e-commerce-supply-chain-project

April 23, 2024

[16]:
```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
import plotly.colors as colors

pio.templates.default = "plotly_white"
```

[17]:
```
data = pd.read_csv("E-Commerce Supply Chain.csv")
```

[18]:
```
data
```

[18]:
```
   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
..          …      …        …           …                      …
95     haircare  SKU95  77.903927            65                      672
96    cosmetics  SKU96  24.423131            29                      324
97     haircare  SKU97   3.526111            56                       62
98     skincare  SKU98  19.754605            43                      913
99     haircare  SKU99  68.517833            17                      627

    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
..                …                   …           …         …
95        7386.363944               Unknown            15          14
96        7698.424766            Non-binary            67           2
97        4370.916580                  Male            46          19
98        8525.952560                Female            53           1
```

```
99        9185.185829              Unknown           55           8

    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
..               …  …       …       …                 …
95                26  …    Mumbai        18                 450
96                32  …    Mumbai        28                 648
97                 4  …    Mumbai        10                 535
98                27  …   Chennai        28                 581
99                59  …   Chennai        29                 921

   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
..                     …                 …                 …
95                      26           58.890686             Pending
96                      28           17.803756             Pending
97                      13           65.765156                Fail
98                       9            5.604691             Pending
99                       2           38.072899                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
..           …                   …      …         …
95      1.210882                   Air  Route A  778.864241
96      3.872048                  Road  Route A  188.742141
97      3.376238                  Road  Route A  540.132423
98      2.908122                  Rail  Route A  882.198864
99      0.346027                  Rail  Route B  210.743009

[100 rows x 24 columns]
```

[19]:

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

[19]:

```
Product type               0
SKU                        0
```

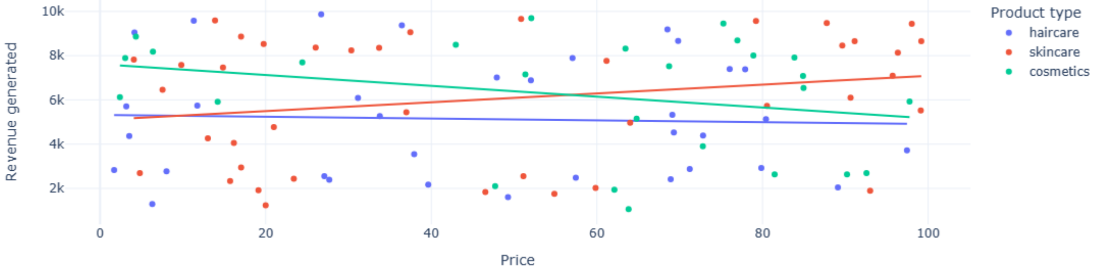
```
Price                      0
Availability               0
Number of products sold    0
Revenue generated          0
Customer demographics      0
Stock levels               0
Lead times                 0
Order quantities           0
Shipping times             0
Shipping carriers          0
Shipping 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
```

[20]:
```
data.columns
```

[20]:
```
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')
```

0.1 Relation of price of the product and Revenue generated by them.

[29]:
```
fig = px.scatter(data, x='Price', y='Revenue generated', color= 'Product type',
                 hover_data =['Number of products sold'],
                    trendline="ols")
fig.show()
```



1 Sales by Product type

```
[33]: sales_type= data.groupby('Product type')['Number of products sold'].sum().
        ↪reset_index()
```

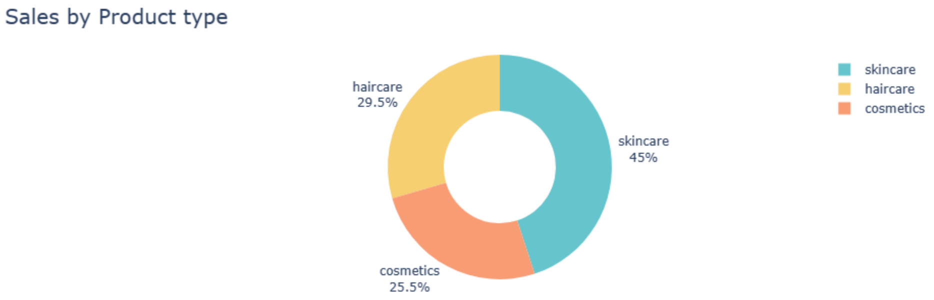
```
[34]: sales_type
```

```
[34]:   Product type  Number of products sold
      0    cosmetics                    11757
      1     haircare                    13611
      2     skincare                    20731
```

```
[32]: sales_type= data.groupby('Product type')['Number of products sold'].sum().
        ↪reset_index()

      fig = px.pie(sales_type, values= 'Number of products sold',
                   names= 'Product type',  hole=0.5,
                   hover_data =['Number of products sold'],
                   color_discrete_sequence= px.colors.qualitative.Pastel)
      # Donut chart so there is a hole
      fig.update_traces (textposition = 'outside', textinfo= 'percent+label' )
      fig.update_layout (title_text= 'Sales by Product type',title_font=␣
        ↪dict(size=20))

      fig.show()
```



2 Total Revenue generated from Shipping carriers

[37]:

```
sales_revenue_shipping_carrier = data.groupby('Shipping carriers')['Revenue␣
 ↪generated'].sum().reset_index()
```

[38]:

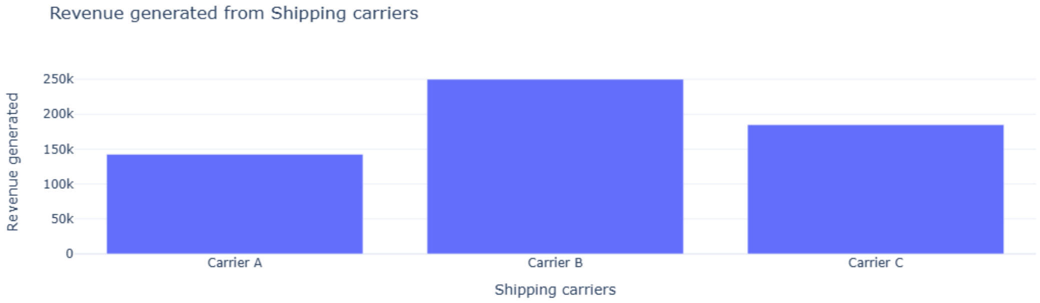
```
sales_revenue_shipping_carrier
```

[38]:

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

[40]:

```
sales_revenue_shipping_carrier = data.groupby('Shipping carriers')['Revenue␣
 ↪generated'].sum().reset_index()
fig=go.Figure()
fig.add_trace (go.Bar(x= sales_revenue_shipping_carrier['Shipping carriers'],
                      y=sales_revenue_shipping_carrier['Revenue generated']))
fig.update_layout (title_text= 'Revenue generated from Shipping carriers',
                   xaxis_title= 'Shipping carriers', yaxis_title= 'Revenue␣
 ↪generated')
fig.show()
```



3 Revenue generated by each SKU

[41]:

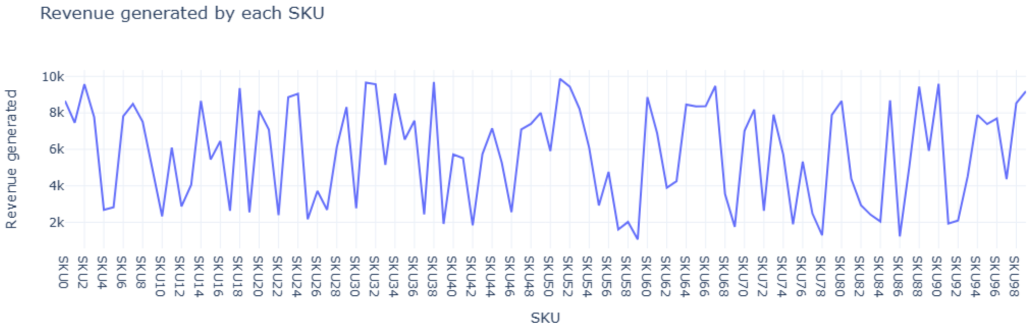
```
data.columns
```

[41]:

```
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')
```

[42]:

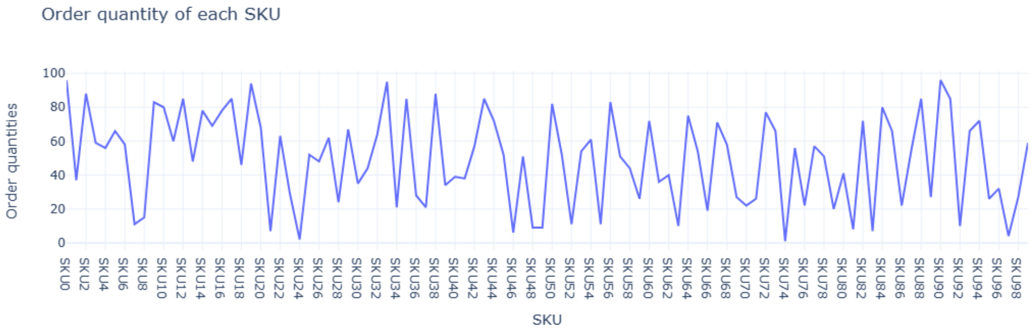
```
revenue_chart = px.line (data, x= 'SKU', y = 'Revenue generated', title =␣
 ↪'Revenue generated by each SKU')
revenue_chart.show()
```



4 Order quantity of each SKU

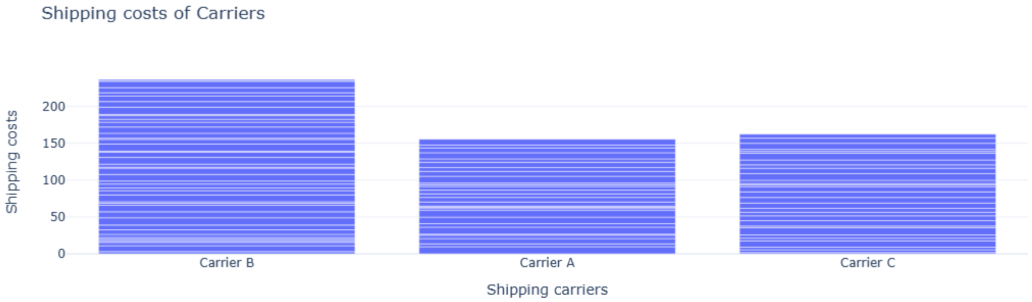
[43]:

```
revenue_Sku_quantity = px.line (data, x= 'SKU', y = 'Order quantities', title =␣
 ↪'Order quantity of each SKU')
revenue_Sku_quantity.show()
```



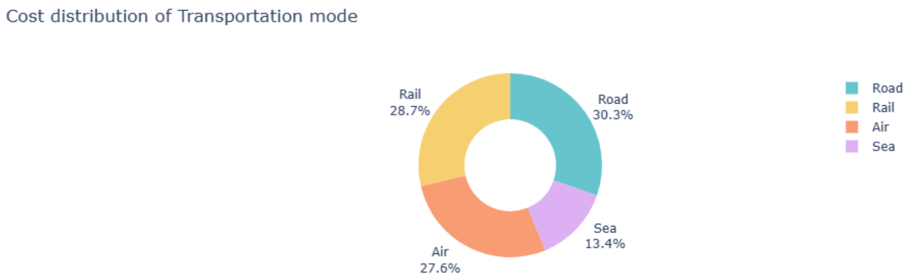
5 Shipping costs of Carriers

[45]:
```
shipping = px.bar (data, x= 'Shipping carriers', y = 'Shipping costs', title =␣
  ↪'Shipping costs of Carriers')
shipping.show()
```



6 Cost distribution of Transportation mode

[50]:
```
fig = px.pie(data, values= 'Costs',
             names= 'Transportation modes', title = 'Cost distribution of␣
  ↪Transportation mode' ,
             hole=0.5,
             hover_data =['Number of products sold'],
             color_discrete_sequence= px.colors.qualitative.Pastel)
fig.update_traces (textposition = 'outside', textinfo= 'percent+label' )
fig.show()
```



7 Defect Rate during Shipping

```
[52]: data.columns
```

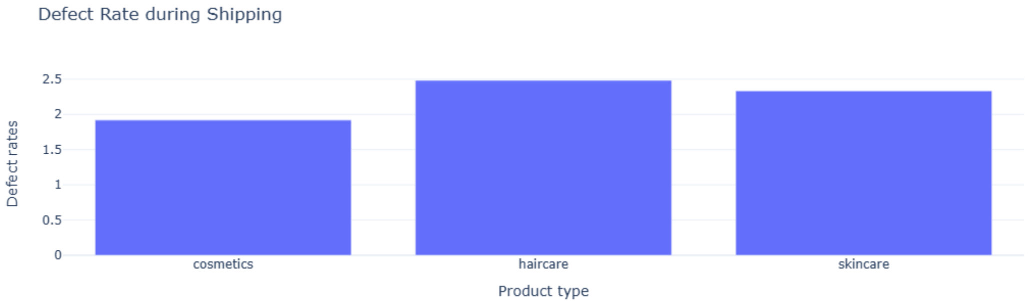
```
[52]: Index(['Product type', 'SKU', 'Price', 'Availability',
             'Number of products sold', 'Revenue generated', 'Customer demographics',
             'Stock levels', 'Lead times', 'Order quantities', 'Shipping times',
             'Shipping carriers', 'Shipping costs', 'Supplier name',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

```
[55]: Defect_rate = data.groupby('Product type')['Defect rates'].mean().reset_index()
```

```
[56]: Defect_rate
```

```
[56]:   Product type  Defect rates
      0    cosmetics      1.919287
      1     haircare      2.483150
      2     skincare      2.334681
```

```
[54]: Defect_rate = data.groupby('Product type')['Defect rates'].mean().reset_index()
      fig = px.bar (Defect_rate, x= 'Product type', y = 'Defect rates', title =␣
       ↪'Defect Rate during Shipping')
      fig.show()
```


Defect Rate during Shipping
Defect rates
2.5
2
1.5
1
0.5
0
cosmetics
haircare
skincare
Product type

[ ]: