**Analyzing supply chain data from**

**AdventureWorks2019**

Contents

[First: Introduction 2](#_Toc196515596)

[Second: Tables 3](#_Toc196515597)

[Third: KPIs 4](#_Toc196515598)

[Fifth:Python 7](#_Toc196515599)

[Sixth: Power bi 10](#_Toc196515600)

[Seventh: insight 14](#_Toc196515601)

[**1. HR analysis** 14](#_Toc196515602)

[**2. sales analysis** 15](#_Toc196515603)

[**3. profit analysis** 15](#_Toc196515604)

[**4. inventory analysis** 15](#_Toc196515605)

[**5. shipping analysis** 15](#_Toc196515606)

[**6. delivery analysis** 15](#_Toc196515607)

[**7. cost analysis** 16](#_Toc196515608)

**Team work**

* **Hazem Ahmed Abdel Hameed**
* **Doaa Saeed Issa**
* **Ahmed Salah Mohamed**
* **Soad Samir Munazie**
* **Amr Gamal Eldin**

First: Introduction

**Supply Chain Analysis Using AdventureWorks2019**

AdventureWorks2019 is a sample database from Microsoft that simulates a real-world supply chain for a fictional bicycle company. It includes data on products, sales, purchasing, inventory, and suppliers, making it an excellent dataset for supply chain analysis.

**Objective of the Analysis**

goal is to analyze key supply chain KPIs (Key Performance Indicators) using SQL and Python, then visualize the results using Power Query, Power BI, and Python. This will help identify inefficiencies, optimize inventory, and improve supplier performance.

**Key Supply Chain KPIs in AdventureWorks2019**

1. Inventory Turnover – How frequently products are sold and restocked.

2. Supplier Performance – Lead times and on-time deliveries.

3. Order Accuracy – Measuring discrepancies in purchase orders and deliveries.

4. Stock Availability – Identifying out-of-stock or overstocked items.

5. Delivery Performance – Analyzing on-time vs. delayed shipments.

6. Cost Analysis – Evaluating purchasing costs and profitability.

**Methodology for Analysis**

1. Extract Data Using SQL

Query tables, Join tables to connect purchasing, sales, and inventory data.

2. Perform Data Analysis Using Python

Use Pandas for data cleaning and transformation.

Apply Matplotlib/Seaborn for visual insights.

3. Visualize KPIs in Power BI

Build interactive dashboards for inventory, supplier, and sales analysis.

Create DAX measures for real-time calculations.

**Expected Insights**

Identify slow-moving or fast-selling products.

Detect supply chain bottlenecks.

Optimize purchasing and inventory strategies.

Improve supplier selection and performance tracking.

**Next Steps…**

Second: Tables

Identify where **the relevant data is stored in the database**. Here’s a breakdown of key **tables** related to supply chain KPIs:

Sales.SalesOrderDetail – Helps calculate inventory turnover.

Sales.SalesOrderHeader – Order shipment and delivery details.

Sales.SalesTerritory

Production.ProductInventory – Contains inventory levels per location.

Production.Product – Product information.

Production.ProductSubcategory

Production.ProductCategory

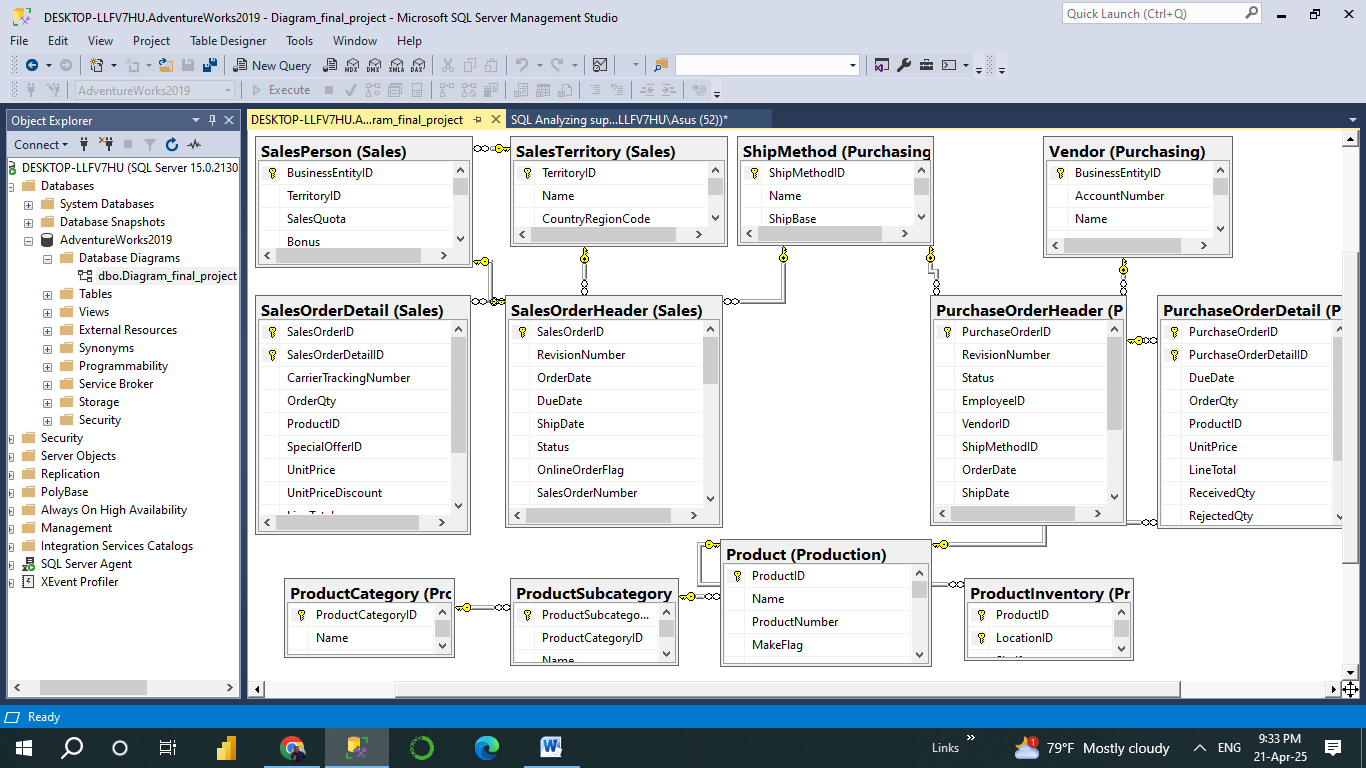
Purchasing.PurchaseOrderHeader – Stores purchase orders.

Purchasing.PurchaseOrderDetail – Contains order line details.

Purchasing.Vendor – Supplier/vendor details.

Purchasing.ShipMethod

**erd and maping database supply chain .**



**Next Steps…**

Third: KPIs

KPIs (Key Performance Indicators) should focus on efficiency, cost reduction, accuracy, and overall supply chain optimization. Here are some essential **KPIs categorized by function**:

**1. Inventory Management KPIs**

Inventory Turnover Ratio = Cost of Goods Sold (COGS) ÷ Average Inventory

Measures how quickly inventory is sold and replaced.

Days Sales of Inventory (DSI) = (Average Inventory ÷ COGS) × 365

Indicates how many days inventory is held before being sold.

Stock Accuracy = (System Inventory – Physical Inventory) ÷ System Inventory × 100

Measures the accuracy of inventory records.

Stock out Rate = (Stock out Events ÷ Total Orders) × 100

Tracks how often an item is out of stock when needed.

**2. Procurement & Supplier Performance KPIs**

Purchase Order Cycle Time = Time from Purchase Order Creation to Approval

Measures efficiency in procurement processing.

Supplier On-Time Delivery Rate = (On-Time Deliveries ÷ Total Deliveries) × 100

Evaluates supplier reliability.

Cost per Order = Total Procurement Costs ÷ Number of Orders

Monitors procurement efficiency and cost control.

Supplier Defect Rate = (Defective Items ÷ Total Items Received) × 100

Measures quality of supplier products.

**3. Logistics & Distribution KPIs**

On-Time Delivery Rate = (Orders Delivered on Time ÷ Total Orders) × 100

Tracks efficiency in meeting delivery schedules.

Freight Cost per Unit = Total Freight Cost ÷ Total Units Shipped

Assesses transportation cost efficiency.

Order Accuracy Rate = (Accurate Orders ÷ Total Orders) × 100

Warehouse Picking Accuracy = (Correctly Picked Orders ÷ Total Orders Picked) × 100

Measures how often the correct items are picked for shipment.

**4. Cost & Financial KPIs**

Total Supply Chain Cost = (Procurement Cost + Production Cost + Transportation Cost + Warehousing Cost)

Measures total spending on supply chain operations.

Indicates how long capital is tied up in supply chain processes.

Cost of Goods Sold (COGS) Percentage = (COGS ÷ Revenue) × 100

Shows how much revenue is spent on producing goods.

**Next Steps…**

Fourth: SQL

Detailed breakdown of SQL queries for extracting KPI data from AdventureWorks2019.

1. Inventory Management KPIs

Inventory Turnover Ratio

Formula: Inventory Turnover = Cost of Goods Sold (COGS) \ Average Inventory

SELECT

YEAR(soh.OrderDate) AS Year,

SUM(sod.LineTotal) AS COGS,

AVG(pi.Quantity) AS AvgInventory,

SUM(sod.LineTotal) / NULLIF(AVG(pi.Quantity), 0) AS InventoryTurnover

FROM Sales.SalesOrderHeader soh

JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

JOIN Production.ProductInventory pi ON sod.ProductID = pi.ProductID

GROUP BY YEAR(soh.OrderDate)

---

Stockout Rate

Formula:Stockout Rate= Stockouts\ Total Orders

SELECT

COUNT(\*) AS TotalOrders,

SUM(CASE WHEN pi.Quantity = 0 THEN 1 ELSE 0 END) AS StockoutCount,

(SUM(CASE WHEN pi.Quantity = 0 THEN 1 ELSE 0 END) \* 100.0) / COUNT(\*) AS StockoutRate

FROM Sales.SalesOrderDetail sod

JOIN Production.ProductInventory pi ON sod.ProductID = pi.ProductID

---

2. Procurement & Supplier KPIs

Supplier On-Time Delivery

Formula: On-Time Delivery =order date – ship date

SELECT

poh.PurchaseOrderID,

poh.OrderDate,

poh.ShipDate,

DATEDIFF(DAY, poh.OrderDate, poh.ShipDate) AS DaysBetweenOrderAndShip

FROM Purchasing.PurchaseOrderHeader poh

WHERE poh.ShipDate IS NOT NULL

---

Supplier Defect Rate

Formula: Supplier Defect Rate = Defective Items\ Total Items Received

SELECT

COUNT(\*) AS TotalItemsReceived,

SUM(CASE WHEN RejectedQty > 0 THEN RejectedQty ELSE 0 END) AS DefectiveItems,

(SUM(CASE WHEN RejectedQty > 0 THEN RejectedQty ELSE 0 END) \* 100.0) / NULLIF(SUM(ReceivedQty), 0) AS DefectRate

FROM Purchasing.PurchaseOrderDetail

---

3. Logistics & Distribution KPIs

On-Time Delivery Rate

Formula: On-Time Delivery = Orders Delivered On-Time\ Total Orders

SELECT

COUNT(\*) AS TotalOrders,

SUM(CASE WHEN soh.ShipDate <= soh.DueDate THEN 1 ELSE 0 END) AS OnTimeDeliveries,

(SUM(CASE WHEN soh.ShipDate <= soh.DueDate THEN 1 ELSE 0 END) \* 100.0) / COUNT(\*) AS OnTimeDeliveryRate

FROM Sales.SalesOrderHeader soh

---

Freight Cost per Order

Formula: Freight Cost per Order= Total Freight Cost\ Total Orders

SELECT

SUM(Freight) AS TotalFreightCost,

COUNT(SalesOrderID) AS TotalOrders,

SUM(Freight) / COUNT(SalesOrderID) AS FreightCostPerOrder

FROM Sales.SalesOrderHeader

---

4. Demand & Forecasting KPIs

SELECT

YEAR(OrderDate) AS Year,

MONTH(OrderDate) AS Month,

SUM(LineTotal) AS ActualSales

FROM Sales.SalesOrderHeader soh

JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

GROUP BY YEAR(OrderDate), MONTH(OrderDate)

ORDER BY Year, Month

Then, use Python to compare actual sales vs. forecasted values.

---

5. Financial KPIs

Total Supply Chain Cost

Formula: Total Supply Chain Cost = Procurement Cost + Production Cost + Transportation Cost + Warehousing Cost

SELECT

SUM(pod.LineTotal) AS ProcurementCost,

SUM(soh.Freight) AS TransportationCost

FROM Purchasing.PurchaseOrderDetail pod

JOIN Sales.SalesOrderHeader soh ON soh.SalesOrderID = soh.SalesOrderID

---

**Next Steps…**

Fifth:Python

Then, use Python to Perform Data Analysis Using Python.

1. Extract the data from SQL Server then **import** result as **CSV file**.

SELECT \*

FROM Sales.SalesOrderHeader soh JOIN Sales.SalesOrderDetail sod

ON soh.SalesOrderID = sod.SalesOrderID

JOIN Production.ProductInventory pi

ON sod.ProductID = pi.ProductID

join Production.Product pp

on pi.ProductID = pp.ProductID

join Production.ProductSubcategory psc

on pp.ProductSubcategoryID = psc.ProductSubcategoryID

join Production.ProductCategory ppc

on psc.ProductCategoryID = ppc.ProductCategoryID

join Sales.SalesTerritory sst

on soh.TerritoryID = sst.TerritoryID

---

SELECT \*

FROM Purchasing.PurchaseOrderHeader poh JOIN Purchasing.PurchaseOrderDetail pod

ON poh.PurchaseOrderID = pod.PurchaseOrderID

join Purchasing.ShipMethod psm

on poh.ShipMethodID = psm.ShipMethodID

join Purchasing.Vendor pv

on poh.VendorID = pv.BusinessEntityID

JOIN Production.ProductInventory pi

ON pod.ProductID = pi.ProductID

join Production.Product pp

on pi.ProductID = pp.ProductID

join Production.ProductSubcategory psc

on pp.ProductSubcategoryID = psc.ProductSubcategoryID

join Production.ProductCategory ppc

on psc.ProductCategoryID = ppc.ProductCategoryID

**2. Load Data into Python**

**Export** data from SQL Server to python using **CSV file**.

* import pandas as pd
* df = pd.read\_csv('C:/Users/Asus/Desktop/finalproject.csv', header=0)
* print(df.head())
* df.describe()
* df.info()
* df.isnull().sum()
* df.drop(columns = ['Unnamed: 29'], inplace =True)

---

* import matplotlib.pyplot as plt
* df['unitprice'].hist()
* unitprice\_median = df['unitprice'].median()
* print(unitprice\_median)

---

* subcategoryname\_unitprice\_avg = df.groupby('subcategoryname')['unitprice'].mean().sort\_values(ascending = False)
* print(subcategoryname\_unitprice\_avg)
* subcategoryname\_unitprice\_avg.plot(kind='barh', color='orange', figsize=(8, 5))

plt.title("unitprice\_avg by sub Category")

plt.xlabel("unitprice\_avg")

plt.ylabel("Product sub Category")

---

* subcategoryname\_orderqty\_sum = df.groupby('subcategoryname')['orderqty'].sum().sort\_values(ascending = False)
* print(subcategoryname\_orderqty\_sum)
* subcategoryname\_orderqty\_sum.plot(kind='barh', color='green', figsize=(8, 5))

plt.title("orderqty\_sum by sub Category")

plt.xlabel("orderqty\_sum")

plt.ylabel("Product sub Category")

* group\_count = df['group'].value\_counts()
* print(group\_count)
* group\_count.plot(kind = 'pie', autopct ='%1.1f%%')

---

* shipmethodid\_orderqty\_sum = df.groupby('shipmethodid')['orderqty'].sum().sort\_values(ascending = False)
* print(shipmethodid\_orderqty\_sum)
* shipmethodid\_orderqty\_sum.plot(kind = 'pie', autopct ='%1.1f%%')

plt.title("shipmethodid\_orderqty\_sum")

---

* categoryname\_orderqty\_sum = df.groupby('categoryname')['orderqty'].sum().sort\_values(ascending = False)
* print(categoryname\_orderqty\_sum)
* categoryname\_orderqty\_sum.plot(kind = 'pie', autopct ='%1.1f%%')
* plt.title("categoryname\_orderqty\_sum") df[['orderyear' , 'ordermonth' , 'orderday']] = df['orderdate'].str.split('-', n=2, expand = True)

---

* import seaborn as sns
* orderyear\_orderqty\_sum = df.groupby('orderyear')['orderqty'].sum().sort\_values(ascending = False)
* print(orderyear\_orderqty\_sum)
* plt.figure(figsize=(10, 5))

sns.lineplot(x='orderyear', y='orderqty', data=df, marker='o')

plt.title('Salesorderqty Over Time')

plt.xlabel('orderyearDate')

plt.ylabel('orderqtySales')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

---

* df['orderdate'] = pd.to\_datetime(df['orderdate'])

df['shipdate'] = pd.to\_datetime(df['shipdate'])

* df['DaysBetweenOrderAndShip'] = (df['shipdate'] - df['orderdate']).dt.days
* df['DaysBetweenOrderAndShip'].hist()
* On\_Time\_Delivery\_rate = df['DaysBetweenOrderAndShip'].median()
* print(On\_Time\_Delivery\_rate)

**Next Steps…**

Sixth: Power bi

Create advanced Power BI visualizations for your supply chain KPIs:

**1. Import Data into Power BI**

Method 1: Connect to SQL Server Directly

* Open Power BI Desktop.
* Click Home > Get Data > SQL Server.
* Enter Server Name and Database (AdventureWorks2019).
* Click Load to import data into Power BI.

Method 2: Import from Excel or CSV

* Run your SQL queries and export results to CSV/Excel.
* In Power BI, click Get Data > Excel/CSV, then load the file.

---

**2. Create Advanced Visualizations**

**1. HR dashboard**

**HR Analytics Dashboard:**

**Table:**

HumanResources.Department

HumanResources.EmployeeDepartmentHistory

HumanResources.EmployeePayHistory

**Calculated Columns (DAX):**

YearsOfService = DATEDIFF([HireDate], TODAY(), YEAR)

Age = DATEDIFF([BirthDate], TODAY(), YEAR)

**Measures (DAX) KPIS:**

AverageSalary = AVERAGE('HumanResources.EmployeePayHistory'[Rate])

TotalEmployees = DISTINCTCOUNT('HumanResources.Employee'[BusinessEntityID])

TotalSalaryPaid = SUM('HumanResources.EmployeePayHistory'[Rate])

AverageTenure = AVERAGE('HumanResources.Employee'[YearsOfService

**Add Filters:**

* Time slicer
* Department slicer

**---**

**2. Sales dashboard**

**Sales Analytics Dashboard:**

**Tables:**

Sales.SalesOrderHeader

Sales.SalesOrderDetail

Production.Product

Production.ProductSubcategory

Production.ProductCategory

Sales.SalesTerritory

**Measures (DAX) KPIS:**

Total Sales = SUM(Sales.SalesOrderDetail.LineTotal)

Total Orders = DISTINCTCOUNT(Sales.SalesOrderHeader.SalesOrderID)

Average Order Value = [Total Sales] / [Total Orders]

Total Sales qty = SUM( 'Sales SalesOrderDetail'[OrderQty])

average sales unit = [Total Sales val]/[Total Sales qty]

**Add Filters:**

* Time slicer
* territory slicer

---

**3. Profit dashboard**

**Profit Analytics Dashboard:**

**Tables:**

Sales.SalesOrderHeader

Sales.SalesOrderDetail

Production.Product

Production ProductCategory

Production ProductSubcategory

**Measures (DAX) KPIS:**

Total Sales = SUMX(SalesOrderDetail, SalesOrderDetail.OrderQty \* SalesOrderDetail.UnitPrice)

Total Cost = SUMX(SalesOrderDetail, SalesOrderDetail.OrderQty \* RELATED(Product.StandardCost))

Profit = [Total Sales] - [Total Cost]

Profit Margin % = DIVIDE([Profit], [Total Sales])

**Add Filters:**

* Time slicer
* category slicer

---

**4. Inventory** **dashboard**

**Inventory** **Analytics Dashboard:**

**Tables:**

Sales.SalesOrderHeader

Sales.SalesOrderDetail

Production.Product

Production.ProductInventory

**Measures (DAX) KPIS:**

Inventory Turnover

How often inventory is sold/replaced

Inventory Turnover = DIVIDE(SUM('SalesOrderDetail'[LineTotal]), AVERAGE('ProductInventory'[Quantity]))Average InventoryAvg stock on hand over time

Average Inventory = AVERAGE('ProductInventory'[Quantity])Out-of-Stock RateFrequency of stockoutsUse a calculated column: IF('ProductInventory'[Quantity] = 0, 1, 0)

**Add Filters:**

* Time slicer
* category slicer

---

**5. Shipping dashboard**

**Shipping Analytics Dashboard:**

Tables:

Production.Product

Purchasing.PurchaseOrderHeader

Purchasing.PurchaseOrderDetail

Purchasing Vendor

Purchasing ShipMethod

**Measures (DAX) KPIS:**

Supplier Lead TimeAvg time from PO to deliveryLead Time = AVERAGE (DATEDIFF('PurchaseOrderHeader'[OrderDate], 'PurchaseOrderHeader'[ShipDate], DAY))

PO Accuracy =

1 - DIVIDE(

  SUM(PurchaseOrderDetail[RejectedQty]),

  SUM(PurchaseOrderDetail[OrderQty]

Total purchase qty = SUM( 'Purchasing PurchaseOrderDetail'[OrderQty])

Total purchase freight = SUM('Purchasing PurchaseOrderHeader'[Freight])

average shipping freight unit = [Total purchase freight]/[Total purchase qty]

**Add Filters:**

* Time slicer
* shipping slicer

---

**6. Delivery dashboard**

**delivery Analytics Dashboard:**

Tables:

Production.Product

Sales.SalesOrderHeader

Sales.SalesOrderDetail

Purchasing ShipMethod

**Measures (DAX) KPIS:**

OnTimeDeliveryFlag =

DATEDIFF('Sales SalesOrderHeader'[OrderDate], 'Sales SalesOrderHeader'[ShipDate], DAY)

Total Sales qty = SUM( 'Sales SalesOrderDetail'[OrderQty])

Total Freight = sum('Sales SalesOrderHeader'[Freight])

Average freight unit = [total Freight]/[Total Sales qty]

**Add Filters:**

* Time slicer
* shipping slicer

---

**7. Cost dashboard**

**Tables:**

Sales.SalesOrderHeader

Sales.SalesOrderDetail

Production.Product

Production ProductCategory

**Measures (DAX) KPIS:**

Total Sales = SUMX(SalesOrderDetail, SalesOrderDetail.OrderQty \* SalesOrderDetail.UnitPrice)

Total Cost = SUMX(SalesOrderDetail, SalesOrderDetail.OrderQty \* RELATED(Product.StandardCost))

Average cost unit = [Total Cost]/[Total Sales qty]

Cost Margin % = DIVIDE([total cost], [Total Sales])

**Add Filters:**

* Time slicer
* Department slicer

---

**Next Steps…**

Seventh: insight

**1. HR analysis**

**Business Insights:**

The Company has shown consistent hiring trends in 2009 of year.

Although the percentage of male employees is approximately 70% and female employees is 30%, the average number of hours of vacation leave and sick leave is equal, which is an indicator of men commitment compared to women.

The average vacation and sick hours for the shipping and delivery department are higher than other departments.

Average hourly wage for the shipping and delivery department. Lowest average wage rate.

**Recommendations:**

* Pay attention to increasing the average wage rate of the shipping and delivery department to increase and improve the efficiency of shipping raw materials and delivering products to customers, and monitor the rates of vacations and sick leave.

**2. sales analysis**

**Business Insights:**

Sales volume indicators according to quantities are positive and increasing systematically. The products sold are divided into four categories: bikes, clothing, accessories, and components. The line graph according to time shows that the quantity of bikes sales until the second quarter of 2013 was the highest sales group.

It is also evident that sales quantities reaches its highest level during the second quarter of each year during the spring season and decreases to its lowest level during the first quarter of each year during the winter season.

The highest level of the company's sales is in the southwest region, and the lowest sales volume is in the Germany region.

**Recommendations:**

• Review marketing in areas with low sales and maximize sales and marketing in areas with high sales.

**3. Profit analysis**

**Business Insights:**

Although the number of units sold is very close between each product group during different periods, the value of sales in currency is very different. Here, the average unit selling price indicator showed us the reason, as the sales volume is an indicator of sales efficiency. Here, an important question appeared to us: Which of the product groups is more profitable?

According to the profit margin ratio to total sales value indicator, it is clear that the accessories group of products has the highest profit margin ratio, reaching 50% of the peak sales, despite it having the lowest average unit selling price compared to the rest of the product groups.

**Recommendations:**

* Focus on maximizing sales of the group of products, accessories, clothing, and after-sales services, as they achieve the highest and fastest profit margin to investment value.

**4. inventory analysis**

**Business Insights:**

According to the inventory turnover ratio, it is clear that the products of the bikes group have the highest rate of withdrawal from inventory and sale, and the products of the components group have the lowest rate of withdrawal from inventory. This explains the low inventory rates of the products of the components group.

Inventory turnover rates for all groups increase during the second quarter of each year during the spring seasons.

**Recommendations:**

* It is necessary to constantly ensure that the inventory balances of raw materials and finished products are within the safety stock so that there is no bottleneck in production or fulfillment of customer orders.

**5. shipping analysis**

**Business Insights:**

The average raw material shipping indicator shows that 9 days is the average shipping of raw materials from the time of ordering until receipt.

The accuracy of fulfilling purchase orders is approximately 97%, although there was a trend of a 70% decline during January 2013.

Although the largest purchase quantities are shipped via Cargo Transport, it has the highest average unit shipping price at $0.99, and shipping via XRQ - TRUCK GROUND with the lowest average unit shipping price at $0.15.

**Recommendations:**

* It is necessary to ensure that there is a stock reserve that meets production needs during this period, ensuring that production lines are not disrupted.
* Suppliers who adhere to shipping schedules should be reviewed.
* Study shipping via XRQ - TRUCK GROUND to directly impact cost.

**6. delivery analysis**

**Business Insights:**

The Average Product Delivery Index shows that 7 days is the average sales shipping time from the time of order to receipt.

The company uses two methods to ship and deliver finished products to customers: 78% via Cargo Transport and 22% via XRQ - TRUCK GROUND. Although the average unit shipping cost is almost the same in general, the unit shipping cost varies greatly according to each region separately.

**Recommendations:**

* It is necessary to ensure that there is a stock reserve that meets production needs during this.
* Therefore, it is recommended to ship using the least expensive method per unit in each region.

**7. Cost analysis**

**Business Insights:**

The average unit cost of the products at the beginning of the company's activity until the second quarter of 2012 was very high per unit, although the quantity of sales sold was very low. This is due to the distribution of fixed costs over a small number of products. Therefore, the company's maximum capacity must be produced and sold to reduce the cost to the lowest level.

**Recommendations:**

* Therefore, the company's maximum capacity must be produced and sold to reduce the cost to the lowest level.

Thank you…