# Supply Chain Analysis for Fashion and Beauty startup

The supply chain is the network of production and logistics involved in producing and delivering goods to customers. And Supply Chain Analysis means analyzing various components of a Supply Chain to understand how to improve the effectiveness of the Supply Chain to create more value for customers.

## Supply Chain Analysis: Dataset

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.

I found an ideal dataset for this task which includes data about the supply chain of a Fashion and Beauty startup. You can download the dataset from [**here**](https://statso.io/supply-chain-analysis-case-study/).

In the section below, I will take you through the task of Supply Chain Analysis using the Python programming language.

## Supply Chain Analysis using Python

Let’s get started with the task of Supply Chain Analysis by importing the necessary Python libraries and the [**dataset**](https://statso.io/supply-chain-analysis-case-study/):

import pandas as pd

import plotly.express as px

import plotly.io as pio

import plotly.graph\_objects as go

pio.templates.default = "plotly\_white"

data = pd.read\_csv("supply\_chain\_data.csv")

print(data.head())

**Product type SKU Price Availability Number of products sold \**

**0 haircare SKU0 69.808006 55 802**

**1 skincare SKU1 14.843523 95 736**

**2 haircare SKU2 11.319683 34 8**

**3 skincare SKU3 61.163343 68 83**

**4 skincare SKU4 4.805496 26 871**

**Revenue generated Customer demographics Stock levels Lead times \**

**0 8661.996792 Non-binary 58 7**

**1 7460.900065 Female 53 30**

**2 9577.749626 Unknown 1 10**

**3 7766.836426 Non-binary 23 13**

**4 2686.505152 Non-binary 5 3**

**Order quantities ... Location Lead time Production volumes \**

**0 96 ... Mumbai 29 215**

**1 37 ... Mumbai 23 517**

**2 88 ... Mumbai 12 971**

**3 59 ... Kolkata 24 937**

**4 56 ... Delhi 5 414**

**Manufacturing lead time Manufacturing costs Inspection results \**

**0 29 46.279879 Pending**

**1 30 33.616769 Pending**

**2 27 30.688019 Pending**

**3 18 35.624741 Fail**

**4 3 92.065161 Fail**

**Defect rates Transportation modes Routes Costs**

**0 0.226410 Road Route B 187.752075**

**1 4.854068 Road Route B 503.065579**

**2 4.580593 Air Route C 141.920282**

**3 4.746649 Rail Route A 254.776159**

**4 3.145580 Air Route A 923.440632**

**[5 rows x 24 columns]**

Let’s have a look at the descriptive statistics of the dataset:

print(data.describe())

**Price Availability Number of products sold Revenue generated \**

**count 100.000000 100.000000 100.000000 100.000000**

**mean 49.462461 48.400000 460.990000 5776.048187**

**std 31.168193 30.743317 303.780074 2732.841744**

**min 1.699976 1.000000 8.000000 1061.618523**

**25% 19.597823 22.750000 184.250000 2812.847151**

**50% 51.239831 43.500000 392.500000 6006.352023**

**75% 77.198228 75.000000 704.250000 8253.976921**

**max 99.171329 100.000000 996.000000 9866.465458**

**Stock levels Lead times Order quantities Shipping times \**

**count 100.000000 100.000000 100.000000 100.000000**

**mean 47.770000 15.960000 49.220000 5.750000**

**std 31.369372 8.785801 26.784429 2.724283**

**min 0.000000 1.000000 1.000000 1.000000**

**25% 16.750000 8.000000 26.000000 3.750000**

**50% 47.500000 17.000000 52.000000 6.000000**

**75% 73.000000 24.000000 71.250000 8.000000**

**max 100.000000 30.000000 96.000000 10.000000**

**Shipping costs Lead time Production volumes \**

**count 100.000000 100.000000 100.000000**

**mean 5.548149 17.080000 567.840000**

**std 2.651376 8.846251 263.046861**

**min 1.013487 1.000000 104.000000**

**25% 3.540248 10.000000 352.000000**

**50% 5.320534 18.000000 568.500000**

**75% 7.601695 25.000000 797.000000**

**max 9.929816 30.000000 985.000000**

**Manufacturing lead time Manufacturing costs Defect rates Costs**

**count 100.00000 100.000000 100.000000 100.000000**

**mean 14.77000 47.266693 2.277158 529.245782**

**std 8.91243 28.982841 1.461366 258.301696**

**min 1.00000 1.085069 0.018608 103.916248**

**25% 7.00000 22.983299 1.009650 318.778455**

**50% 14.00000 45.905622 2.141863 520.430444**

**75% 23.00000 68.621026 3.563995 763.078231**

**max 30.00000 99.466109 4.939255 997.413450**

Now let’s get started with analyzing the Supply Chain by looking at the relationship between the price of the products and the revenue generated by them:

fig = px.scatter(data, x='Price',

y='Revenue generated',

color='Product type',

hover\_data=['Number of products sold'],

trendline="ols")

fig.show()

Thus, the company derives more revenue from skincare products, and the higher the price of skincare products, the more revenue they generate. Now let’s have a look at the sales by product type:

sales\_data = data.groupby('Product type')['Number of products sold'].sum().reset\_index()

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=0.5,

color\_discrete\_sequence=px.colors.qualitative.Pastel)

pie\_chart.update\_traces(textposition='inside', textinfo='percent+label')

pie\_chart.show()

So 45% of the business comes from skincare products, 29.5% from haircare, and 25.5% from cosmetics. Now let’s have a look at the total revenue generated from shipping carriers:

total\_revenue = data.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()

So the company is using three carriers for transportation, and Carrier B helps the company in generating more revenue. Now let’s have a look at the Average lead time and Average Manufacturing Costs for all products of the company:

avg\_lead\_time = data.groupby('Product type')['Lead time'].mean().reset\_index()

avg\_manufacturing\_costs = data.groupby('Product type')['Manufacturing costs'].mean().reset\_index()

result = pd.merge(avg\_lead\_time, avg\_manufacturing\_costs, on='Product type')

result.rename(columns={'Lead time': 'Average Lead Time', 'Manufacturing costs': 'Average Manufacturing Costs'}, inplace=True)

print(result)

**Product type Average Lead Time Average Manufacturing Costs**

**0 cosmetics 13.538462 43.052740**

**1 haircare 18.705882 48.457993**

**2 skincare 18.000000 48.993157**

#### Analyzing SKUs

There’s a column in the dataset as SKUs. You must have heard it for the very first time. So, SKU stands for Stock Keeping Units. They’re like special codes that help companies keep track of all the different things they have for sale. Imagine you have a large toy store with lots of toys. Each toy is different and has its name and price, but when you want to know how many you have left, you need a way to identify them. So you give each toy a unique code, like a secret number only the store knows. This secret number is called SKU.

I hope you have now understood what’s SKU. Now let’s 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()

There’s another column in the dataset as Stock levels. Stock levels refer to the number of products a store or business has in its inventory. Now let’s have a look at the stock levels of each SKU:

stock\_chart = px.line(data, x='SKU',

y='Stock levels',

title='Stock Levels by SKU')

stock\_chart.show()

Now let’s have a look at the 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()

#### Cost Analysis

Now let’s analyze the shipping cost of Carriers:

shipping\_cost\_chart = px.bar(data, x='Shipping carriers',

y='Shipping costs',

title='Shipping Costs by Carrier')

shipping\_cost\_chart.show()

In one of the above visualizations, we discovered that Carrier B helps the company in more revenue. It is also the most costly Carrier among the three. Now let’s have a look at the cost distribution by transportation mode:

transportation\_chart = px.pie(data,

values='Costs',

names='Transportation modes',

title='Cost Distribution by Transportation Mode',

hole=0.5,

color\_discrete\_sequence=px.colors.qualitative.Pastel)

transportation\_chart.show()

So the company spends more on Road and Rail modes of transportation for the transportation of Goods.

#### Analyzing Defect Rate

The defect rate in the supply chain refers to the percentage of products that have something wrong or are found broken after shipping. Let’s have a look at the average defect rate of all product types:

defect\_rates\_by\_product = data.groupby('Product type')['Defect rates'].mean().reset\_index()

fig = px.bar(defect\_rates\_by\_product, x='Product type', y='Defect rates',

title='Average Defect Rates by Product Type')

fig.show()

So the defect rate of haircare products is higher. Now let’s have a look at the defect rates by mode of transportation:

pivot\_table = pd.pivot\_table(data, values='Defect rates',

index=['Transportation modes'],

aggfunc='mean')

transportation\_chart = px.pie(values=pivot\_table["Defect rates"],

names=pivot\_table.index,

title='Defect Rates by Transportation Mode',

hole=0.5,

color\_discrete\_sequence=px.colors.qualitative.Pastel)

transportation\_chart.show()

Road transportation results in a higher defect rate, and Air transportation has the lowest defect rate.

So this is how you can analyze a company’s supply chain using the Python programming language.

### Summary

Supply Chain Analysis means analyzing various components of a Supply Chain to understand how to improve the effectiveness of the Supply Chain to create more value for customers.