Double-click (or enter) to edit

Supply Chain Dynamics with Python

Introduction: The **Supply Chain** orchestrates the journey of goods, from production to delivery, shaping customer value. In this article, we explore **Supply Chain Analysis** using Python.

Dataset: Our dataset, from a **Fashion and Beauty startup**, encapsulates stages like **sourcing**, **manufacturing**, **transportation**, **inventory**, **sales**, and **customer demographics**.

Key Steps:

1. **Data Collection:** Gather comprehensive data spanning the supply chain stages.
2. **Data Preparation:** Cleanse and refine the dataset for analysis.
3. **Exploratory Data Analysis (EDA):** Unveil structural intricacies, distributions, and trends.
4. **Supply Chain Analysis:** Analyze sourcing, manufacturing, transportation, inventory, sales, and customer data.
5. **Visualization:** Craft insightful visualizations for clear communication.
6. **Insights and Recommendations:** Derive actionable insights to enhance supply chain efficiency.
7. **Implementation and Monitoring:** Execute enhancements and monitor performance for ongoing optimization.

Leverage Python and libraries like **Pandas**, **Matplotlib**, and **Seaborn** for a transformative journey through Supply Chain Analysis.

```
# Mounting to you own Google Colab drive
from google.colab import drive
try:
  drive.mount('/gdrive')
except:
  drive.mount('/content/gdrive', force_remount=True)

%cd '/gdrive/MyDrive/projects'

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('/gdrive/My Drive/projects/supply_chain_data.csv')
print(data.head())
```

```
Drive already mounted at /gdrive; to attempt to forcibly remount, call drive.mount("/gdrive", force_remount=Tru
/gdrive/MyDrive/projects
  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]
```

```
print(data.describe())
```

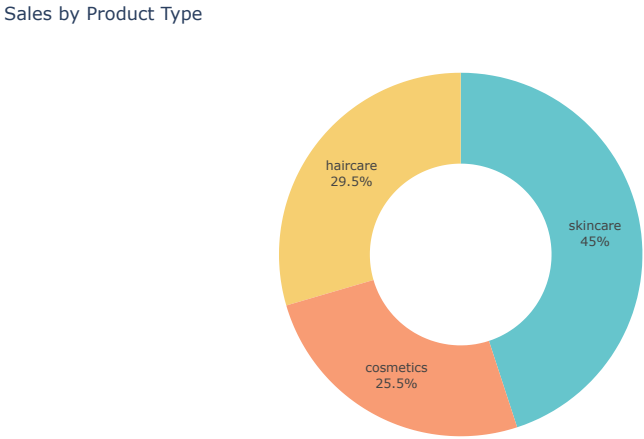
```
              Price  Availability  Number of products sold  Revenue generated  \
count  100.000000    100.000000               100.000000         100.000000
mean    49.462461     48.400000               460.990000        5776.048187
std     31.168193     30.743317               303.780074        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
```

Lets's analyze the relationship between the price of products and the revenue generated.

```
fig = px.scatter(data, x='Price',
                 y='Revenue generated',
                 color='Product type',
                 hover_data=['Number of products sold'],
                 trendline="ols")
fig.show()
```

Skincare products stand out as the primary revenue generators for the company, with sales positively correlated to **higher product prices**. Now, let's delve into the distribution of **sales across different product types**.

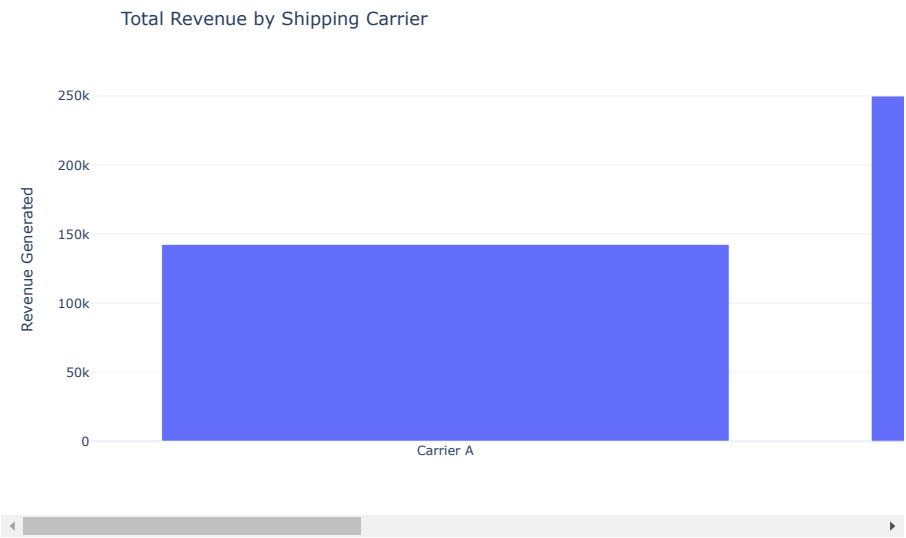
```
sales_data = data.groupby('Product type')['Number of products sold'].sum().reset_index()

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()
```



A significant portion of the business, **45%**, stems from **skincare products**, while **haircare** and **cosmetics** contribute **29.5%** and **25.5%**, respectively. Now, let's examine the **total revenue** attributed to **shipping carriers**.

The company's revenue distribution underscores the dominance of skincare products, accounting for **45%** of the total revenue. Meanwhile, **haircare** and **cosmetics** contribute **29.5%** and **25.5%**, respectively. With this insight into product-based revenue proportions, let's turn our attention to understanding the impact of **shipping carriers** on overall revenue generation.

```
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()
```



The company employs three carriers for transportation, with Carrier B notably contributing to **higher revenue generation**. Now, let's shift focus to examine the **average lead time** and **average manufacturing costs** across all products in the company's portfolio.

```
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'}, inp
print(result)
```

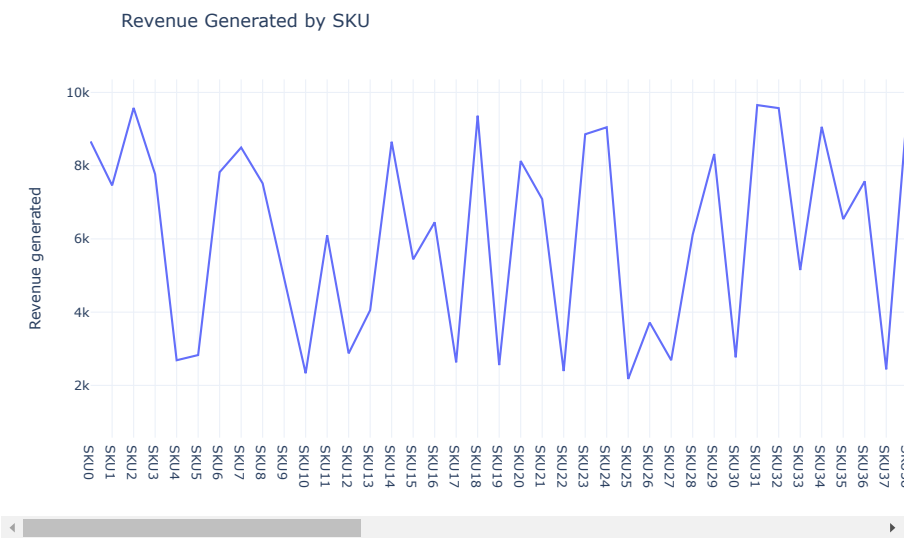
```
  Product type  Average Lead Time  Average Manufacturing Costs
0    cosmetics          13.538462                    43.052740
1     haircare          18.705882                    48.457993
2     skincare          18.000000                    48.993157
```

SKUs

In the dataset, there's a column labeled **SKUs**—a term you might encounter for the first time. **SKU**, an acronym for **Stock Keeping Units**, serves as a vital tool for companies to meticulously track their diverse range of products. Picture a sprawling toy store teeming with myriad toys. Each toy, while distinct in name and price, necessitates a unique identifier to monitor its inventory accurately. Thus, every toy is assigned a **unique code**, akin to a confidential number known exclusively to the store —a code referred to as **SKU**.

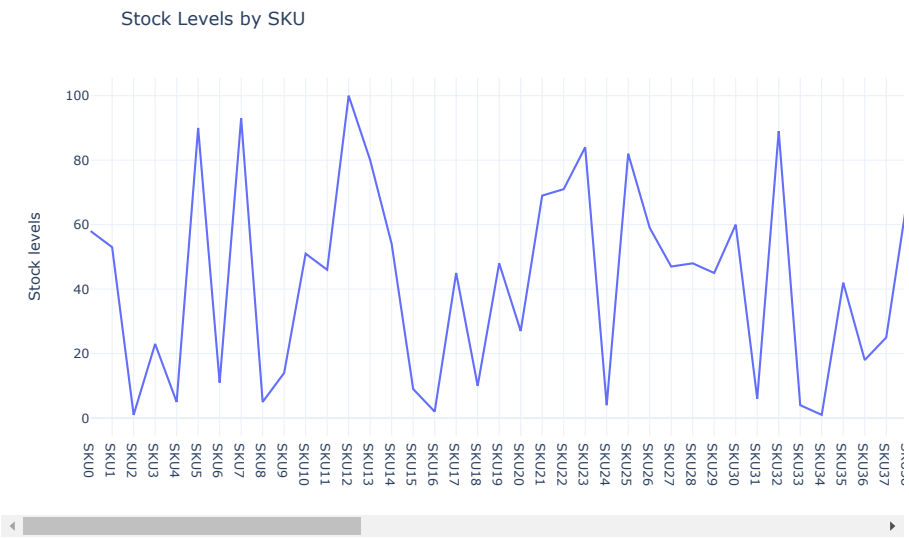
The **SKU**, or **Stock Keeping Unit**, serves as a crucial inventory management component for businesses. It provides a unique identifier for each product in a company's inventory, allowing for precise tracking and management. Consider a scenario in a bustling toy store where an extensive array of toys fills the shelves. While each toy possesses its own distinct characteristics such as name and price, the need arises to efficiently monitor and manage their inventory levels. Enter the **SKU**, a secret code assigned to each toy, enabling the store to accurately track stock levels, manage replenishments, and facilitate efficient operations. Thus, the **SKU** serves as a cornerstone in the realm of inventory management, ensuring seamless operations and optimal customer satisfaction.

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

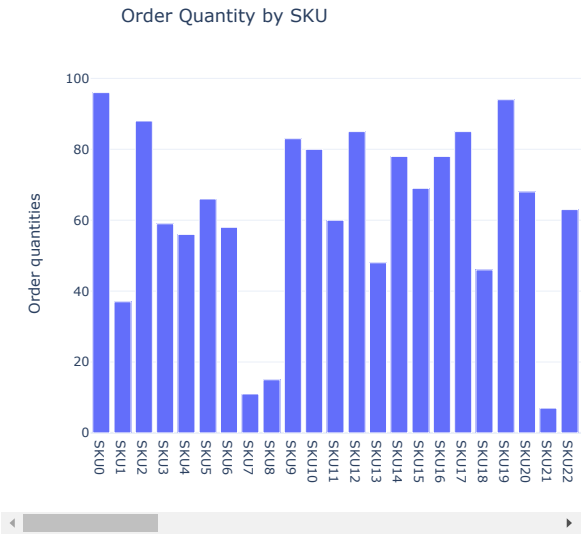


In the dataset, another column labeled "Stock levels" awaits exploration. Stock levels represent the quantified inventory count of products within a store or business. Let's delve into the stock levels attributed to each SKU, shedding light on the intricacies of inventory management within the company.

```
stock_chart = px.line(data, x='SKU',
                      y='Stock levels',
                      title='Stock Levels by SKU')
stock_chart.show()
```




```
#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

In the dataset, another column labeled "Stock levels" awaits exploration. Stock levels represent the quantified inventory count of products within a store or business. Let's delve into the stock levels attributed to each SKU, shedding light on the intricacies of inventory management within the company.

```
shipping_cost_chart = px.bar(data, x='Shipping carriers',
                             y='Shipping costs',
                             title='Shipping Costs by Carrier')
shipping_cost_chart.show()
```

Following our visualization revealing Carrier B's significant revenue contribution despite being the most costly among the carriers, let's delve into the distribution of costs across transportation modes. This analysis aims to provide insights into how expenses are allocated among different modes of transportation utilized by the company.

```
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()
```

Average Defect Rates

Within supply chain management, understanding and managing product quality is paramount. The defect rate, representing the percentage of products that exhibit flaws or damage post-shipping, serves as a critical metric for assessing quality control measures. In this analysis, we delve into the dataset to unveil the average defect rate across all product types. By examining this key indicator, we aim to gain insights into the overall quality performance within the supply chain, identifying areas for improvement and optimization.

```
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()
```

Having identified higher defect rates among haircare products, our focus now shifts to scrutinizing defect rates by mode of transportation. This analysis aims to unravel any correlations between transportation modes and product quality discrepancies. By delving into defect rates across different modes of transportation, we seek to uncover insights that could inform strategic decisions aimed at mitigating quality issues within the supply chain.