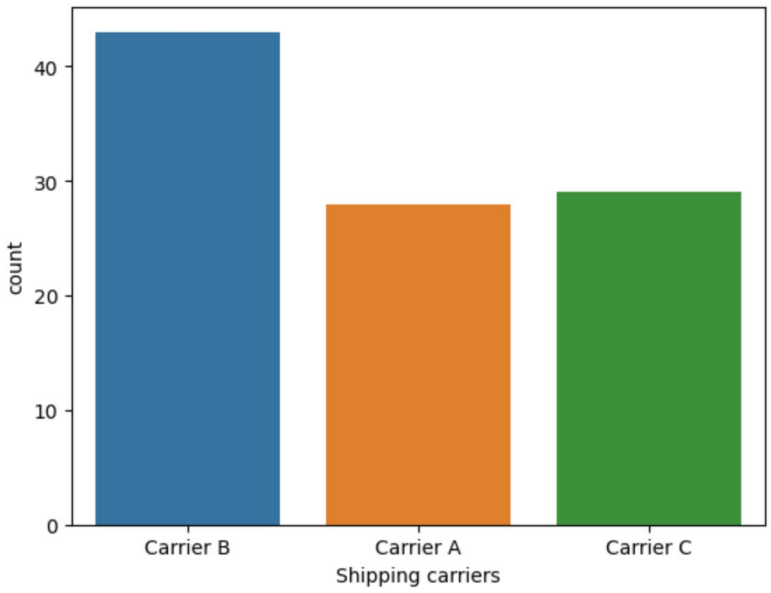

Figure 6-14: Shipping carriers count

The next chart indicates information of 3 shipping carriers, namely A, B and C, along with their counts. It can be interpreted from the chart and carrier B has the highest count which is above 40. While, carrier A and carrier C have roughly the same amount which is approximately 30.

```
In [34]: sns.countplot(data=slc_df, x='Location')
Out[34]: <Axes: xlabel='Location', ylabel='count'>
```

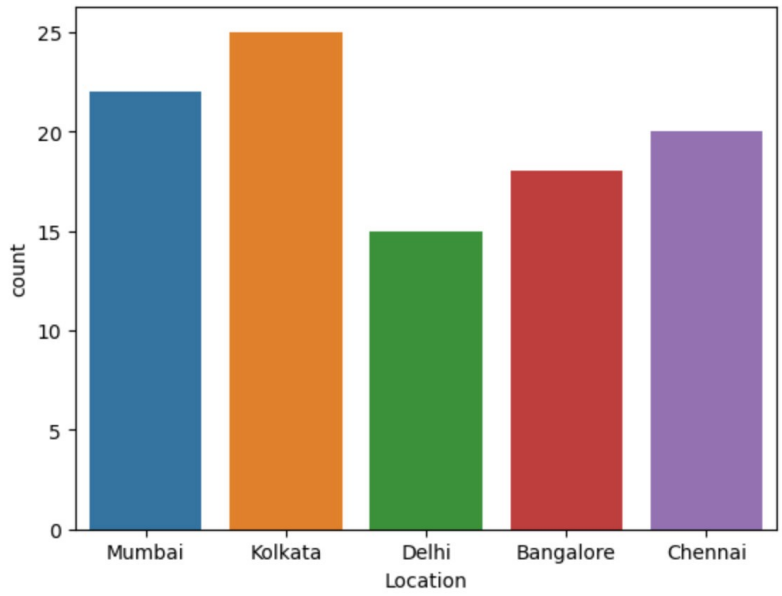

Figure 6-15: Locations count

The next chart provides recorded counts in several cities in India namely Mumbai, Kolkata, Delhi location, Bangalore and Chennai. The lowest count that was documented belongs to the Delhi location which is only about 15 counts.

Following that is the city of Bangalore and Chennai with about 17 and 20 respectively. The city with the second highest count is Mumbai with more than 20 counts and the city with the highest count is Kolkata.

```
In [74]: sns.countplot(data=slc_df, x='Customer demographics')
Out[74]: <Axes: xlabel='Customer demographics', ylabel='count'>
```

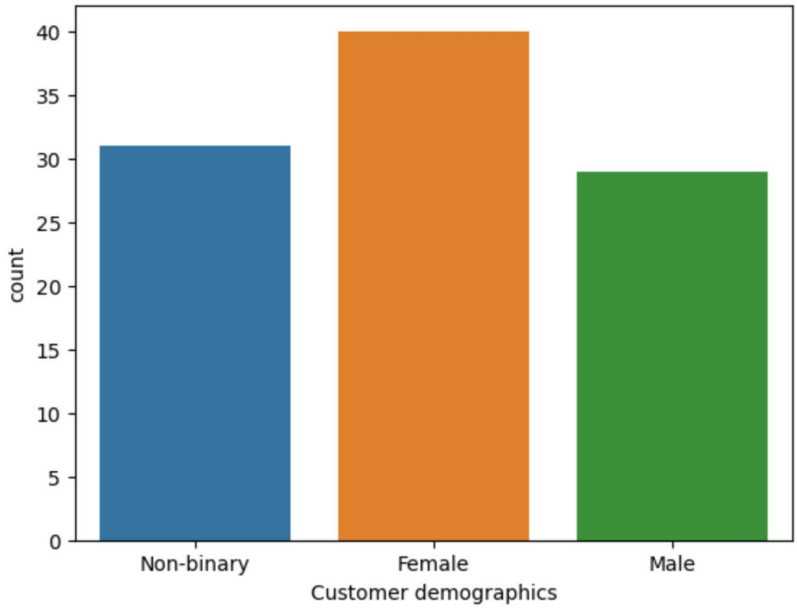

Figure 6-16: Customer demographics count

The next is the chart containing the information of customer demographics and there are 3 bars representing non-binary, female and male. It is clear from the chart that the majority of demographics is female buyers having 40 counts. Next up are non-binary individuals and male buyers which are quite similar at around 30 counts.

```
In [73]: sns.countplot(data=slc_df, x='Product type')
Out[73]: <Axes: xlabel='Product type', ylabel='count'>
```

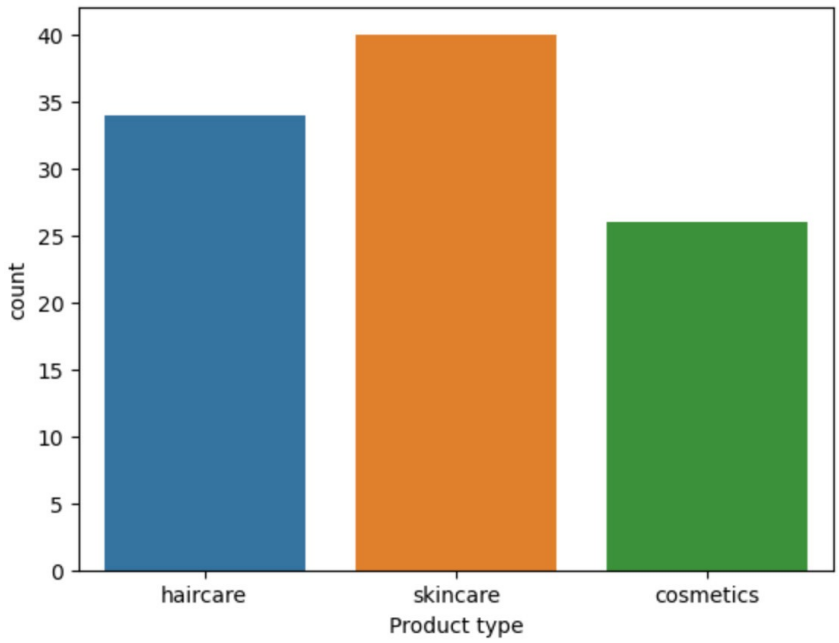

Figure 6-17: Product type count

Product type is the information that was provided in the next chart, there are 3 main product types and those are haircare, skincare and cosmetics. From the chart's information, skincare products have the highest counts of all 3 that is 40. Below that is haircare, the figure for that is 35. Cosmetics has a count of 25 which is the lowest in this chart.

```
In [75]: sns.countplot(data=slc_df, x='Supplier name')
Out[75]: <Axes: xlabel='Supplier name', ylabel='count'>
```

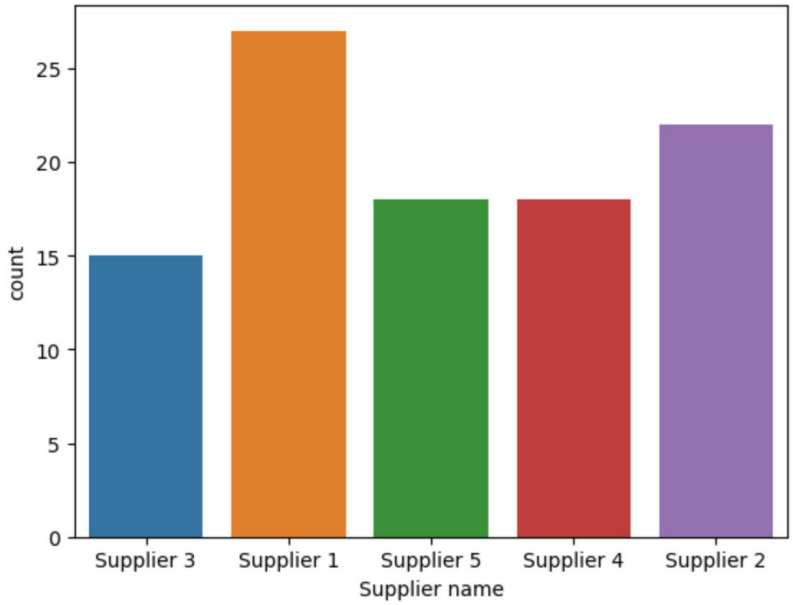

Figure 6-18: Supplier name count

The following chart provides information on the counts of suppliers, there were 5 suppliers that were documented. First, supplier number 3 has the lowest count which is only 15. Supplier 1 contains the most amount of count, which is more than 25 counts. Suppliers 5 and 4 are exactly identical with the count of about 18. Lastly, supplier 2 with the second highest amount of counts staying at more than 20.

```
In [76]: sns.countplot(data=slc_df, x='Inspection results')
Out[76]: <Axes: xlabel='Inspection results', ylabel='count'>
```

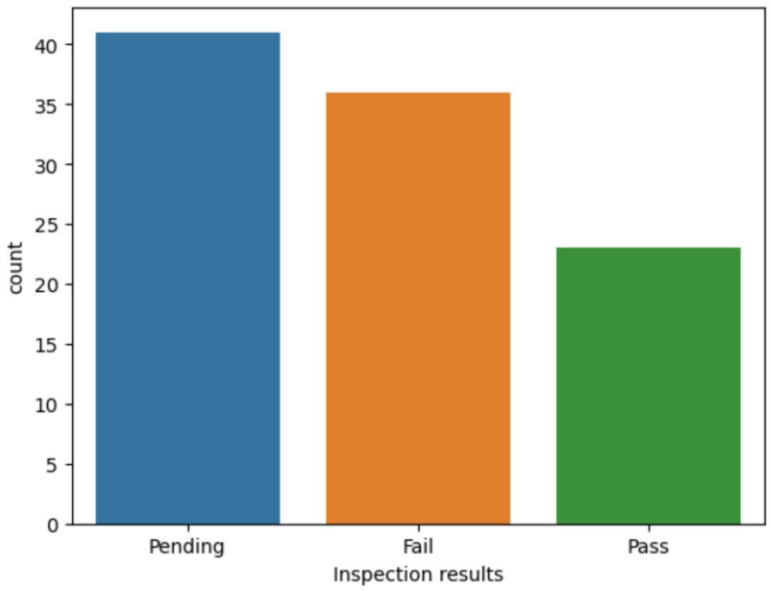

Figure 6-19: Inspection results count

The next chart contains the information of frequency of inspection. There are 3 results that can occur after an inspection that are pending, fail and pass. According to the chart, most of the inspection results are not included because of the pending counts of 40 which is also the highest figure in this chart. Next, the fail results are more frequent than passes which are 35 and 25 respectively.

```
In [77]: sns.countplot(data=slc_df, x='Transportation modes')
Out[77]: <Axes: xlabel='Transportation modes', ylabel='count'>
```

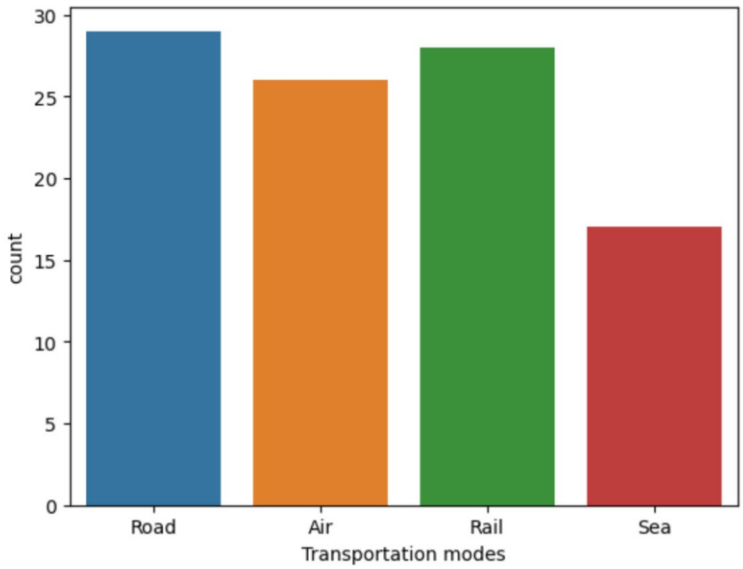

Figure 6-20: Transportation modes count

The chart now displays the mode of transportation that is most commonly used. In this chart, there are 4 main modes namely road, air, rail and sea. The most popular mode is road that has the figure of almost 30 following with rail which is only a little less. Next figure is the statistic for air transport having 25 counts and the least common mode takes up only 15 counts.

```
In [78]: sns.countplot(data=slc_df, x='Routes')
Out[78]: <Axes: xlabel='Routes', ylabel='count'>
```

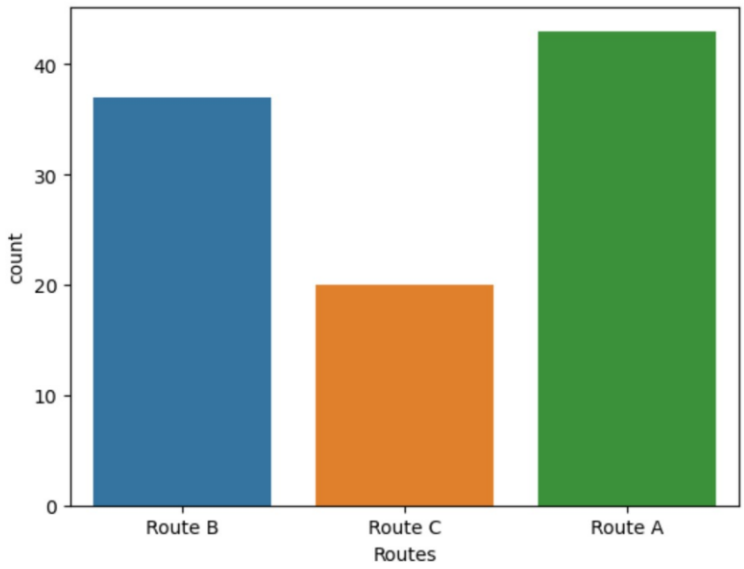

Figure 6-21: Routes count

The last frequency chart is the frequency of routes. There are 3 routes that are mainly used by companies and those are route A, B and C. It is noticeable that route A is the most commonly utilized route, having more than 40 counts in total. Route B is not too far away from route A in terms of popularity, consisting of almost 40 counts. Lastly, route C has the least amount of counts that is only about 20.

```
#Sales by product type
sales_data = data.groupby('Product type')['Number of products sold'].sum().reset_index()
sales_data
```

	Product type	Number of products sold
0	cosmetics	11757
1	haircare	13611
2	skincare	20731

Figure 6-22: Sales by Product type calculation

The table above illustrates the total number of products sold for 3 product types that was mentioned on a previous chart. Starting off with cosmetics, it is the least sold product, only having the number of 11757 products sold. Next is haircare

which only sold 1854 more than cosmetics. The highest figure belongs to skincare products, selling 20731 items.

```
pie_chart = px.pie(sales_data, values='Number of products sold', names='Product type',
                   title='Sales by Product Type',
                   hover_data=['Number of products sold'],
                   hole=0.45,
                   color_discrete_sequence=px.colors.qualitative.Pastel)
pie_chart.update_traces(textposition='inside',textinfo='percent')
pie_chart.show()
```

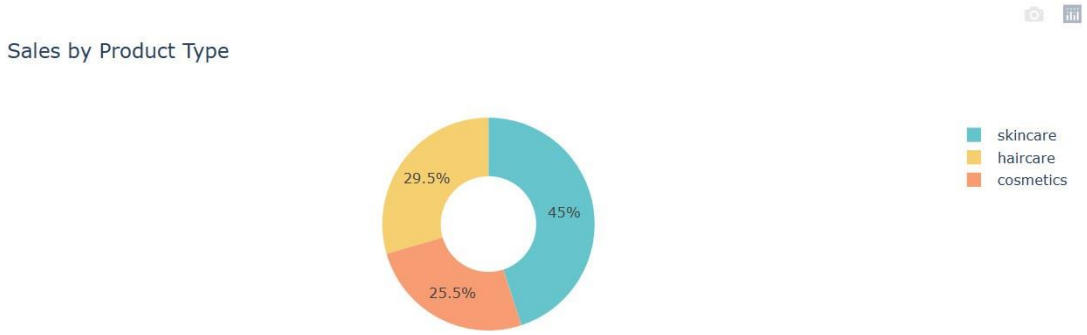

Figure 6-23: Sales by Product type (Pie chart visualization)

The next up is the pie chart representing the percentage of sales by product type skincare, haircare and cosmetics. The product type that takes almost half the percentage, which is 45%, is skincare. The rest belongs to cosmetics and hair care which are approximately the same in statistics, 25.5% and 29.5% respectively.

6.2. Data transformation

```
# Function for log transformation of the column
def log_transform(data,col):
    for colname in col:
        if (data[colname] == 1.0).all():
            data[colname + '_log'] = np.log(data[colname]+1)
        else:
            data[colname + '_log'] = np.log(data[colname])
    data.info()
```

```
log_transform(data,['Number of products sold','Price'])
```

Figure 6-24: Function for log transformation of the column

```
#Log transformation of the feature 'Number of products sold'
sns.distplot(data["Number of products sold"], axlabel="Number of products sold");
```

Figure 6-25: Log transformation of the feature 'Number of products sold' (code)

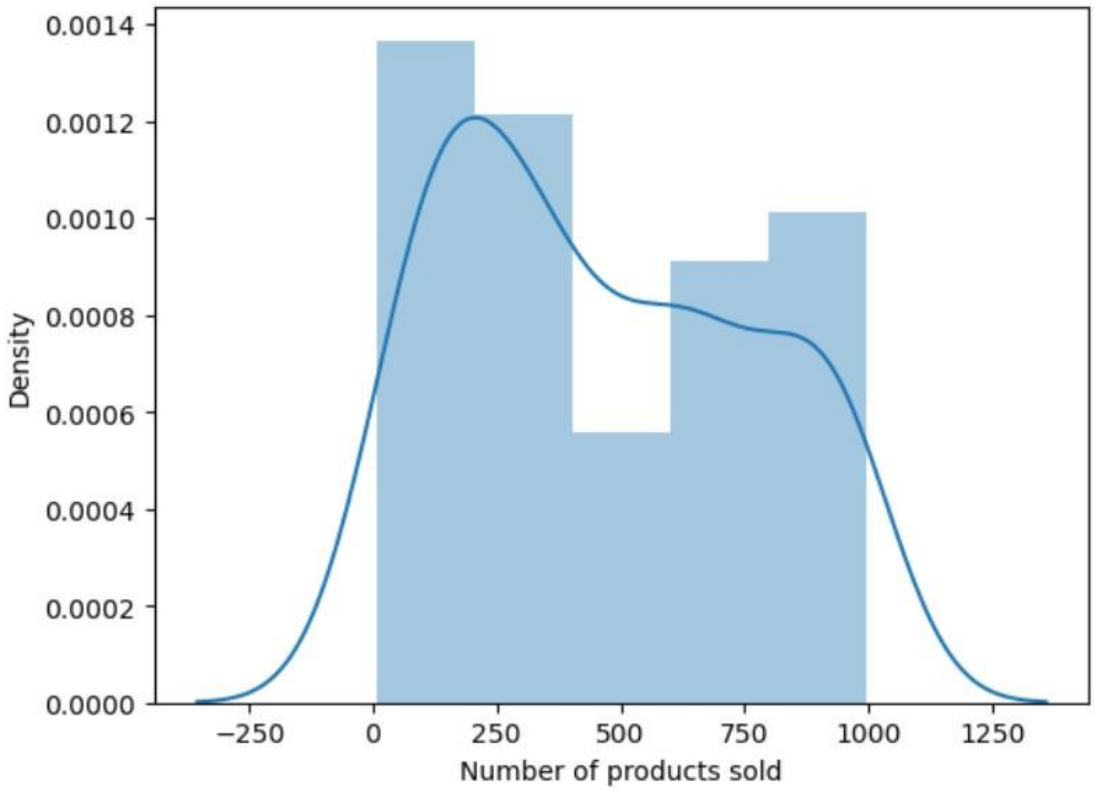

Figure 6-26: Log transformation of the feature 'Number of products sold' (visualization)

The chart is a density chart titled "Number of products sold". The x-axis of the chart shows the number of products sold and the y-axis shows the density.

The density plot shows a distribution with a peak of about 250 products sold. This means the business sells about 250 products.

Overall, these are some of the key observations from the chart:

- The data may represent sales data from a business that sells a variety of products through a network of retail stores.
- The majority of businesses may sell about 250 products.

```
#Log transformation of the feature 'Price'
sns.distplot(data["Price"], axlabel="Price");
```

Figure 6-27: Log transformation of the feature "Price" (code)

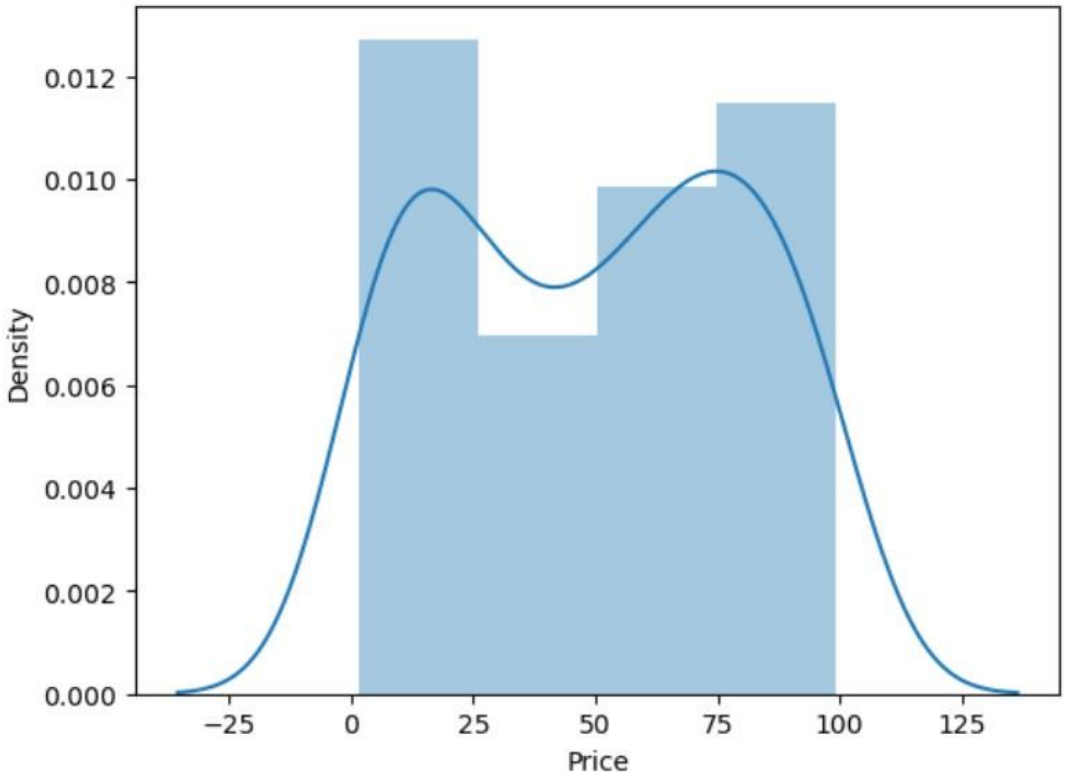

Figure 6-28: Log transformation of the feature "Price" (visualization)

The chart is a density plot which is a type of estimated probability density function. It shows the distribution of values for a single variable. The x-axis shows price and the y-axis shows density.

In this particular density chart, the price appears to be between $0.00 and $0.10. The highest density is about $0.005. This means there are more data points clustered around $0.005 than any other price level.

Overall, it may be said that most of products cost around $0.005.

6.3. EDA bivariate analysis

```
In [53]: total_revenue = slc_df.groupby('Shipping carriers')['Revenue generated'].sum().reset_index()

fig = go.Figure()
fig.add_trace(go.Bar(x=total_revenue['Shipping carriers'],
                     y=total_revenue['Revenue generated']))
fig.update_layout(title='Total revenue by shipping carrier',
                  xaxis_title='Shipping carrier',
                  yaxis_title='Revenue Generated')
fig.show()
```

Figure 6-29: Total revenue by shipping carrier

The next bar chart contains the information of total revenue generated by 3 different shipping carriers namely A, B and C. As we can see from the chart, carrier B generated the most amount of value in all 3, making up to 250.000. Carrier C generates second highest revenue but significantly less than carrier B, making about almost 200.000.

```
#Total revenue generated from each states
total_revenue = data.groupby('Location')['Revenue generated'].sum().reset_index()

fig = go.Figure()
fig.add_trace(go.Bar(x=total_revenue['Location'],
                     y=total_revenue['Revenue generated']))
fig.update_layout(title='Total Revenue based on location',
                  xaxis_title='Locations',
                  yaxis_title='Revenue generated')
fig.show()
```

Figure 6-30: Total Revenue based on locations

The next chart displays the information of revenue generated in 5 indian cities namely Bangalore, Chennai, Delhi, Kolkata and Mumbai. First, the cities that generated the most revenue are Kolkata and Mumbai, both having the figure of 150k. Following that is the city of Chennai, generating over 100k in revenue. Bangalore's figure is exactly at 100k and the lowest is Delhi which is only around 75k.

```
#Analyze the revenue generated by each SKU
revenue_chart = px.line(data,x='SKU',y='Revenue generated',
                        title='Revenue Generated by SKU')
revenue_chart.show()
```

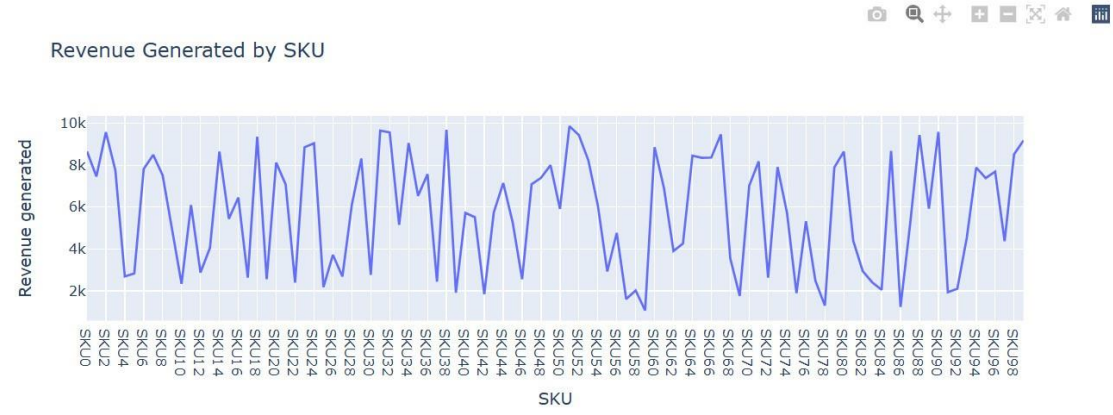

Figure 6-31: Revenue Generated by SKU

The next chart illustrates the revenue generated by SKU, the chart as a whole is quite fluctuating and the revenue can vary between less than 2k to 10k. There are a total of 34 products which are named SKU0, SKU2, SKU4 to SKU98.

```
#Order quantity of each SKU
order_quantity_chart = px.bar(data,x ='SKU',
                              y='Order quantities',
                              title='Order Quantity by SKU')
order_quantity_chart.show()
```

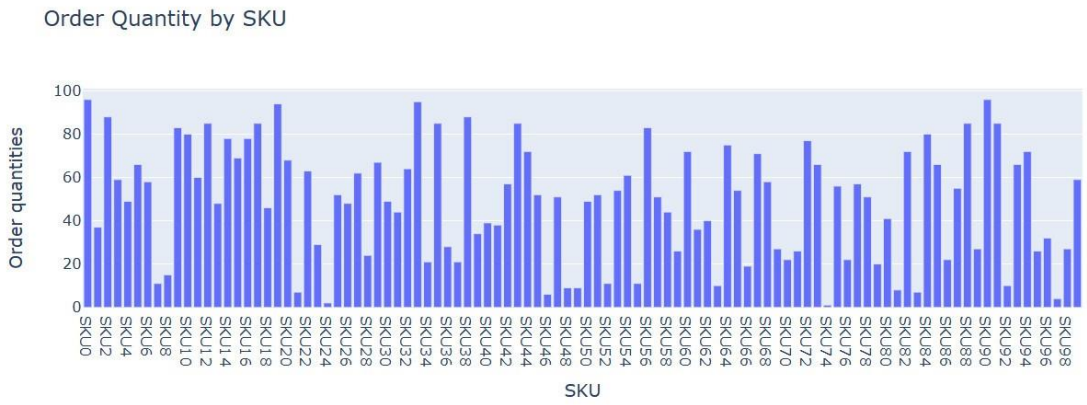

Figure 6-32: Order quantities of each SKU

The next chart represents the order quantity by SKU which are the same 34 SKU products that were mentioned on the previous chart. Overall, there is a diverse amount of ordering quantities across the chart, it could be close to 0 or some are at 100 orders.

Some notable products are SKU0, SKU18 and SKU90 have very significant numbers of orders.

```
In [67]: total_revenue = slc_df.groupby('Transportation modes')['Revenue generated'].sum().reset_index()

fig = go.Figure()
fig.add_trace(go.Bar(x=total_revenue['Transportation modes'],
                     y=total_revenue['Revenue generated']))
fig.update_layout(title='Total Revenue based on transportation modes',
                  xaxis_title='Transportation modes',
                  yaxis_title='Revenue generated')
fig.show()
```

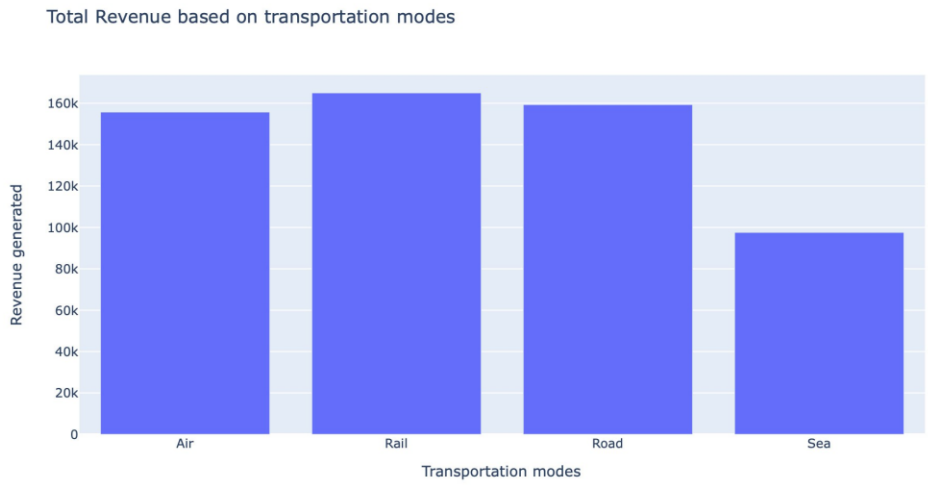

Figure 6-33: Total Revenue based on transportation modes

The following chart interprets the revenue generated via transportation types. Overall, more transportation modes roll in a significant amount of money beside the sea mode.

First, the highest revenue count belongs to rail transportation, making upwards of 160k. Not less than that, the figure for air and road modes are identical in value which is 140k. Last type is sea transportation, only generating about 100k.

```
#Average defect rate of all product types
defect_rate_by_product = data.groupby('Product type')['Defect rates'].mean().reset_index()

fig = px.bar(defect_rate_by_product, x='Product type', y='Defect rates',
             title='Average defect rates by product type')
fig.show()
```

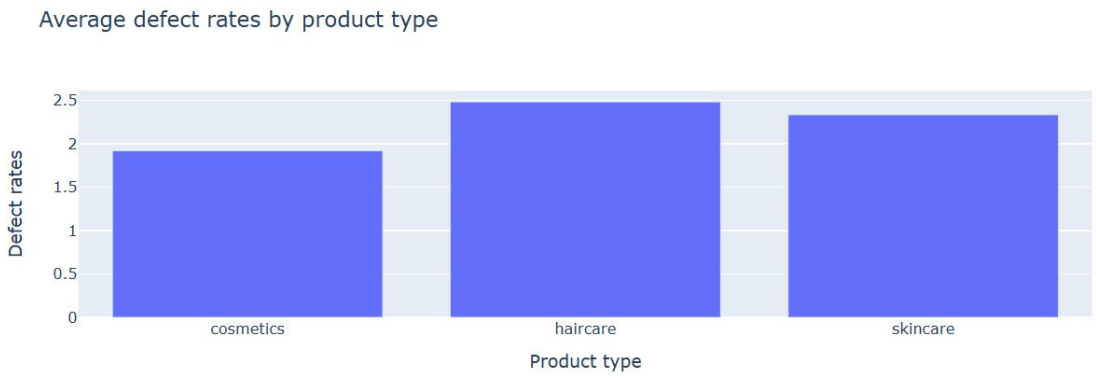

Figure 6-34: Average defect rates by product type

Moving on, we have a bar chart showing the defect rates of 3 product types namely cosmetics, hair care, and skincare. Generally, defect rates are quite low, only ranging at about 0-2.5%.

Starting off with the product type that has the least faulty products, that is cosmetics with the percentage of 2%. Next on the line is skincare products having near 2.5% defect rate and hair care products having the most defected product rate of exactly 2.5%.

```
#Total revenue generated from Product types
total_revenue = data.groupby('Product type')['Revenue generated'].sum().reset_index()

fig = go.Figure()
fig.add_trace(go.Bar(x=total_revenue['Product type'],
                     y=total_revenue['Revenue generated']))
fig.update_layout(title='Total Revenue based on Product type',
                  xaxis_title='Product type',
                  yaxis_title='Revenue generated')
fig.show()
```

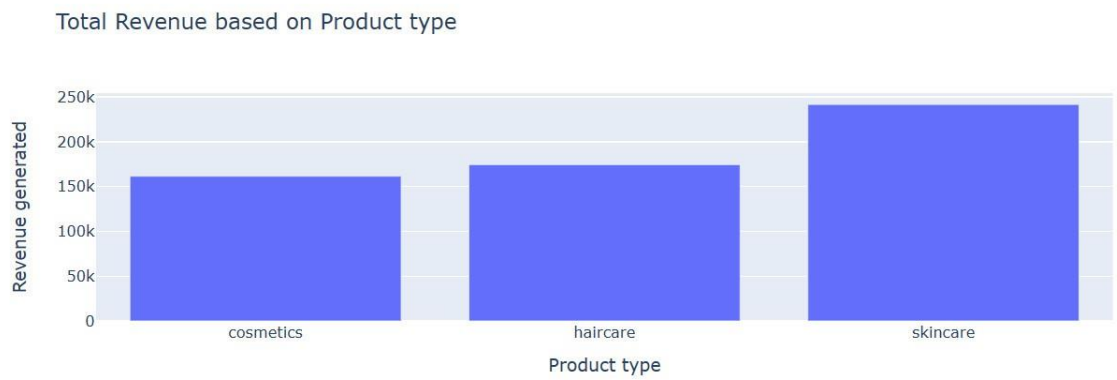

Figure 6-35: Total Revenue based on Product type

Here is the presentation of the total revenue based on product type. There are 3 product types that generate a revenue flow and those are cosmetics, hair care and skin care.

First the least profitable type is cosmetics, only made 150k which is only slightly worse than hair care products. Hair care is the second highest but only generated slightly more than 150k. The highest revenue count is skin care products, the figure is dramatically higher than the 2 previous products making almost 250k.

```
In [49]: sns.countplot(data=slc_df, hue='Location', x='Transportation modes')
Out[49]: <Axes: xlabel='Transportation modes', ylabel='count'>
```

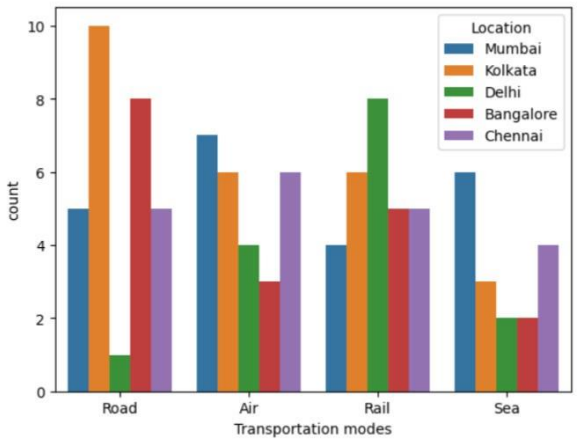

Figure 6-36: Transportation modes count for each location

The chart illustrates the information of the most popular modes of transport in 5 Indian cities. Overall, there are 5 main modes that were documented and the numbers vary from city to city.

First, the road mode is the most common in Kolkata having 10 counts. Following that is Bangalore at 8 counts then Chennai and Mumbai where both contain 5 counts. Road is the least common method of transport which only got about 1 count. Next up is air, Mumbai is the city where air is the most popular method, having 7 counts, and following not too far are Kolkata and Chennai at 6 counts. Delhi and Bangalore have the most minute figures in this category, 4 and 3 respectively.

Rail is quite popular in the city Delhi containing 8 counts, which is a bit more than the second highest figure that is Kolkata at 6. Bangalore and Chennai have the same amount of counts and the lowest is Mumbai, all of which were roughly the same in counts. The last mode is sea and it is used in Mumbai the most, having 6 counts. 4 counts is the number of Chennai of which is the second most widely used in. Finally, Kolkata, Delhi, and Bangalore have roughly the same metric of about 2, with Kolkata having slightly more counts.

```
In [102]: trans_data = slc_df.groupby('Transportation modes')['Number of products sold'].sum().reset_index()

pie_chart = px.pie(slc_df,values='Number of products sold', names='Transportation modes',
                   title='Number of products sold through shipping modes',
                   hover_data=['Number of products sold'],
                   hole=0.5,
#                    color_discrete_sequence=px.colors.qualitative.Pastel
                   )
pie_chart.update_traces(textposition='inside',textinfo='percent+label')
pie_chart.show()
```

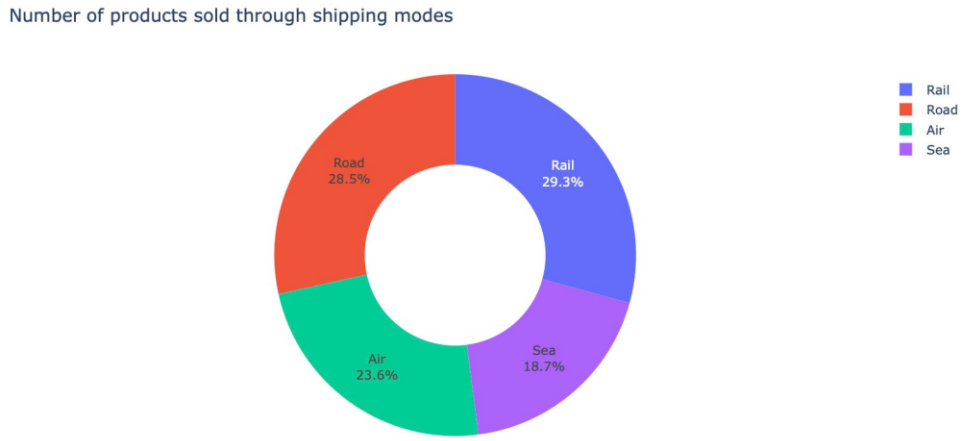

Figure 6-37: Number of products sold through shipping modes

The pie chart represents the percentage of products sold through multiple modes of transportation. The statistics are quite evenly distributed across all 4 modes.

First, road and rail are the 2 modes that the most products are sold through, 28.5% and 29.3% respectively. Air takes the third most percentage, 23.6%, and the least product sold is through sea, which is only 18.7%.

x

```
In [59]: stock_chart = px.line(slc_df,x='SKU',
                               y='Stock levels',
                               title='Stock Levels by SKU')
         stock_chart.show()
```

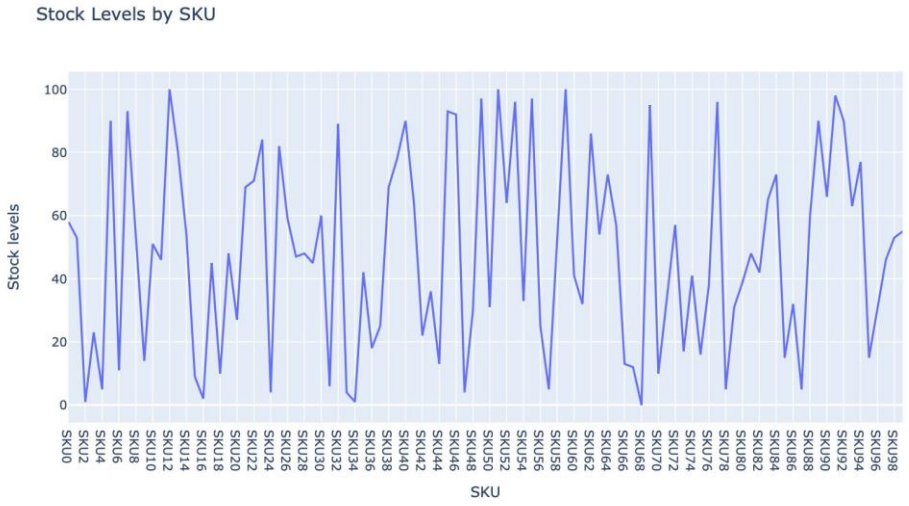

Figure 6-38: Stock Levels by SKU

The x-axis of the chart shows the SKU, and the y-axis shows the stock level. SKU98 has the highest stock level.

Overall, these are some of the key observations from the chart:

- There may be fluctuations in demand for certain SKUs.
- There may be supply chain issues affecting the availability of certain SKUs.
- The business may be in the process of phasing out certain SKUs and no longer producing them.

Businesses can investigate why inventory levels are uneven across different SKUs. This can involve sales data, production data, and inventory data.

Businesses can adjust their production plans or purchasing methods to meet demand for certain SKUs.

Businesses may consider conducting a product review to identify any SKUs that are no longer profitable or in demand.

```
In [62]: transportation_chart = px.pie(slc_df,
                                       values='Costs',
                                       names='Transportation modes',
                                       title='Cost Distribution by Transportation Mode',
                                       hole=0.5,
                                       color_discrete_sequence=px.colors.qualitative.Pastel)
         transportation_chart.update_traces(textposition='inside',textinfo='percent+label')
         transportation_chart.show()
```

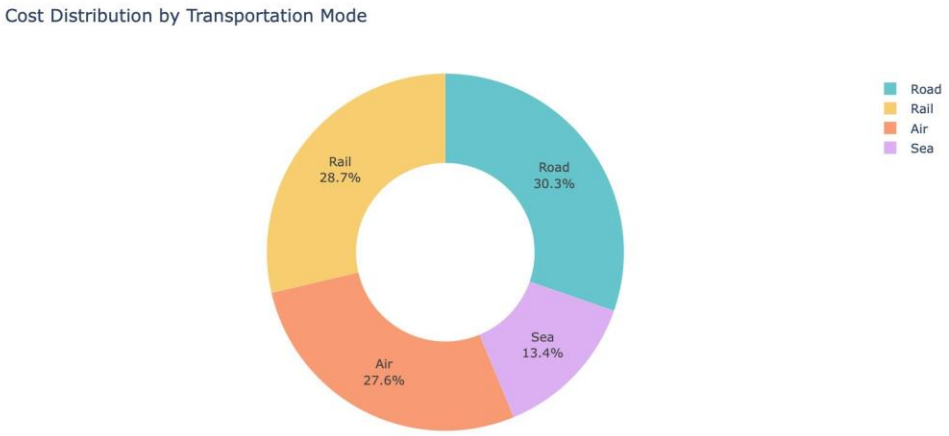

Figure 6-39: Cost Distribution by Transportation Mode

The pie chart displays the information regarding the cost distribution by transportation mode. There are four main modes that take up the expenses, rail, road, air and sea.

First, road is the one that takes up the most costs out of all four, 30.3% of the chart. Following that, rail and air take roughly the same amount of cost, 28.7% and 27.6% respectively. The last one is air, taking up the smallest amount of money to operate, only 13.4% of all expenses.

```
In [64]: trans = slc_df.groupby('Transportation modes')['Defect rates'].mean().reset_index()

trans_chart = px.pie(trans,values='Defect rates',
                     names='Transportation modes',
                     title='Defect Rates by Transportation Mode',
                     hole=0.5,
                     color_discrete_sequence=px.colors.qualitative.Pastel)
trans_chart.update_traces(textposition='inside',textinfo='label+percent')
trans_chart.show()
```

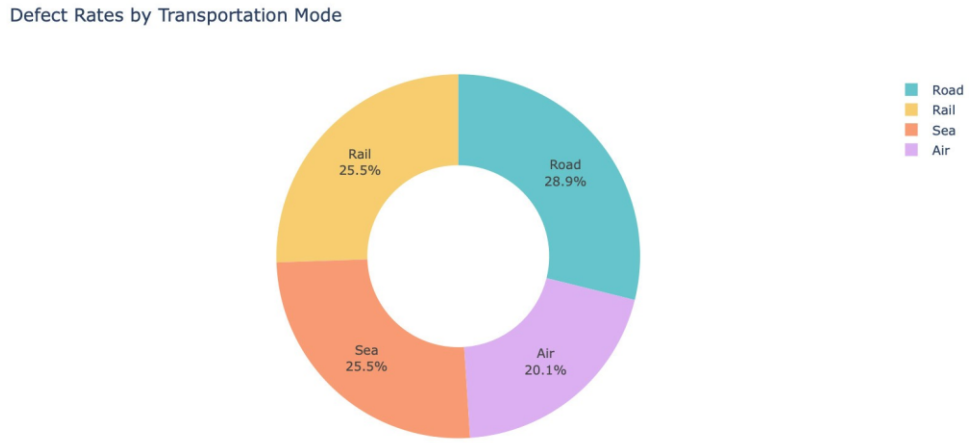

Figure 6-40: Defect Rates by Transportation Mode

Following that is the statistic for defect rates that occur on 4 modes of transportation. Overall, defected products are detected evenly across all 4 modes.

First, the rate for sea and rail are exactly identical in percentage, at 25.5% for both of them. The mode with the highest defect rates is road at 28.9% and contrast that, the lowest defect rate that was recorded is 20.1% for air.

```
In [127]: trans1 = slc_df.groupby('Location')['Costs'].mean().reset_index()

trans_chart = px.pie(trans1,values='Costs',
                     names='Location',
                     title='Allocate costs across locations',
                     hole=0.5,
                     color_discrete_sequence=px.colors.qualitative.Pastel)
trans_chart.update_traces(textposition='inside',textinfo='label+percent')
trans_chart.show()
```

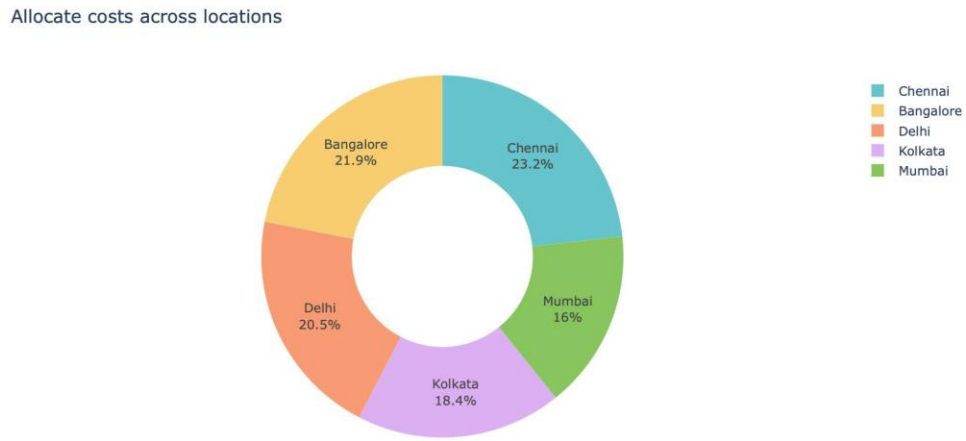

Figure 6-41: Allocate costs across locations

The next pie chart displays the information of allocated cost across 4 Indian cities. In general, costs are evenly allocated for 5 cities except for Mumbai.

First the city that takes up the most amount of cost is Chennai with 23.2%. Next up is Bangalore and Delhi that are quite similar in stats, which are at 21.9% and 20.5%. Second lowest is Kolkata with the percentage of 18.4% and the lowest is Mumbai with 16%.

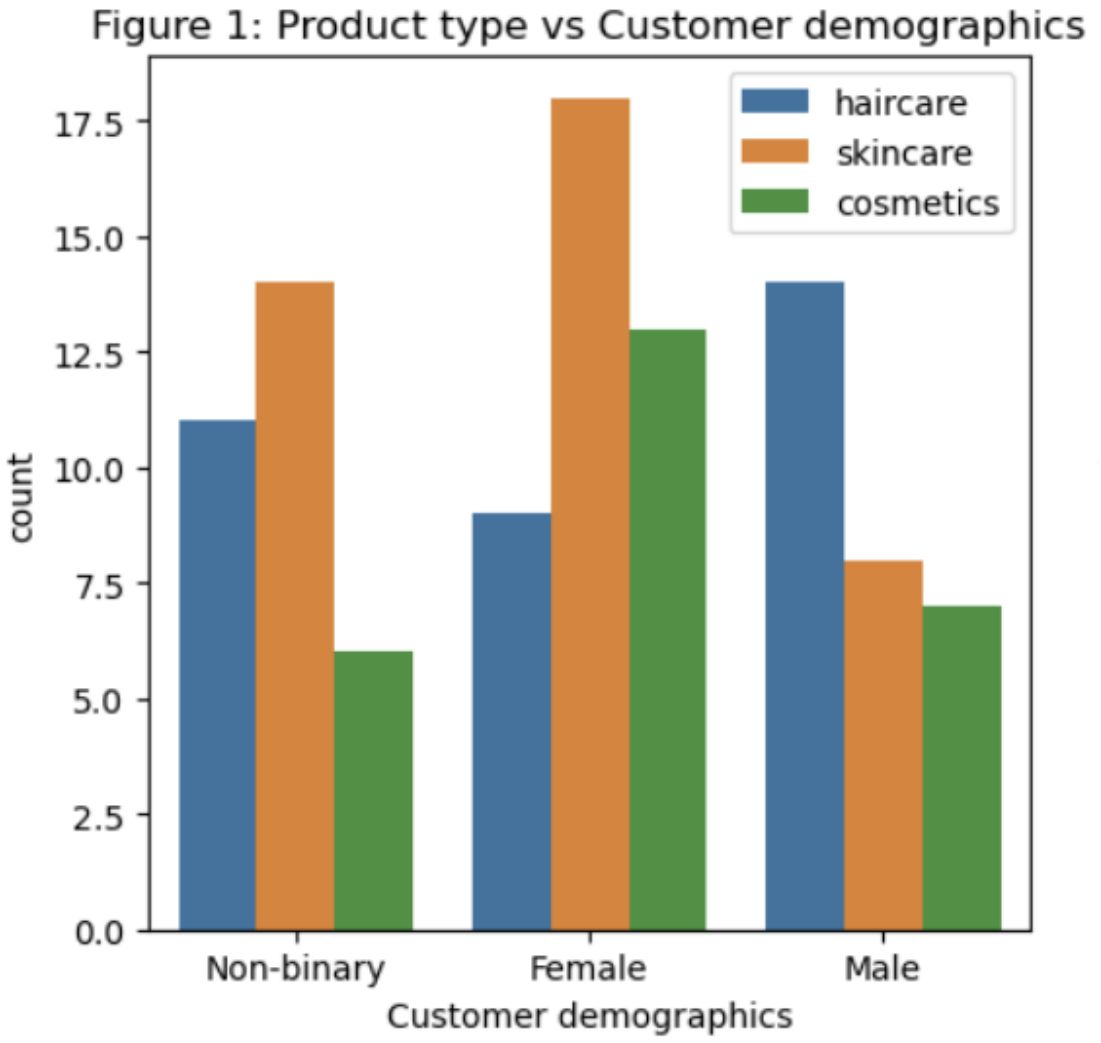

Figure 6-42: Compare product types for customer demographics

Next chart is the demographic for using the 3 product types, with 3 categories namely non-binary, female and male. With this chart, we can observe the differentiation between product categories and customer demographics.

First is the non-binary category, they use skin care the most with 14 counts. Lower by 3 is the number of hair care and cosmetics having less than 6. Females overall use the most of these products, highest being skin care at well over 17.5%, next in line is cosmetics at 12.5 and last one is hair 9. Last one is the male category, hair care dominates in men at 14 counts with skin care and cosmetics at around 7.5.

In the chart, we can make detailed observations such as:

- For the Non-binary customer demographic group: they tend to purchase Skincare products more than Haircare and Cosmetics.
- For the Female customer group: they have a higher tendency to purchase Skincare products compared to Cosmetics, and they are the least likely to purchase Haircare products.
- In contrast, for the Male customer group: they tend to choose Haircare products more than Skincare, and they are the least likely to purchase Cosmetics.

In summary, males have a higher tendency to purchase Haircare products compared to females, while females tend to choose Skincare products more than males. This reflects the consumption trends of each customer demographic group and shows the uneven differentiation among customer groups for each product category.

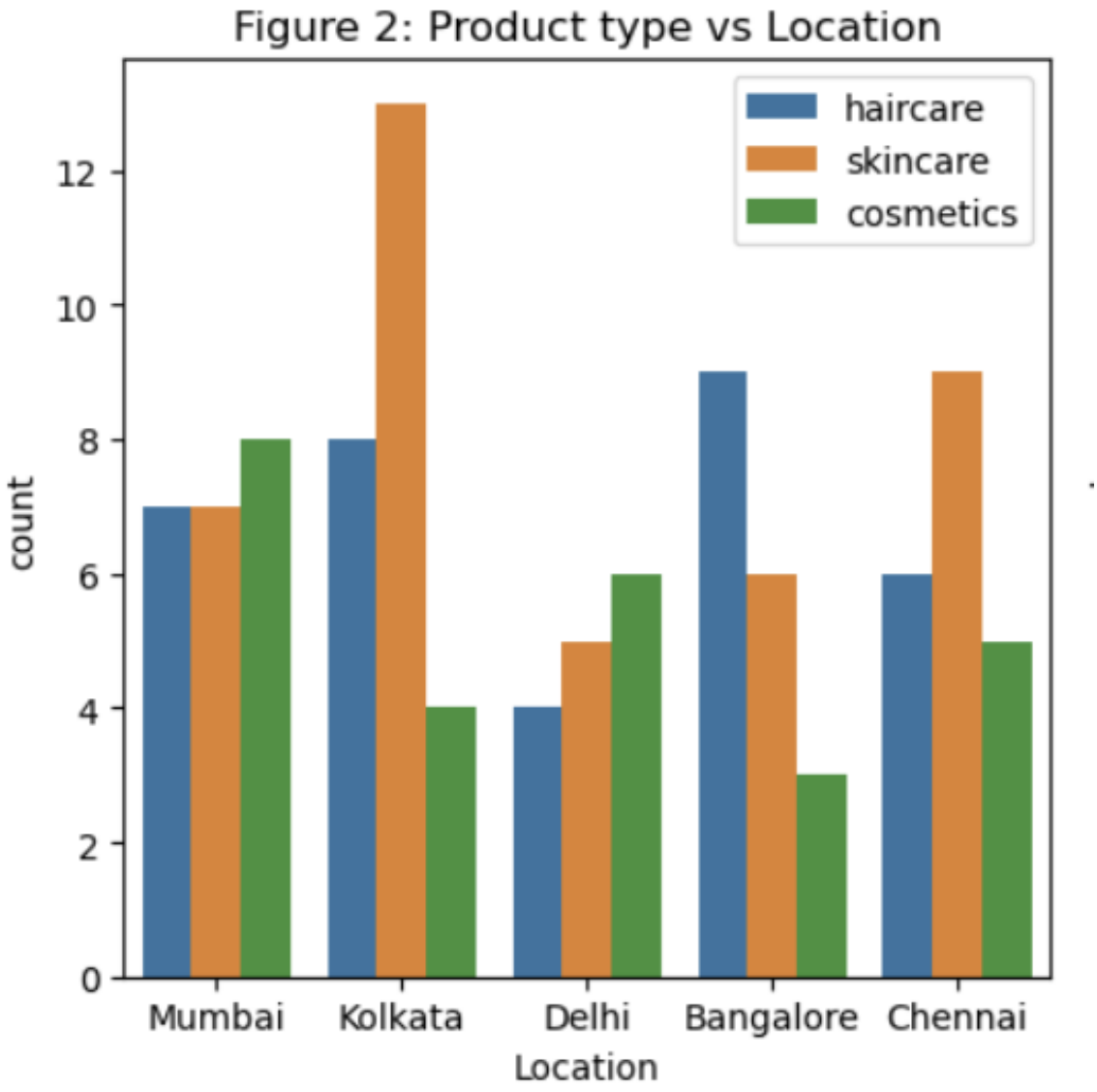

Figure 6-43: Compare product types for location

Next chart, we have the chart of products used in 5 different major indian cities. The stats are quite different in each city.

Mumbai, the city that has the most even distribution of products, hair care and skin care are the same at 7 and cosmetic is only 1 more than the rest. For Kolkata, skin care dominates the chart at dramatically more than 12 counts with hair care at 8 and cosmetics at 4. In Delhi, the figure differentiates quite evenly, ranging from 4 to 6 ranking hair care at the lowest and cosmetics at the highest. Same thing happens in Bangalore but the distance is way more significant, hair care is the highest at 9, and then skin care and cosmetics with each lower by 3.

Finally, Chennai with the most product sold being skin care at 9 with hair care and cosmetics at roughly the same number of about 6.

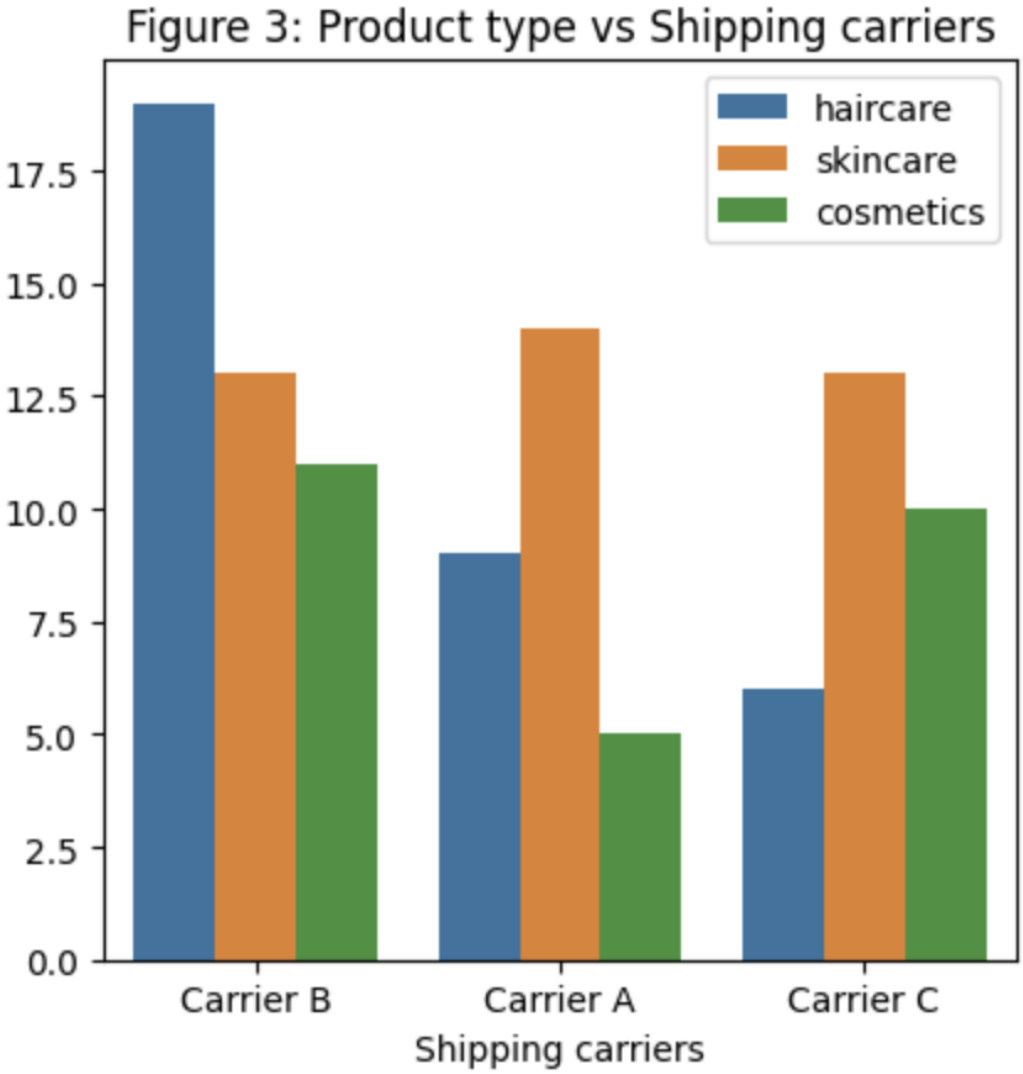

Figure 6-44: Compare product types for shipping carriers

The chart displays the information of 3 carriers namely A, B and C that carry the 3 product types. Overall, carrier B delivers the most products meanwhile A and c are quite similar in numbers.

First, carrier B transports the highest amount of hair care products, having more than 17.5 counts with skin care at 12.5 and cosmetics at 11. Skin care product type is the most in carrier A with 15.0 counts while hair care is 10 and cosmetics is about 1/3rd of skin care. Carrier C having similar stats for skin care except hair care is 5 and cosmetics is 10.

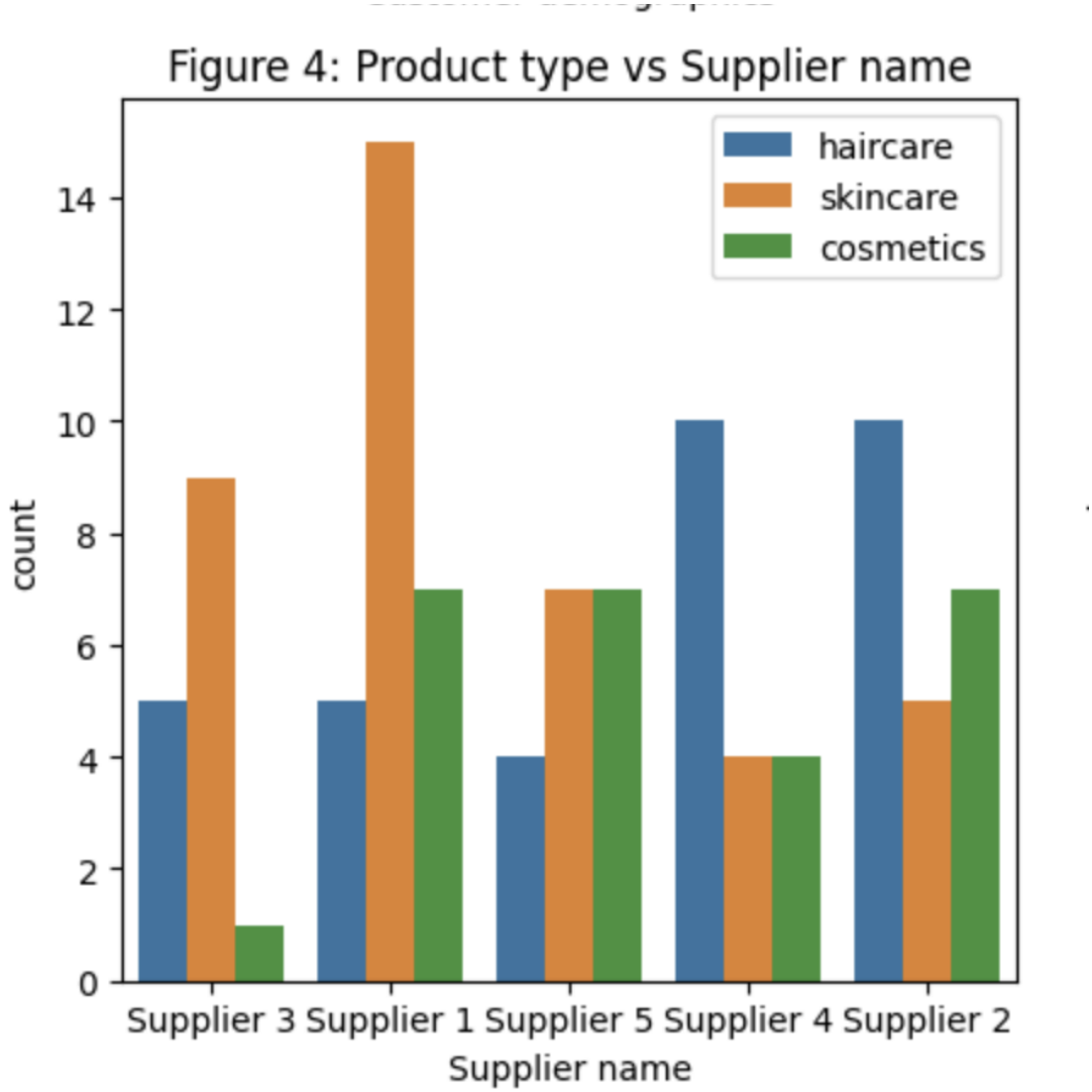

Figure 6-45: Compare product types for supplier name

Now we have a chart of suppliers providing the 3 types of products. Overall, most of the statistics are all quite similar except supplier 1 and 3.

First supplier 3, skin care having the highest count of 9 in this category with hair care is 5 and the smallest figure of 1 belongs to cosmetics for this supplier. Next supplier 1 is quite unique in terms that they supply the highest count of 15, more than any other suppliers. Hair care and cosmetics are quite standard having only 7 and 5 respectively.

For supplier 5 both skin care and cosmetics have the same 7 counts with hair care at 3 counts lower. Lastly, supplier 4 and 2 are both quite similar because hair care in both of them is 10. The only difference is in supplier 4 skin care and

cosmetics are both 4 but they are 5 and 7 respectively in supplier 2

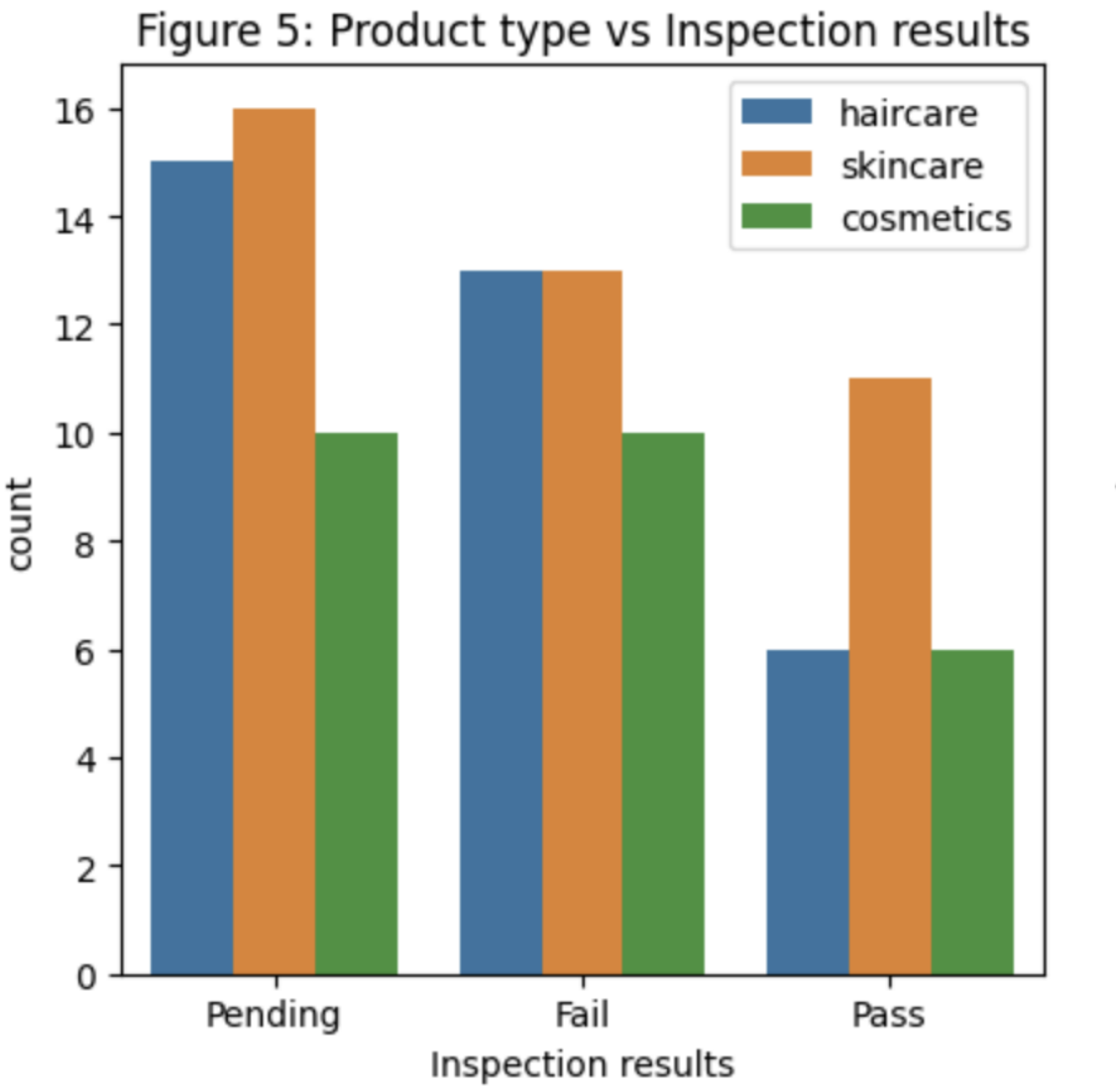

Figure 6-46: Compare product types for inspection results

The next chart presents the rate of 3 results of inspections tests for 3 product types. Overall, most of the products are in the inspection stage while not many products have passed.

First, hair care and skin care have the most significant figure in the pending stage, having 15 and 16 counts and cosmetics is only 10. A similar pattern can be spotted in the fail stage where hair care and skin care are the highest, both are 13, and cosmetics still have the same number of 10. Lastly, skin hair has the most number in pass with 13 counts, with hair care and cosmetics having only 6.

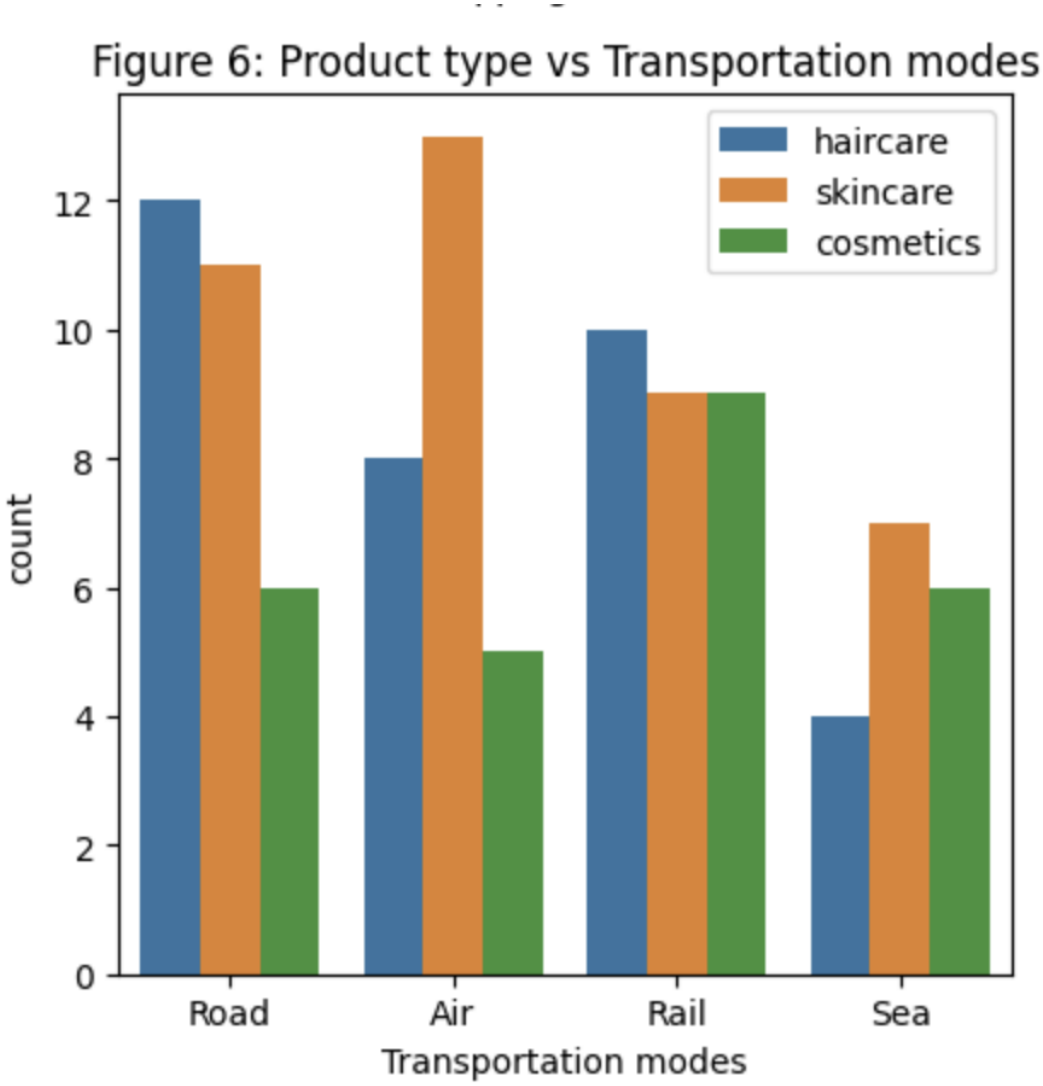

Figure 6-47: Compare product types for transportation modes

The next chart is about the amount of times that the 3 product types use road, air, rail and sea transportation modes. Overall, the figure for road, air and rail are generally more than sea.

First hair care and skin care products utilize road mode quite often, having 12 and 11 counts unlike cosmetics having only 6 counts. For air transportation, skin care contains the highest number of counts, which is more than 12 counts and the highest of all the figures. Hair care and cosmetics do not use air transport that much, the numbers are 8 and 5 respectively. For the rail method, the number of all 3 product types are very similar to each other. Skin care and cosmetics are at 9 counts and hair care is at 10 counts. Last one is sea, which is the least popular

mode to use. Only ranging from 4 to 6 with the highest being skin care and lowest is hair care.

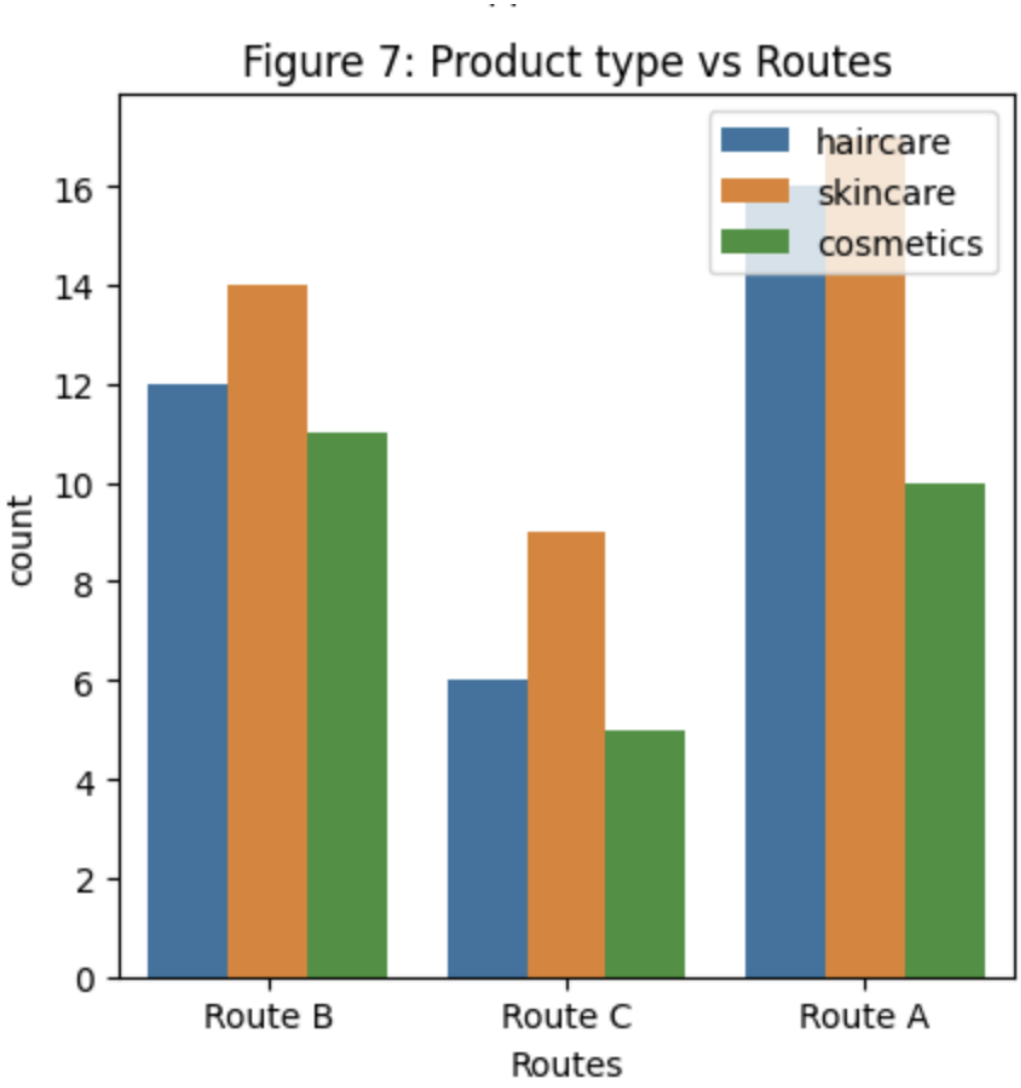

Figure 6-48: Compare product types for routes

The next chart analyzes the frequency of 3 product types using 3 routes, route A, B and C. Overall the most commonly used route is route A and the least popular is route C.

First, in route B, the highest count of 14 belongs to skincare with cosmetics and haircare at about 12 count. Route C overall has the least counts with the most being skincare at 10, both haircare and cosmetics are at around 5.

```
In [96]: avg_lead_time_by_sku = slc_df.groupby(["SKU","Product type"])['Lead time'].mean().reset_index()
         avg_lead_time_by_sku.head()
```

Out[96]:

	SKU	Product type	Lead time
0	SKU0	haircare	29.0
1	SKU1	skincare	23.0
2	SKU10	skincare	18.0
3	SKU11	skincare	28.0
4	SKU12	haircare	3.0

Figure 6-49: Average lead time by SKU (code)

```
In [97]: bar_lead_time_by_sku = px.bar(avg_lead_time_by_sku, x='SKU', y='Lead time', title='Average lead time',color='Product
         bar_lead_time_by_sku.show()
```

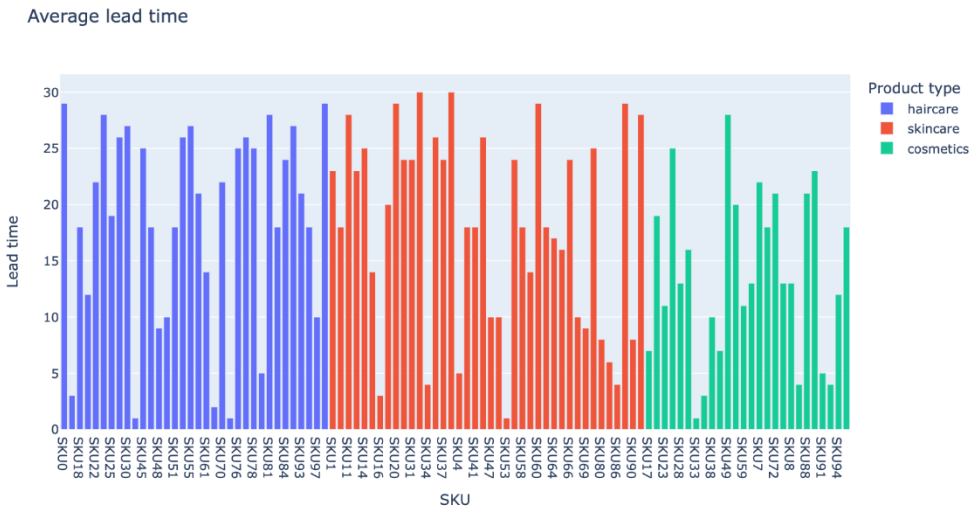

Figure 6-50: Average lead time by SKU (bar chart)

```
In [94]: avg_lead_time = slc_df.groupby("Product type")['Lead time'].mean().reset_index()
         avg_lead_time
```

Out[94]:

	Product type	Lead time
0	cosmetics	13.769231
1	haircare	18.647059
2	skincare	18.000000

Figure 6-51: Average lead time by product types (code)

```
In [95]: bar_lead_time = px.bar(avg_lead_time, x='Product type', y='Lead time', title='Average lead time')
         bar_lead_time.show()
```

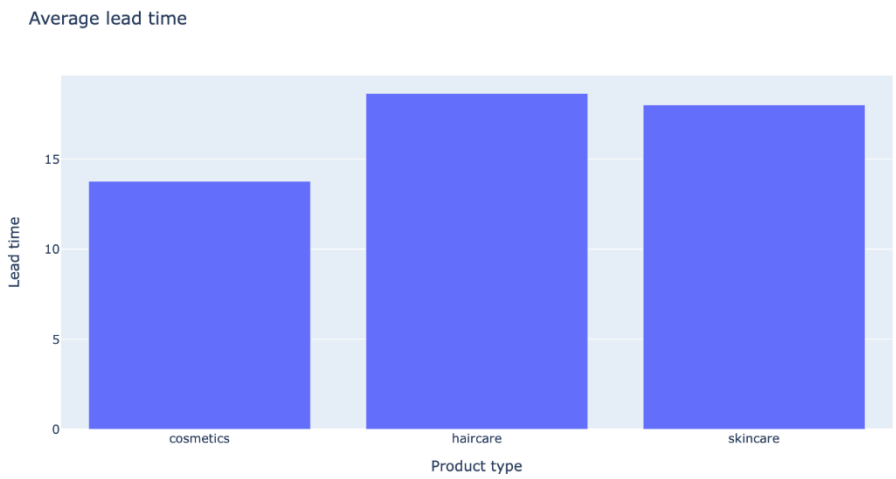

Figure 6-52: Average lead time by product types (bar chart)

The chart is a bar chart titled "Average Lead Time" and it shows the average lead time for different product types. The x-axis of the chart shows the product type, and the y-axis shows the lead time in days. The lead time for a product type is the amount of time it takes to get the product from the beginning of the production process to the finished good.

- Cosmetics has the shortest lead time, at 5 days.

- Haircare has a lead time of 10 days.

- Skincare has the longest lead time, at 15 days.

Overall, these are some of the key observations from the chart:

- Skin care products may have more complex ingredients or manufacturing processes than other product types.
- Skin care products may require more testing or quality control procedures than other products.
- There may be a longer lead time for sourcing the raw materials for skincare products.

Businesses can research ways to reduce production time for skin care products. This may involve streamlining the manufacturing process, improving communication between different departments, finding new suppliers with shorter

lead times, integrating production time of skin care products into their inventory planning and forecasting.

```
In [100]: supplier_lead_time_location = slc_df.groupby(["Supplier name","Location"])['Lead time'].mean().reset_index()
          supplier_lead_time_location
```

Out[100]:

	Supplier name	Location	Lead time
0	Supplier 1	Bangalore	15.600000
1	Supplier 1	Chennai	16.250000
2	Supplier 1	Delhi	6.500000
3	Supplier 1	Kolkata	19.375000
4	Supplier 1	Mumbai	12.166667
5	Supplier 2	Bangalore	19.800000
6	Supplier 2	Chennai	27.666667
7	Supplier 2	Delhi	18.833333
8	Supplier 2	Kolkata	25.333333
9	Supplier 2	Mumbai	10.600000
10	Supplier 3	Bangalore	10.333333
11	Supplier 3	Chennai	19.250000
12	Supplier 3	Delhi	23.000000
13	Supplier 3	Kolkata	22.666667
14	Supplier 3	Mumbai	23.333333
15	Supplier 4	Bangalore	16.000000
16	Supplier 4	Chennai	18.000000
17	Supplier 4	Delhi	3.000000
18	Supplier 4	Kolkata	15.666667
19	Supplier 4	Mumbai	17.500000
20	Supplier 5	Bangalore	22.333333
21	Supplier 5	Chennai	15.200000
22	Supplier 5	Delhi	28.000000
23	Supplier 5	Kolkata	18.600000
24	Supplier 5	Mumbai	15.250000

Figure 6-53: The lead time of suppliers by locations (code)

```
In [101]: bar_supplier_lead_time_location = px.bar(supplier_lead_time_location, x='Supplier name', y='Lead time', \
              color='Location', title='Lead time by supplier and location')
          bar_supplier_lead_time_location.update_traces(texttemplate='%{y:.2f}%', textposition='outside')
          bar_supplier_lead_time_location.show()
```

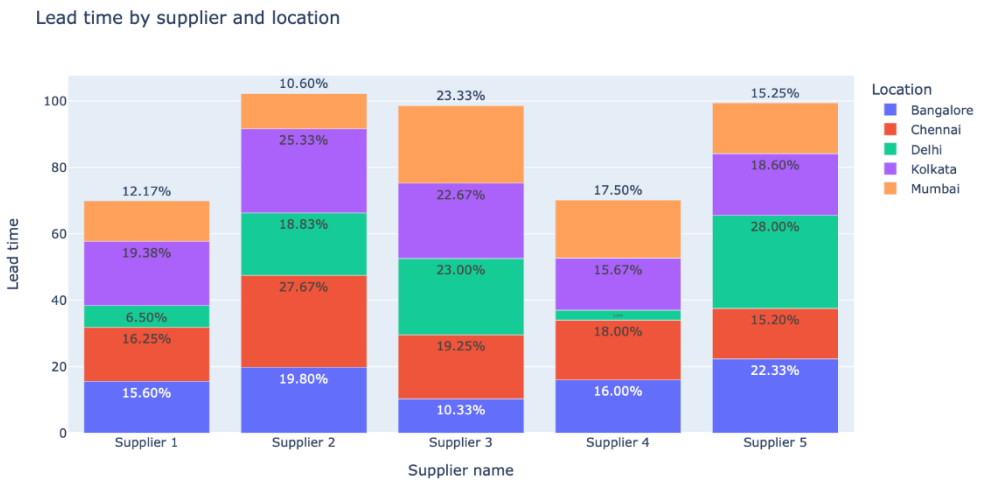

Figure 6-54: The lead time of suppliers by locations (bar chart)

This chart represents the response time of suppliers across different regions. Each column represents a region, and the height of the column indicates

the response time. The chart can be used to compare the performance and response time of suppliers in different regions.

Based on the chart, we can observe the differences in response time among the regions. Regions with taller columns indicate longer response times, while regions with shorter columns indicate faster response times. This may indicate variations in performance and responsiveness among suppliers in each region.

Through this analysis, we can identify the regions that need improvement in terms of response time to ensure customer satisfaction and optimize business operations. Improvement measures may include finding alternative suppliers or implementing other measures to reduce response time and enhance the performance of current suppliers.

```
In [81]: sales_portion_by_sku = slc_df.groupby(["SKU","Product type"])['Number of products sold'].sum().reset_index()
         sales_portion_by_sku.head()
```

Out[81]:

	SKU	Product type	Number of products sold
0	SKU0	haircare	802
1	SKU1	skincare	736
2	SKU10	skincare	996
3	SKU11	skincare	960
4	SKU12	haircare	336

Figure 6-55: Number of products sold by SKU (code)

```
In [82]: bar_sales_by_sku = px.bar(sales_portion_by_sku, x='SKU', y='Number of products sold', title='Sales by SKU',color='Pr
         bar_sales_by_sku.show()
```

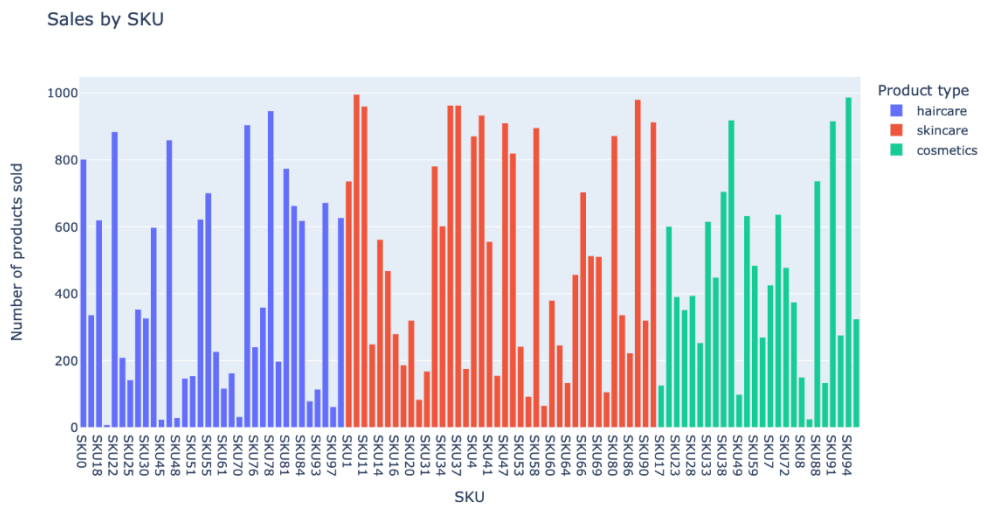

Figure 6-56: Number of products sold by SKU (bar chart)

```
In [102]: customer_data = slc_df[['SKU','Product type','Number of products sold','Revenue generated','Customer demographics']]
          customer_data.head()
```

Out[102]:

	SKU	Product type	Number of products sold	Revenue generated	Customer demographics
0	SKU0	haircare	802	8661.996792	Non-binary
1	SKU1	skincare	736	7460.900065	Female
2	SKU2	haircare	8	9577.749626	Female
3	SKU3	skincare	83	7766.836426	Non-binary
4	SKU4	skincare	871	2686.505152	Non-binary

Figure 6-57: Number of products, revenue generated sold by customer demographics (code)

```
In [103]: pie_customer_demographics = px.pie(customer_data, values='Number of products sold', names='Customer demographics', \
              title='Number of products sold by customer demographics', hole=0.5,color_discrete_sequence=px.colors.qualitative
          pie_customer_demographics.update_traces(textposition='inside', textinfo='percent+label')
          pie_customer_demographics.show()
```

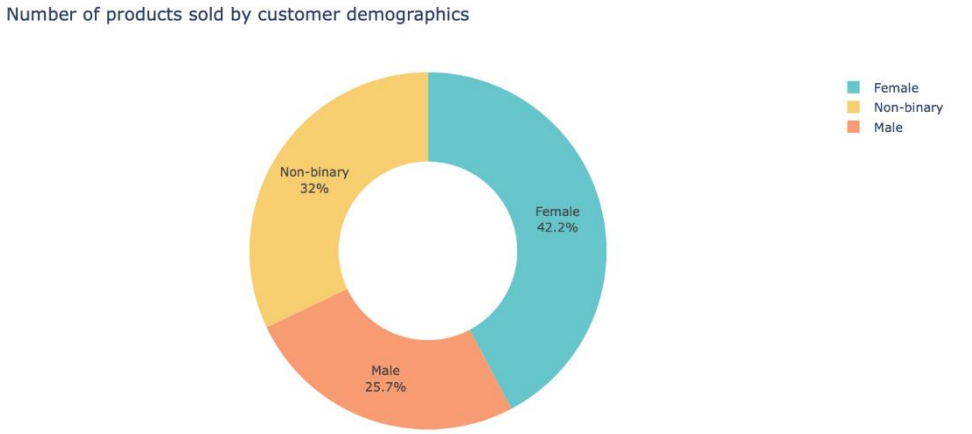

Figure 6-58: Number of products, revenue generated sold by customer demographics (pie chart)

The sales volume shows an uneven differentiation. Once again, the female customer group accounts for a weight of 42.2% of total revenue, which also reflects and reinforces the revenue chart by customer groups, indicating that sales volume is proportional to revenue for each group. This shows that the female customer group tends to be interested in beauty care products and allocates a larger financial resource for shopping compared to other customer groups.

```
In [98]: supplier_portion = slc_df.groupby("Supplier name")['Production volumes'].sum().reset_index()
         supplier_portion
```

Out[98]:

	Supplier name	Production volumes
0	Supplier 1	13545.0
1	Supplier 2	14105.0
2	Supplier 3	7997.0
3	Supplier 4	11756.0
4	Supplier 5	8751.0

Figure 6-59: Production volumes by suppliers (code)

```
In [99]: pie_supplier_portion = px.pie(supplier_portion, values='Production volumes', names='Supplier name', \
             title='Production volumes by supplier', hole=0.5,color_discrete_sequence=px.colors.qualitative.Pastel)
         pie_supplier_portion.update_traces(textposition='inside', textinfo='percent+label')
         pie_supplier_portion.show()
```

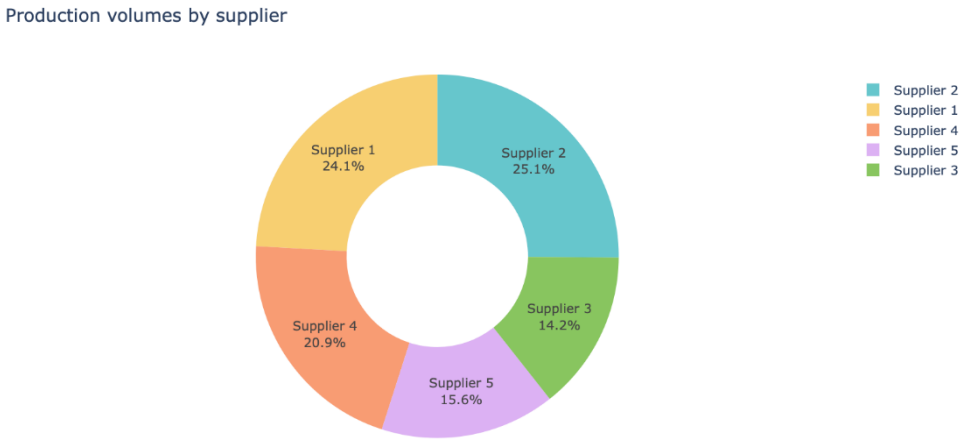

Figure 6-60: Production volumes by suppliers (pie chart)

The pie chart illustrates the allocation of production volume among different product types. Each segment of the chart represents a specific product type, and the size of each segment corresponds to the proportion of production volume assigned to that product type. The chart uses different colors to differentiate between product types.

We can determine the relative contribution of each product type to the total production volume. This information can be valuable in understanding the market demand for different products and making informed decisions regarding production planning, resource allocation, and supplier management.

```
In [77]: sales_revenue_portion = slc_df.groupby("Product type")['Revenue generated'].sum().reset_index()
         sales_revenue_portion
```

Out[77]:

	Product type	Revenue generated
0	cosmetics	161521.265999
1	haircare	174455.390605
2	skincare	241628.162133

Figure 6-61: Revenue generated by product types (code)

```
In [114]: pie_sales_revenue_portion = px.pie(sales_revenue_portion, values='Revenue generated', names='Product type',
                                             title='Sales revenue by product type',hole=0.5,\
              color_discrete_sequence=px.colors.qualitative.Pastel)
          pie_sales_revenue_portion.update_traces(textposition='inside', textinfo='percent+label')
          pie_sales_revenue_portion.show()
```

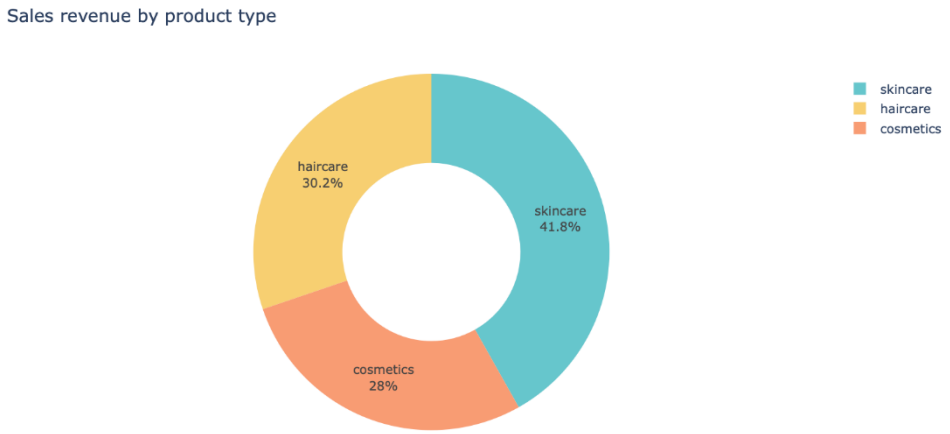

Figure 6-62: Sale revenue by product types (pie chart)

The chart is a pie chart titled "Sales by Product Type" and shows revenue generated by three product types: Skin Care, Hair Care, and Cosmetics

- Skincare generates the most revenue, accounting for 41.8% of the total sales revenue.

- Haircare generates 30.2% of the total sales revenue.

- Cosmetics generates 28% of the total sales revenue.

Overall, these are some of the key observations from the chart:

- Skin care products may be in higher demand than other product categories. This could be due to increased awareness among men and women of the importance of sun protection or a growing interest in organic or natural beauty products.

- The business focuses mainly on a wider range of skin care products or they may have a more favorable price compared to other product categories.

Businesses should focus on continuing to develop and market high-quality skin care products.

Businesses may consider expanding their skin care product line or investing in marketing and advertising to further increase brand and product awareness.

```
In [86]: revenue_by_carrier = slc_df.groupby("Shipping carriers")['Revenue generated'].sum().reset_index()
         revenue_by_carrier
```

Out[86]:

	Shipping carriers	Revenue generated
0	Carrier A	142629.994607
1	Carrier B	250094.646988
2	Carrier C	184880.177143

Figure 6-63: Revenue generated by shipping carriers (code)

```
In [87]: pie_chart_shipping_carrier = px.pie(revenue_by_carrier, values='Revenue generated', names='Shipping carriers', \
             title='Revenue Generated by Shipping Carriers', hole=0.5,color_discrete_sequence=px.colors.qualitative.Pastel)
         pie_chart_shipping_carrier.update_traces(textposition='inside', textinfo='percent+label')
         pie_chart_shipping_carrier.show()
```

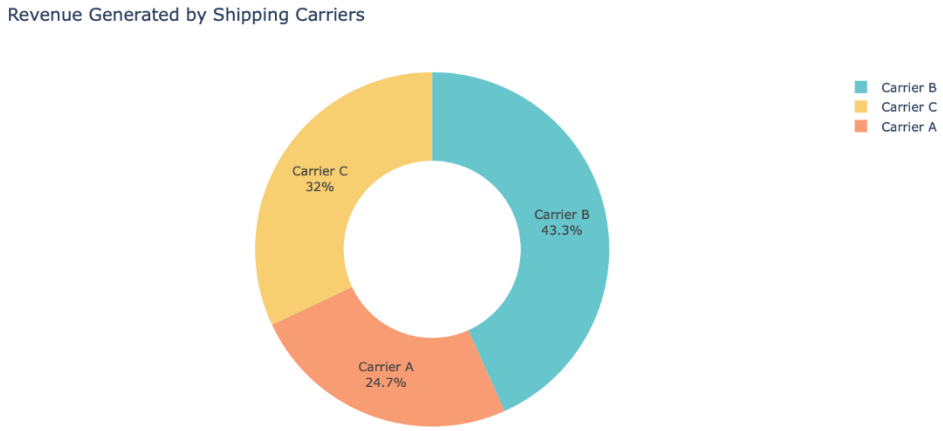

Figure 6-64: Revenue generated by shipping carriers (pie chart)

The chart is a pie chart titled "Revenue Generated by Shipping Carriers" and it shows the revenue generated by three different shipping carriers, Carrier A, Carrier B, and Carrier C.

- Carrier B generates the most revenue, accounting for 43.3% of the total revenue generated by shipping carriers.

- Carrier C generates 32% of the total revenue generated by shipping carriers.

- Carrier A generates the least revenue, accounting for 24.7% of the total revenue generated by shipping carriers.

Overall, these are some of the key observations from the chart:

- Carrier B may offer the most competitive prices or the most reliable service.
- Carrier B can handle larger shipment volumes than other carriers.

The business has a close relationship with Carrier B. It may be worth investigating ways to strengthen this relationship, such as negotiating lower prices or entering into strategic partnerships.

Businesses may consider diversifying their carrier base by using Carrier A and Carrier C more frequently, especially if they can offer consistent pricing or competitive service levels with Carrier B.

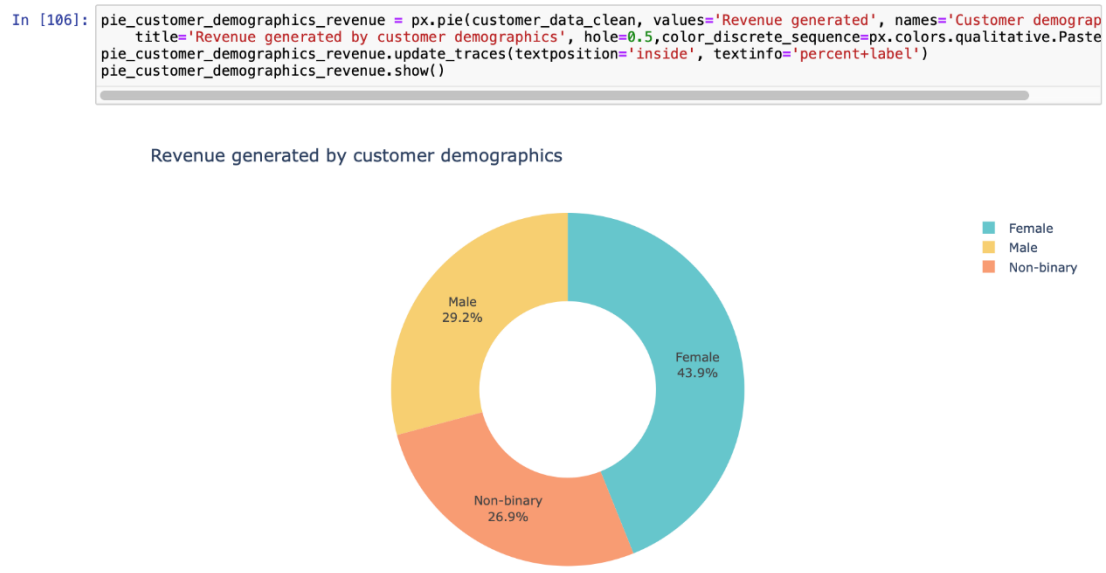

Figure 6-65: Revenue generated by customer demographics (pie chart)

Revenue breakdown by demographics:

- Males account for 29.2% of total revenue.
- Females account for 43.9% of total revenue.
- The Non-binary group accounts for the remaining 26.9%.

Overall, the female customer segment still holds the highest proportion in the revenue breakdown, indicating that female customers tend to make more purchases.

```
In [79]: sales_revenue_by_sku = slc_df.groupby(["SKU","Product type"])['Revenue generated'].sum().reset_index()
         sales_revenue_by_sku
```

Out[79]:

	SKU	Product type	Revenue generated
0	SKU0	haircare	8661.996792
1	SKU1	skincare	7460.900065
2	SKU10	skincare	2330.965802
3	SKU11	skincare	6099.944116
4	SKU12	haircare	2873.741446
...	...	...	...
95	SKU95	haircare	7386.363944
96	SKU96	cosmetics	7698.424766
97	SKU97	haircare	4370.916580
98	SKU98	skincare	8525.952560
99	SKU99	haircare	9185.185829

100 rows × 3 columns

Figure 6-66: Revenue generated by SKU (code)

```
In [80]: bar_sales_revenue_by_sku = px.bar(sales_revenue_by_sku, x='SKU', y='Revenue generated', \
             title='Sales revenue by SKU',color='Product type')
         bar_sales_revenue_by_sku.show()
```

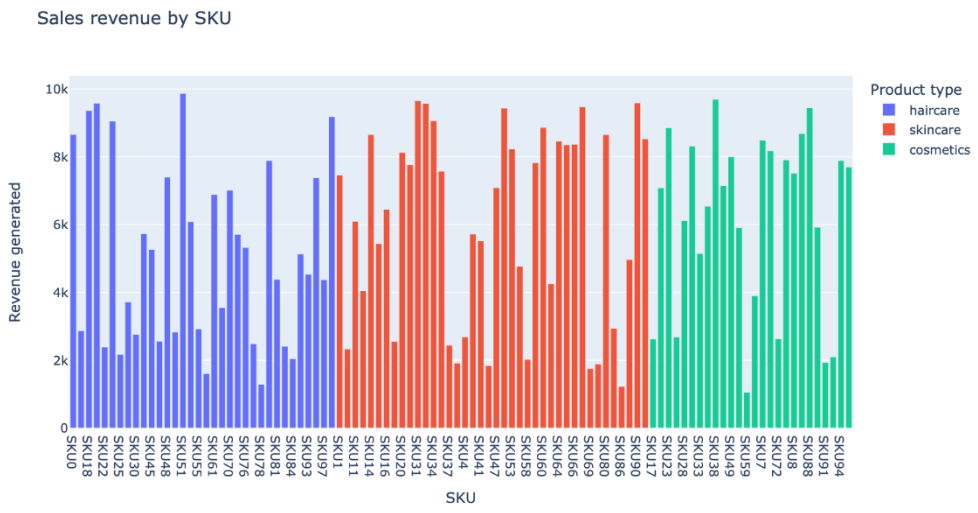

Figure 6-67: Sales revenue by SKU (bar chat)

```
In [90]: transportation_mode = slc_df.groupby("Transportation modes")['Costs'].sum().reset_index()
         transportation_mode
```

Out[90]:

	Transportation modes	Costs
0	Air	14604.527498
1	Rail	15168.931559
2	Road	16048.193639
3	Sea	7102.925520

Figure 6-68: Costs by transportation modes (code)

```
In [91]: pie_transportation_mode = px.pie(transportation_mode, values='Costs', names='Transportation modes', \
             title='Costs by Transportation Modes', hole=0.5,color_discrete_sequence=px.colors.qualitative.Pastel)
         pie_transportation_mode.update_traces(textposition='inside', textinfo='percent+label')
         pie_transportation_mode.show()
```

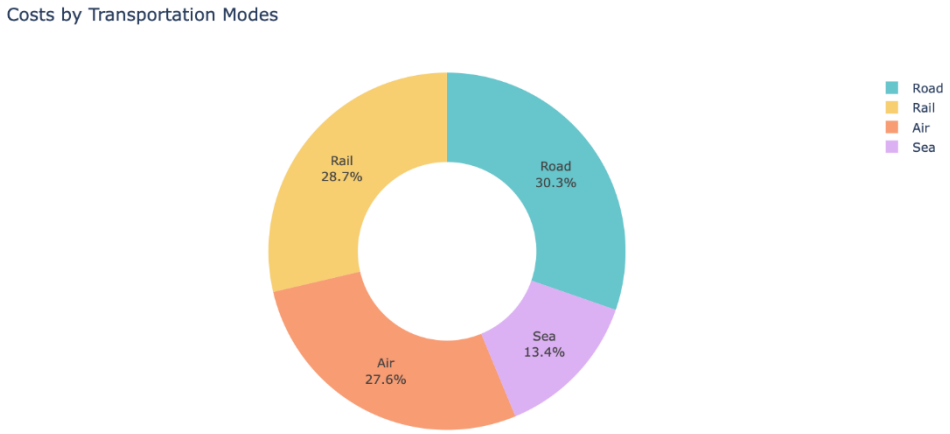

Figure 6-69: Costs by transportation modes (pie chart)

The pie chart shows the percentage of transportation costs for four different modes of transportation: Road, Rail, Sea, and Air.

- Rail is the second most expensive mode of transport, accounting for 28.7% of total transport costs.

- Road is the most expensive mode of transport, accounting for 30.3% of total transport costs.

- Sea is the least expensive mode of transport, accounting for 13.4% of total shipping costs.

- Air is the second least expensive mode of transport, accounting for 27.6% of total transport costs.

Road and Rail transport can be more expensive due to factors such as fuel costs, tolls, maintenance and labor costs for drivers or train operators.

Sea and Air freight can be less expensive due to economies of scale. Large container ships and airplanes can transport large quantities of goods at a lower cost per unit than smaller trucks or trains.

The pie chart shows only the cost ratio for each mode of transportation. It doesn't show other factors that may be important to decision making, such as speed, reliability and capacity.

The cost-effectiveness of each mode of transport will depend on the specific needs of the shipment. For example, if a shipment is time-sensitive, air shipping may be the best option, even if it is more expensive.

```
In [84]: avg_manufacturing_cost_by_sku = slc_df.groupby(["SKU","Product type"])['Manufacturing costs'].mean().reset_index()
         avg_manufacturing_cost_by_sku
```

Out[84]:

	SKU	Product type	Manufacturing costs
0	SKU0	haircare	46.279879
1	SKU1	skincare	33.616769
2	SKU10	skincare	96.527353
3	SKU11	skincare	27.592363
4	SKU12	haircare	32.321286
...	...	...	...
95	SKU95	haircare	58.890686
96	SKU96	cosmetics	17.803756
97	SKU97	haircare	65.765156
98	SKU98	skincare	5.604691
99	SKU99	haircare	38.072899

100 rows × 3 columns

Figure 6-70: Manufacturing costs by product types (code)

```
In [85]: bar_manufacturing_cost_by_sku = px.bar(avg_manufacturing_cost_by_sku, x='SKU', y='Manufacturing costs', \
             title='Average manufacturing costs',color='Product type')
         bar_manufacturing_cost_by_sku.show()
```

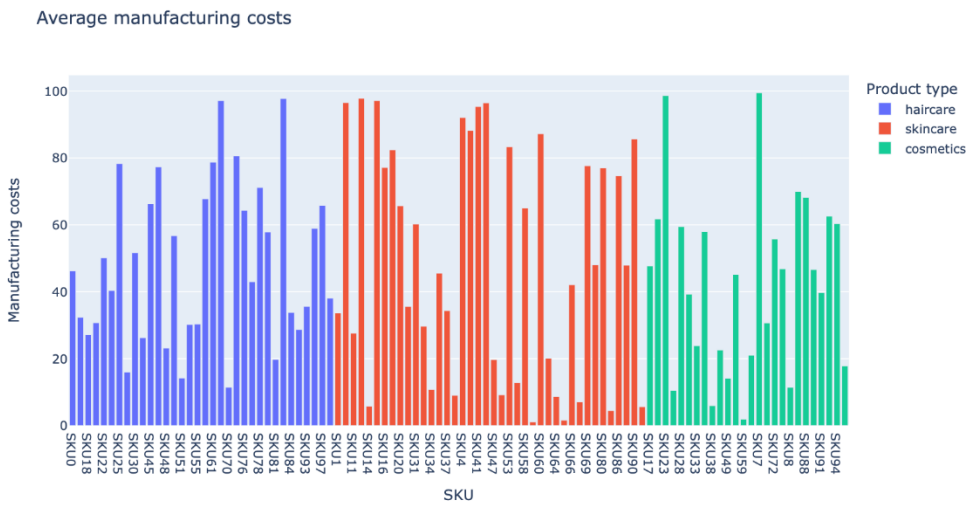

Figure 6-71: Manufacturing costs by product types (bar chart)

```
In [88]: shipping_carrier_cost = slc_df.groupby("Shipping carriers")['Costs'].sum().reset_index()
         shipping_carrier_cost
```

Out[88]:

	Shipping carriers	Costs
0	Carrier A	13927.071704
1	Carrier B	22725.444266
2	Carrier C	16272.062246

Figure 6-72: Costs by shipping carriers (code)

```
In [89]: pie_shipping_cost = px.pie(shipping_carrier_cost, values='Costs',  names='Shipping carriers', \
             title='Costs by Shipping Carriers', hole=0.5,color_discrete_sequence=px.colors.qualitative.Pastel)
         pie_shipping_cost.update_traces(textposition='inside', textinfo='percent+label')
         pie_shipping_cost.show()
```

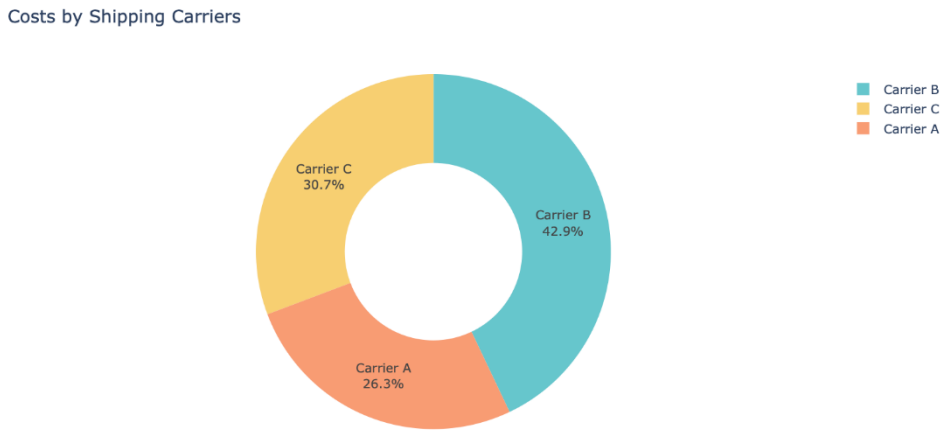

Figure 6-73: Costs by shipping carriers (bar chart)

The pie chart shows the distribution of shipping costs for three different carriers, Carrier A, Carrier B, and Carrier C.

- Carrier B is the carrier with the cheapest shipping costs, accounting for 26.3% of total shipping costs.

- Carrier A also has relatively cheap prices, accounting for 26.3% of total shipping costs.

- Carrier C is the most expensive carrier, accounting for 42.9% of total shipping costs.

Because the pie chart only shows the rates of each service provider, it is difficult to say exactly which shipping carrier is the most cost-effective. This will depend on the volume of product being moved, as the price per pound can vary between movers.

The business spends a significant portion of its shipping costs on Carrier C. It might be worthwhile to investigate if there are ways to reduce these costs, such as negotiating lower rates with Carrier C or finding alternative carriers for some shipments.

Carriers A and B are more cost-effective options. The business may want to consider using these carriers more frequently, especially for lighter weight items.

6.4. EDA multivariate analysis

```
In [109]: plt.figure(figsize=(12, 7))
          sns.heatmap(slc_df.drop(['Product type', 'SKU', 'Customer demographics', 'Shipping carriers',
                'Supplier name', 'Location', 'Inspection results',
                'Transportation modes', 'Routes'],axis=1).corr(), annot = True, vmin = -1, vmax = 1)
          plt.show()
```

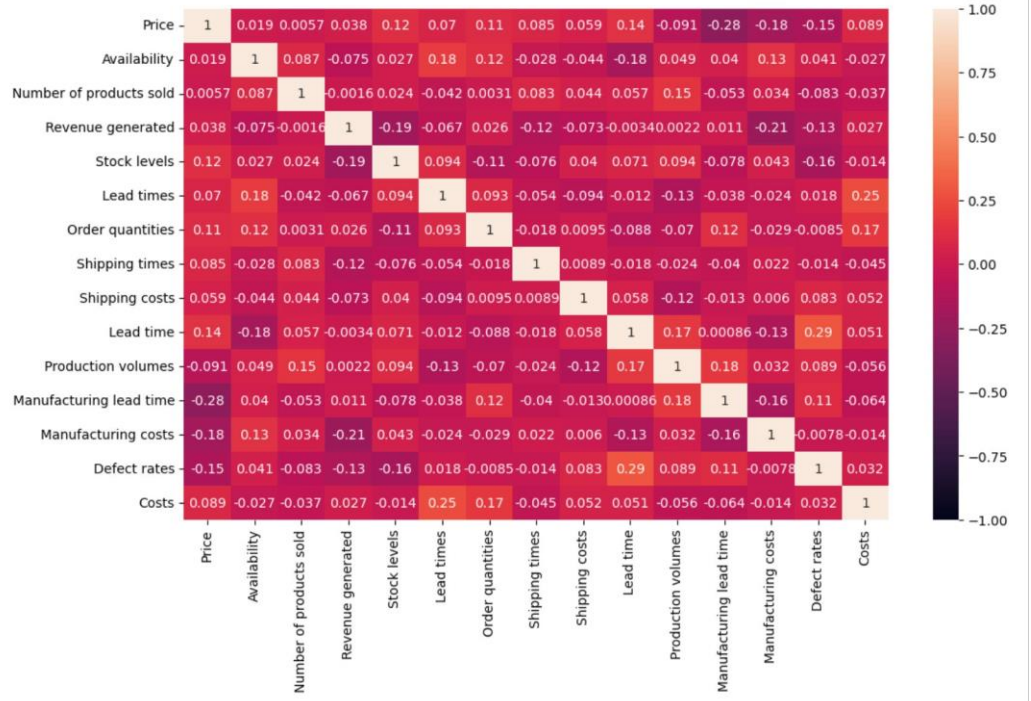

Figure 6-74: Heat map

In the heatmap, the color intensity of a cell indicates the correlation coefficient between the two variables represented by the row and column labels of that cell. Red indicates a negative correlation, and blue indicates a positive correlation. The intensity of the color corresponds to the strength of the correlation.

The heat map presents the correlation between several categories in terms of supply chain. Overall, there are not many connections between a lot of aspects and only a few are notable when paired up together.

As you can interpret from the map, when 2 categories are the same or have correlation to each other, the higher the number is. Correlation rate can range from -1.00 to 1.00 depending on how 2 types interact with each other. When 2 types are identical to each other, the heat would display as, the example for this would be Price, when paired up both make up the value of 1. For categories that are not the same, the statistic does not seem to surpass more than 0.2. Only a few are close like lead time and production volume, manufacturing lead time and production volume which are 0.17 and 0.18 respectively.

However, even though the figure does not go more than 0.2, it also does not go too low as seen from the chart. The only 2 that go down to 0.28 are Price and Manufacturing lead time. The rest of the pair with negative statistics only go down to -0.21.

Here are some conclusions that can be drawn from the heatmap:

- There is a positive correlation between the number of products sold and the revenue generated. This means that as the number of products sold increases, the revenue generated also increases. This is predictable because more sales lead to more revenue.

- There is a negative correlation between the number of products sold and the defect rate. This means that as the number of products sold increases, the error rate will decrease.

- There is a positive correlation between production time and production cost. This means that as production time increases, production costs also increase.

- There is a negative correlation between defect rate and production costs. This means that when error rates decrease, production costs also decrease.

- There is a positive correlation between lead time and production lead time. This means that as lead time increases, production lead time also increases. This shows that manufacturing lead time is a major component of total lead time.

- There is a negative correlation between the number of products sold and availability. This means that as the number of products sold increases, the amount of that product available decreases. This is also predictable since more sales will lead to lower inventory levels.

- There is a positive correlation between lead time and production lead time. This means that as lead time increases, production lead time also increases. This shows that manufacturing lead time is a major component of total lead time.

- There is a negative correlation between production time and defect rate. This means that as production time increases, error rates decrease. This may be because there is more time to identify and correct errors over longer production runs.

Overall, heat maps provide a useful way to visualize relationships between different variables in a data set. By analyzing heat maps, you can identify potential relationships between variables that can be studied further.

CHAPTER 7: RESULTS

Business Intelligence (BI) plays an important and indispensable role in modern supply chain management. By leveraging data analytics tools and techniques, businesses can gain valuable insights and information about their supply chain. Thereby, improving efficiency, reducing costs, and increasing revenue and increase customer satisfaction.

Detailed information how Business Intelligence supports supply chain management data analysis:

- *Data integration and visualization:* Tools in BI collect data from a variety of sources across the supply chain, including suppliers, manufacturers, distributors, and retailers. This comprehensive view gives businesses a comprehensive understanding of material flows, inventory levels and potential risks.

- *Demand forecasting:* BI supports analysis of sales data, consumer behavior and market trends to predict future demand fluctuations. This allows businesses to optimize inventory levels, plan production and allocate resources to meet customer demand without worrying about running out of stock or overstocking.

- *Risk management:* By analyzing previous data, BI helps predict potential disruptions in supply chain management. This may include supplier delays, shipping issues or fluctuations in raw material prices. Businesses can then develop proactive strategies to mitigate these risks and minimize their impact.

- *Performance measurement:* Data statistics and reporting dashboards of BI provide real-time insights into performance metrics such as delivery time, order fulfillment rate and shipping costs. This enables continuous monitoring and improvement of supply chain processes.

Benefits of analyzing supply chain data thanks to BI:

- *Increased efficiency:* Optimized processes lead to faster delivery times, avoid wastage of raw materials and reduce overall costs (production costs, stocking costs, inventory costs, purchase costs, shipping costs, etc).

- *Improved customer satisfaction:* By meeting needs effectively, businesses can deliver products on time and in full, building trust and enhancing customer loyalty.

- *Enhanced decision making:* Data-driven insights help businesses make informed decisions on inventory management, logistics management, procurement management, sourcing, management relationship with suppliers.

- *Greater flexibility:* BI helps businesses respond quickly to changes in market conditions, customer needs and unexpected risks in the supply chain.

In summary, supply chain management analysis in Business Intelligence provides a powerful approach to optimize business operations, gain competitive advantage and ensure the continuity of today's modern business operations. Good supply chain management also brings efficiency in logistics activities and goods to businesses and customers as quickly as possible, minimize costs and increase profits for businesses.

CHAPTER 8: SUGGESTION

There are some suggestions on some specific areas that can research further about supply chain management analysis in the Business Intelligence subject.

- *Predictive analytics for supply chain risk:* use advanced analytics, machine learning and AI in BI to predict and proactively address potential risks such as environmental factors, complexity in partner networks or changes in government regulations affecting the supply chain.

- *Real-time visibility in a multi-tier supply chain:* Exploring the challenges and solutions for achieving real-time visibility across a complex supply chain with multiple tiers of manufacturers, suppliers, distributors and retailers. The role of Business Intelligence is the integration of information systems across departments and the ability to collect, process and analyze data in real time to have a unified, more cost-effective view, creating a stronger supply chain.

- *The impact of business intelligence on sustainable supply chains:* Applying BI to promote sustainable activities in the supply chain. This drives more accurate business decisions, help organizations increase revenue and improve operational efficiency and gain a competitive advantage over competitors.

- *The role of business intelligence in building collaborative relationships with suppliers:* BI tools can facilitate information sharing and collaboration between businesses and suppliers and partners. This can involve general forecasts or real-time inventory visibility to improve overall supply chain efficiency.

- *Using business intelligence for dynamic pricing based on supply chain data:* Business Intelligence can be leveraged to implement dynamic pricing strategies based on real-time supply chain data. This may involve adjusting prices based on inventory levels, competitor prices, expected demand fluctuations helping the business drive sales, maximize profits, provide more insights and gain deeper insight into consumer behavior.

- *Leveraging AI and machine learning for predictive maintenance:* By applying AI and machine learning in BI to predict equipment failures across the supply chain, enables proactive maintenance and minimizes downtime.

- *Using business intelligence with Internet of Things (IoT) sensors:* Real-time data analytics from IoT sensors embedded in the infrastructure can be integrated with BI to gain deeper insights into performance of the supply chain.

In summary, supply chain management analysis in Business Intelligence can focus on predictive analytics for risk, real-time visibility in a multi-tier supply chain, promoting sustainable activities, building collaborative relationships with suppliers, dynamic pricing based on supply chain data, leveraging AI and machine learning for predictive maintenance, and integrating IoT sensors for deeper insights. By integrating information systems across departments, BI can provide a unified, cost-effective view, drive accurate business decisions, and improve operational efficiency. By leveraging AI and machine learning, organizations can drive sales, maximize profits, and gain deeper insights into consumer behavior.

REFERENCES

1) Google Colab là gì? Khám phá nền tảng sổ ghi chép tính toán trực tuyến: https://www.matbao.net/tin-tuc/google-colab-la-gi-kham-pha-nen-tang-so-ghi-chep-tinh-toan-truc-tuyen-134477.html
2) What is Kaggle, Why I Participate, What is the Impact?: https://www.kaggle.com/discussions/getting-started/44916

APPENDIX

[1] Faculty Development and Instructional Design Center. (n-d). *Data Analysis*. Faculty Development and Instructional Design Center. Access date 12/03/2024 from

https://ori.hhs.gov/education/products/n_illinois_u/datamanagement/datopic.html#:~:text=Data%20Analysis,and%20recap%2C%20and%20evaluate%20data.

[2] Gottschalk, L. A. (1995). *Content analysis of verbal behavior: New findings and clinical applications*. *Hillside, NJ: Lawrence Erlbaum Associates, Inc*

[3] Jeans, M. E. (1992). *Clinical significance of research: A growing concern. Canadian Journal of Nursing Research*, 24, 1-4.

[4] Lefort, S. (1993). *The statistical versus clinical significance debate*. Image, 25, 57-62.

[5] Kendall, P. C., & Grove, W. (1988). *Normative comparisons in therapy outcome*. Behavioral Assessment, 10, 147-158.

[6] Nowak, R. (1994). Problems in clinical trials go far beyond misconduct. Science. 264(5165): 1538-41.

[7] Resnik, D. (2000). *Statistics, ethics, and research: an agenda for educations and reform*. Accountability in Research. 8: 163-88

[8] Stephen Weston, Robert Bjornson. (April 2016). *Introduction to Anaconda*. Yale University.