In [2]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
df=pd.read_csv("D:\\Summer Training Video\\ML\\supply_chain.csv")
```

In [4]:
```
df
```

Out[4]:

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58	7
1	skincare	SKU1	14.843523	95	736	7460.900065	Female	53	30
2	haircare	SKU2	11.319683	34	8	9577.749626	Unknown	1	10
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23	13
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5	3
...	...	...	...	...	...	...	...	...	...
95	haircare	SKU95	77.903927	65	672	7386.363944	Unknown	15	14
96	cosmetics	SKU96	24.423131	29	324	7698.424766	Non-binary	67	2
97	haircare	SKU97	3.526111	56	62	4370.916580	Male	46	19
98	skincare	SKU98	19.754605	43	913	8525.952560	Female	53	1
99	haircare	SKU99	68.517833	17	627	9185.185829	Unknown	55	8

100 rows × 24 columns

In [5]:
```
df.head()
```

Out[5]:

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times	q
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58	7	
1	skincare	SKU1	14.843523	95	736	7460.900065	Female	53	30	
2	haircare	SKU2	11.319683	34	8	9577.749626	Unknown	1	10	
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23	13	
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5	3	

5 rows × 24 columns

In [6]:
```
import plotly.express as px
import plotly.io as pio
import plotly.graph_objects as go
pio.templates.default = "plotly_white"
```

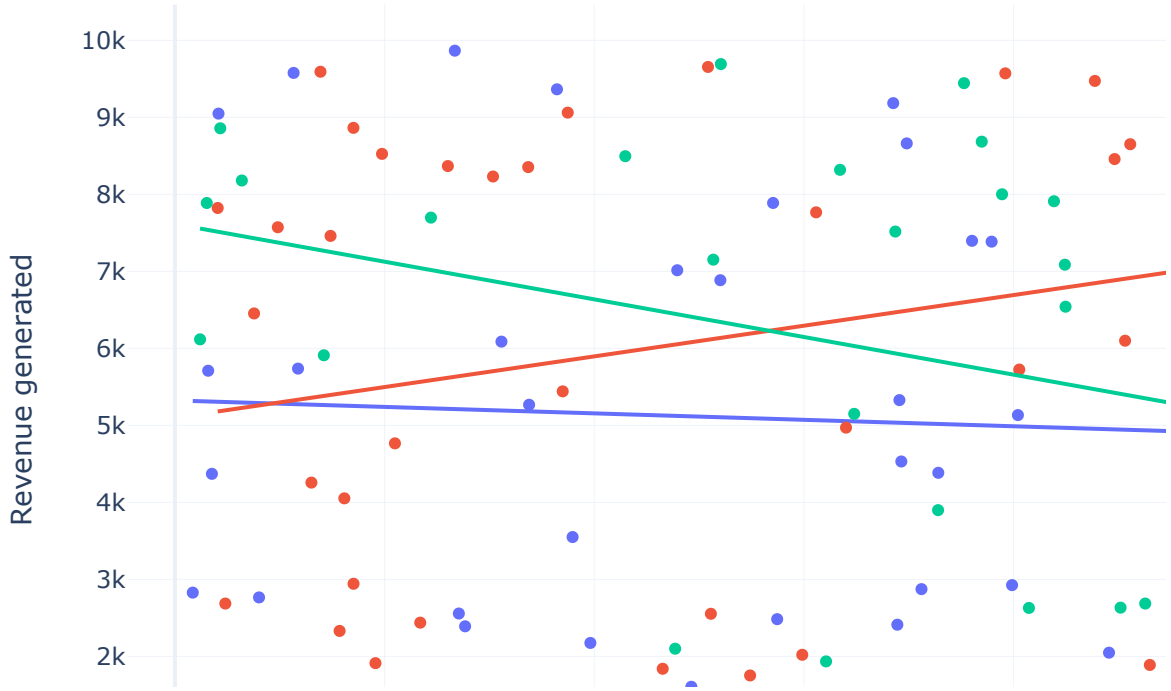
In [7]:
```
df.describe()
```

Out[7]:

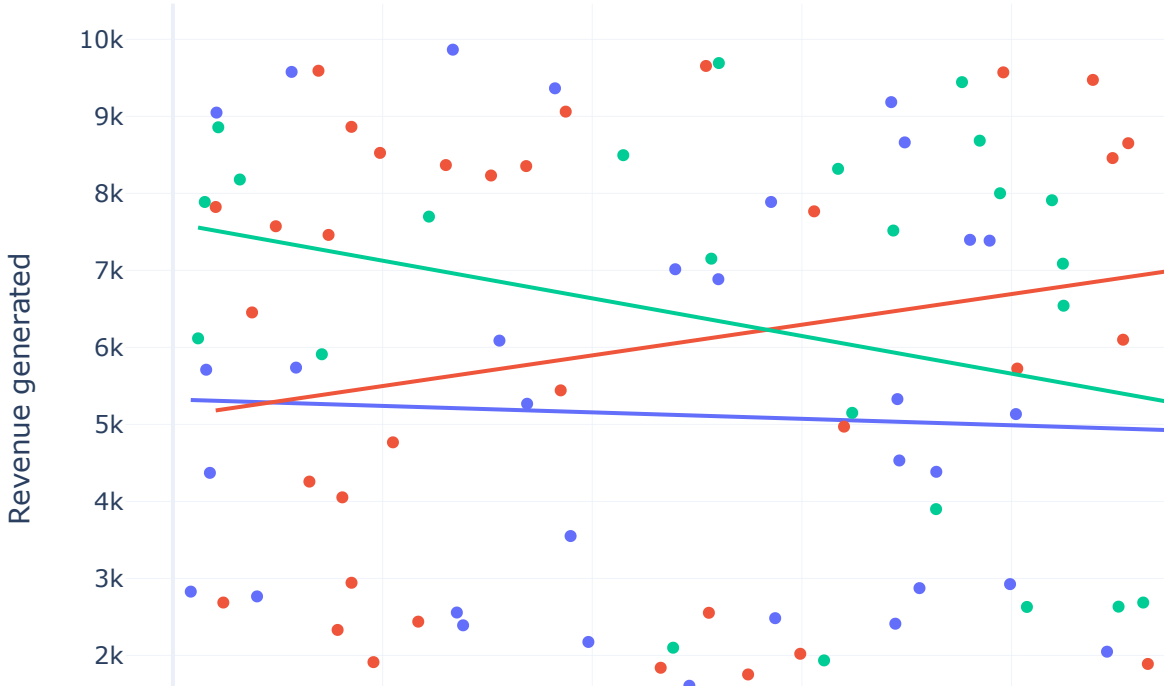
	Price	Availability	Number of products sold	Revenue generated	Stock levels	Lead times	Order quantities	S
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100
mean	49.462461	48.400000	460.990000	5776.048187	47.770000	15.960000	49.220000	5
std	31.168193	30.743317	303.780074	2732.841744	31.369372	8.785801	26.784429	2
min	1.699976	1.000000	8.000000	1061.618523	0.000000	1.000000	1.000000	1
25%	19.597823	22.750000	184.250000	2812.847151	16.750000	8.000000	26.000000	3
50%	51.239831	43.500000	392.500000	6006.352023	47.500000	17.000000	52.000000	6
75%	77.198228	75.000000	704.250000	8253.976921	73.000000	24.000000	71.250000	8
max	99.171329	100.000000	996.000000	9866.465458	100.000000	30.000000	96.000000	10

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:

In [8]:
```
fig = px.scatter(df, x='Price',
                 y='Revenue generated',
                 color='Product type',
                 hover_data=['Number of products sold'],
                 trendline="ols")
fig.show()
```

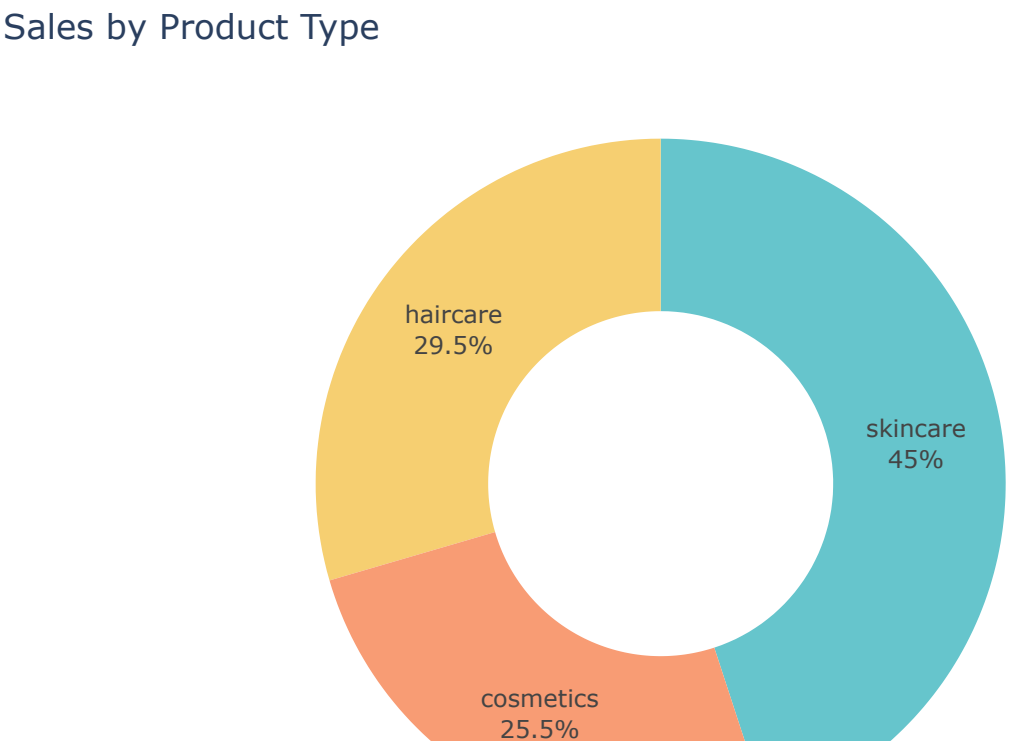


In [9]:
```
fig.show()
```



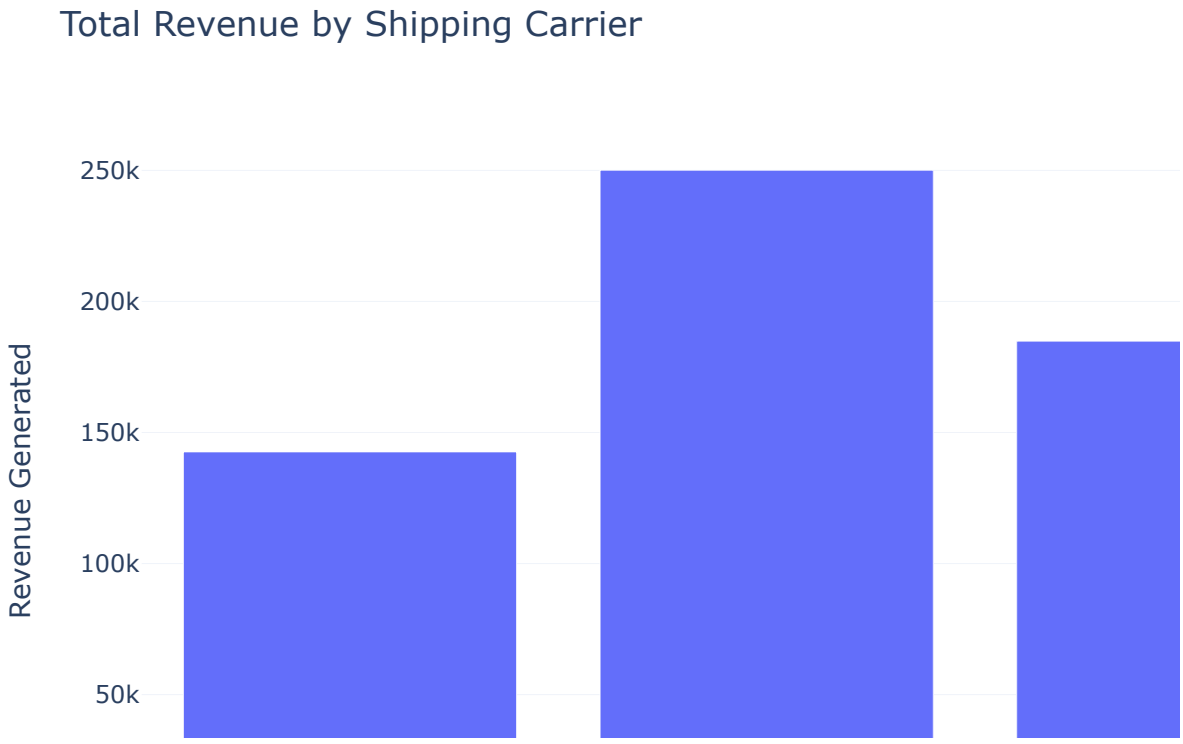
Thus the company derives more revenue from skincare products and the higher the price of skincare products, the more revenue they generate. Now let's have a look at the sales by product type :

In [16]:
```
sales_data = df.groupby('Product type')['Number of products sold'].sum().reset_
#sales_data

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 the skincare products , 29.5% from haircare and 25.5% from cosmetics. Now let's have a look at the total revenue generated from shipping carries:

In [27]:
```
total_revenue = df.groupby('Shipping carriers')['Revenue generated'].sum().rese
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 carrier for transporatation, and Carrier B helps the company is generating more revenue. Now let's have a look at the Average lead time and Average Manufacturing Costs for all products of the company:

In [33]:
```
avg_lead_time = df.groupby('Product type')['Lead time'].mean().reset_index()
#avg_lead_time
avg_manufacturing_costs = df.groupby('Product type')['Manufacturing costs'].mea
result = pd.merge(avg_lead_time, avg_manufacturing_costs, on='Product type')
result.rename(columns={'lead time' : 'Average Lead Time',
                       'Manufacturing costs' : 'Average Manufacturing'}, inplac
print(result)
```

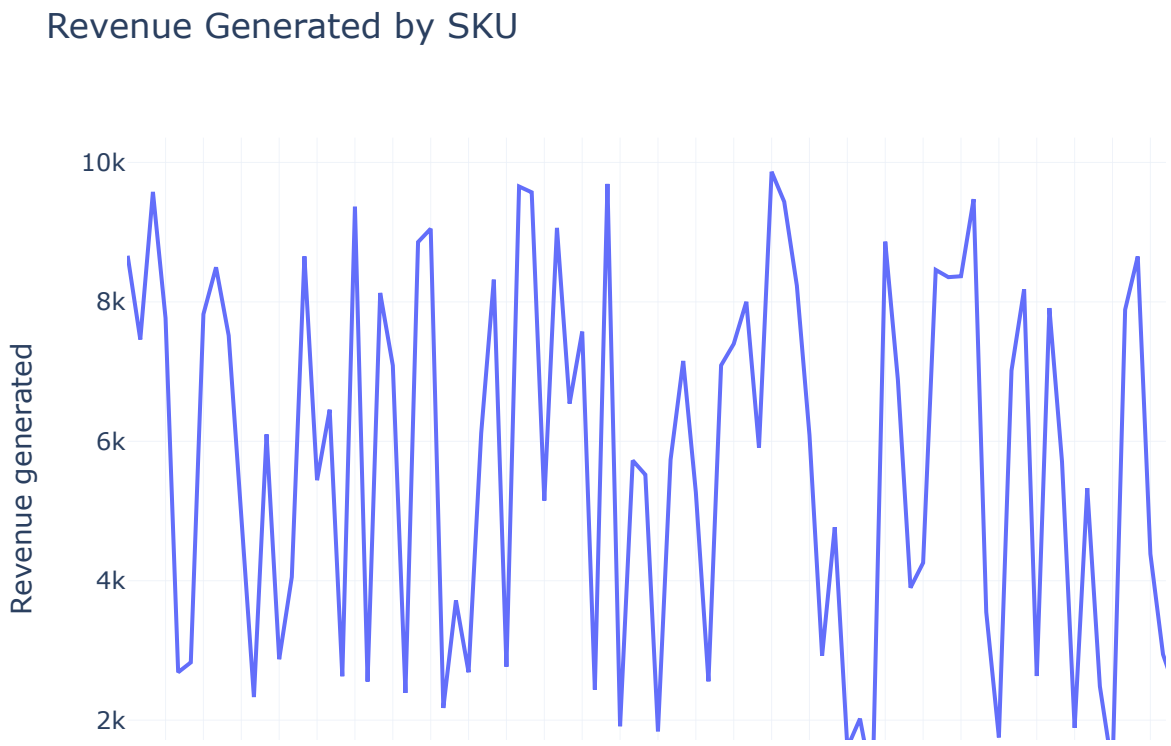
```
  Product type  Lead time  Average Manufacturing
0    cosmetics  13.538462              43.052740
1     haircare  18.705882              48.457993
2     skincare  18.000000              48.993157
```

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 specially 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.

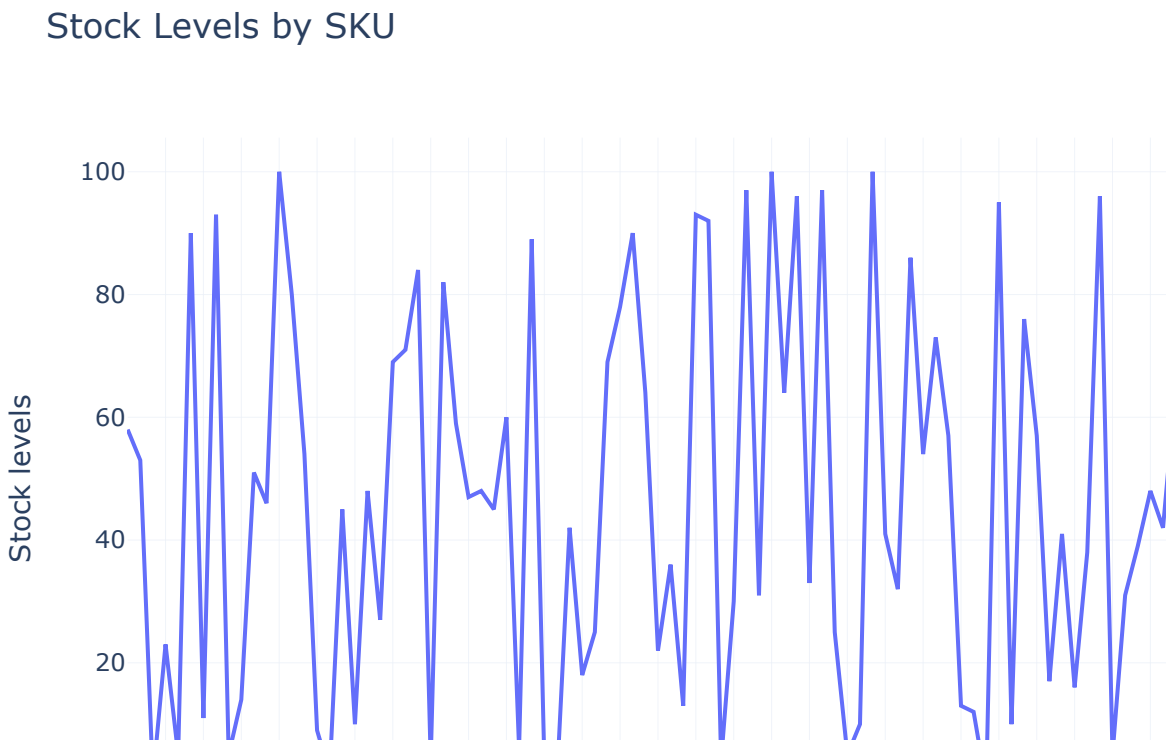
I hope you have now understand what's SKU. Now let's analyze the revenue generated bu each SKU.

In [34]:
```
revenue_chart = px.line(df, x='SKU',
                        y = 'Revenue generated',
                        title = 'Revenue Generated by SKU')
revenue_chart.show()
```



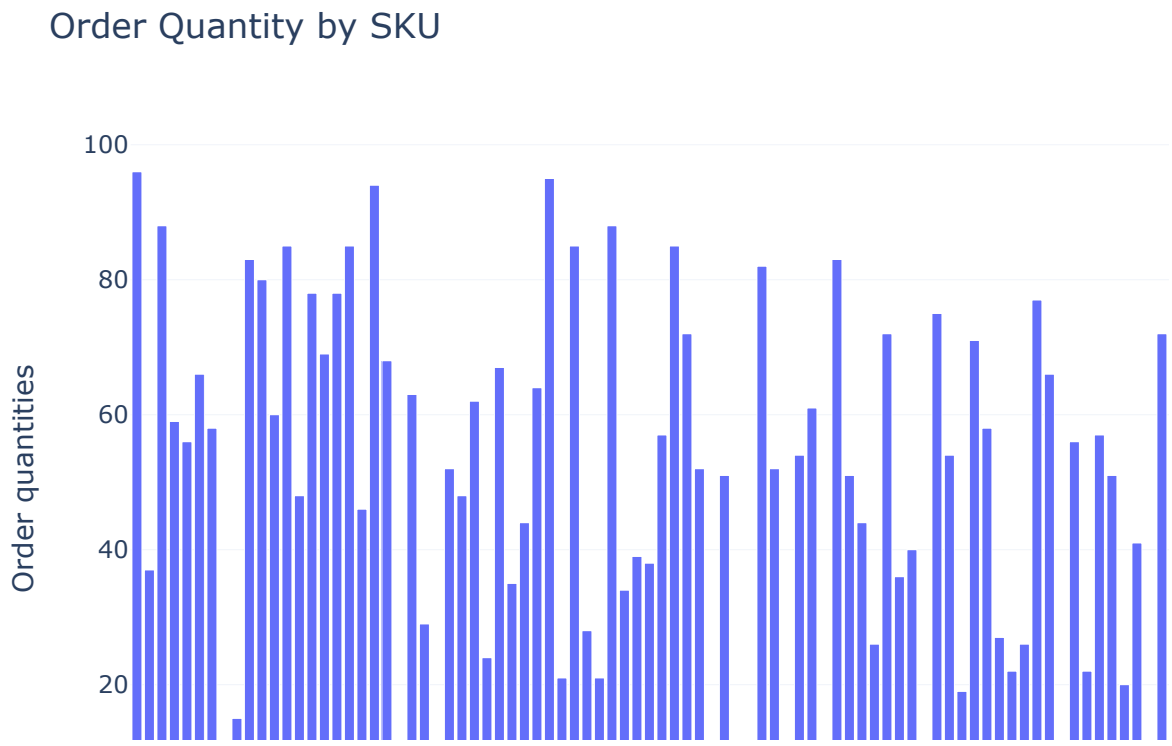
There's another column in the dataset as Stock levels. Stock levels refer to the number of products a store or business has in its inventory. Now let's have a look at stock levels of each SKU:

In [38]:
```
stock_chart = px.line(df, 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:

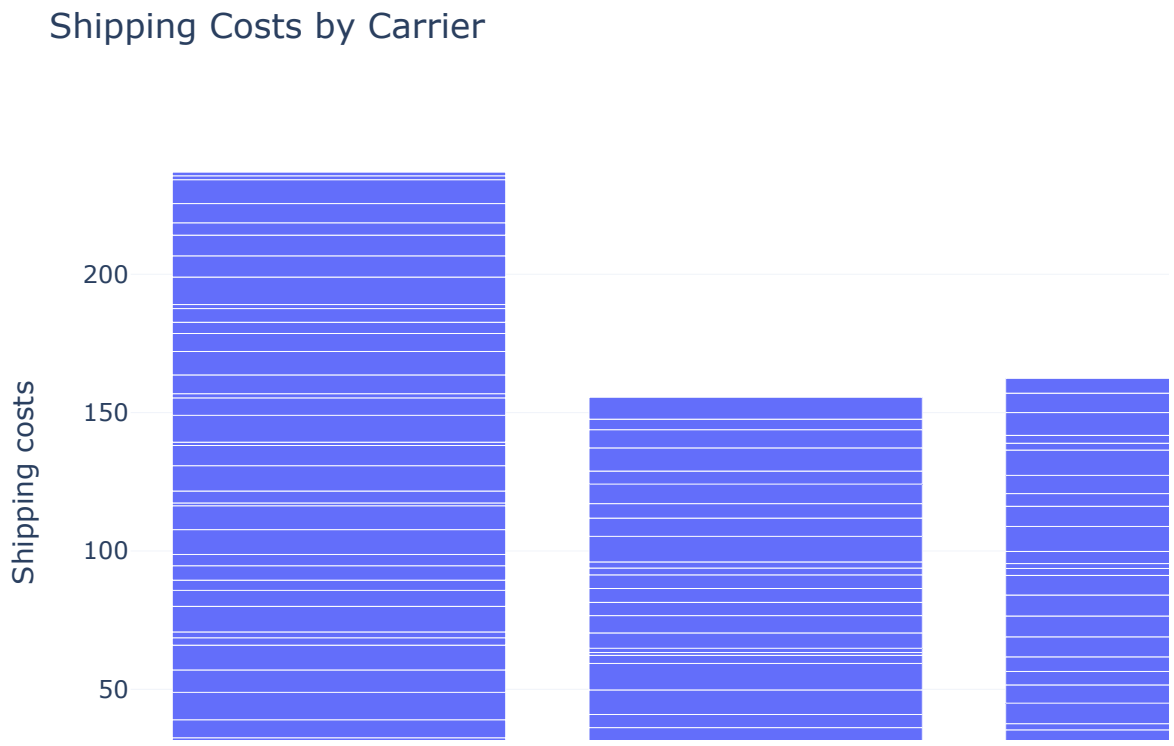
In [39]:
```
order_quantity_chart = px.bar(df, x = 'SKU',
                              y = 'Order quantities' ,
                              title = 'Order Quantity by SKU')
order_quantity_chart
```



Cost Analysis

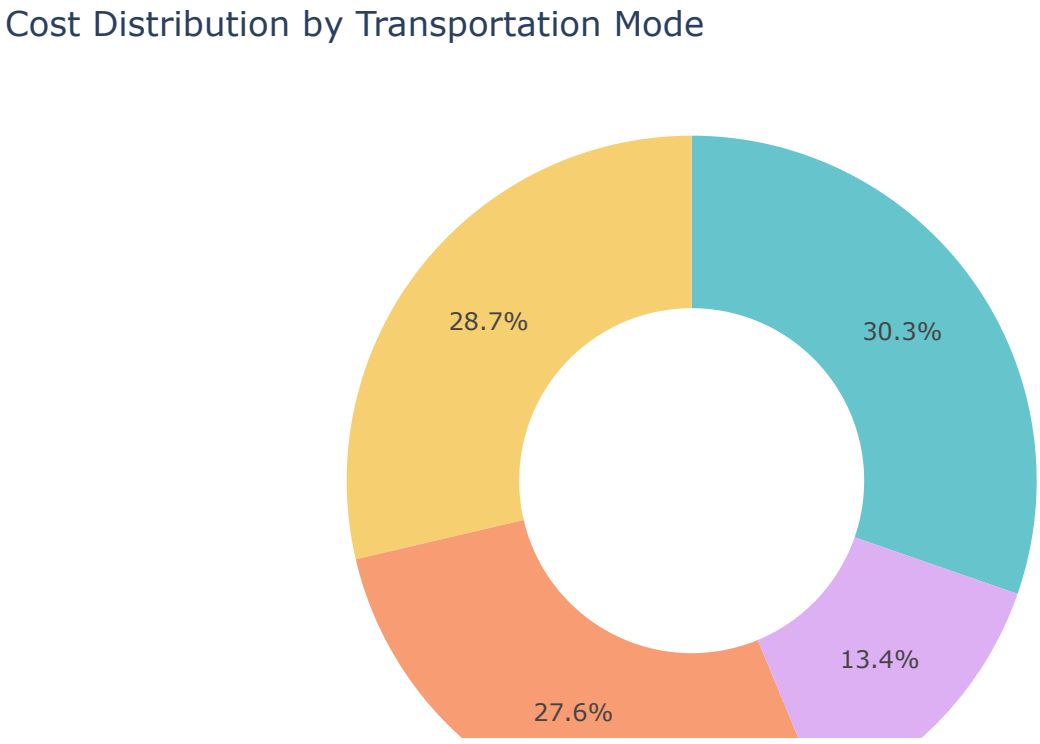
Now let's analyze the shipping cost of Carriers:

In [40]:
```
shipping_cost_chart = px.bar(df, 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 is 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:

In [46]:
```
transportation_chart = px.pie(df,
                              values = 'Costs',
                              names = 'Transportation modes' ,
                              title = 'Cost Distribution by Transportation Mode'
                              hole = 0.5,
                              color_discrete_sequence = px.colors.qualitative.Pa
transportation_chart.show()
```

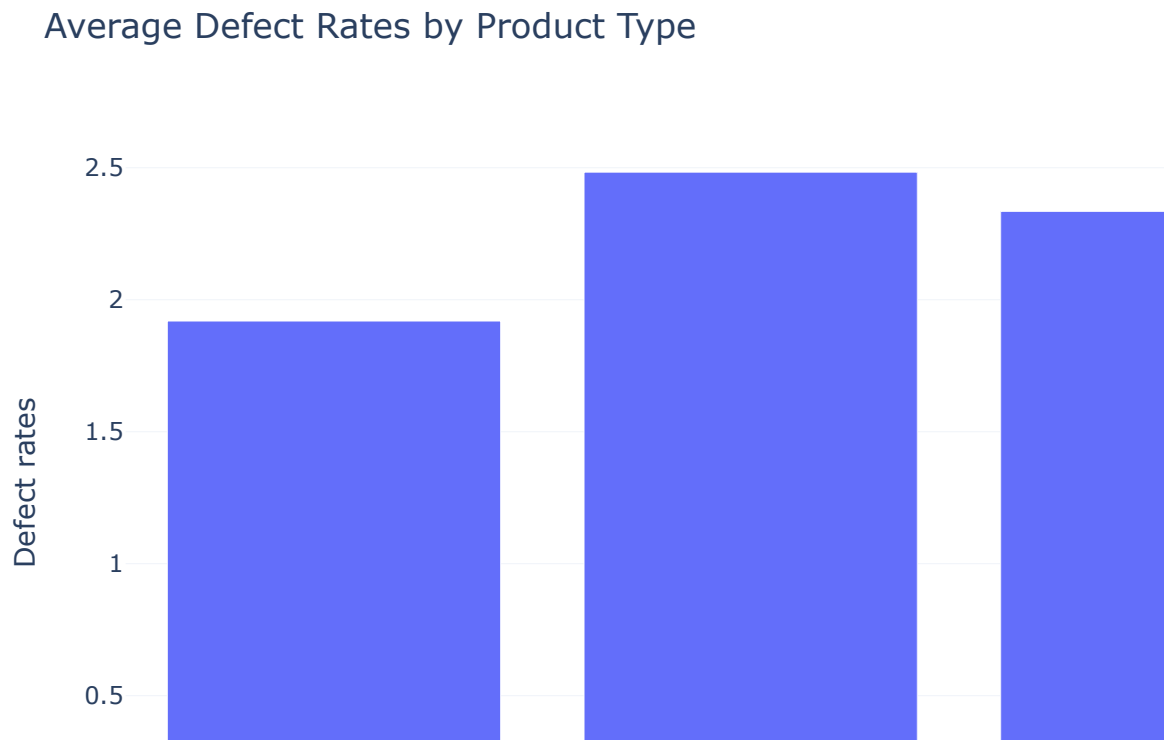


So the company spends more on Road and Rail modes of transportation for the transportation of Goods.

Analyzing Defect Rate

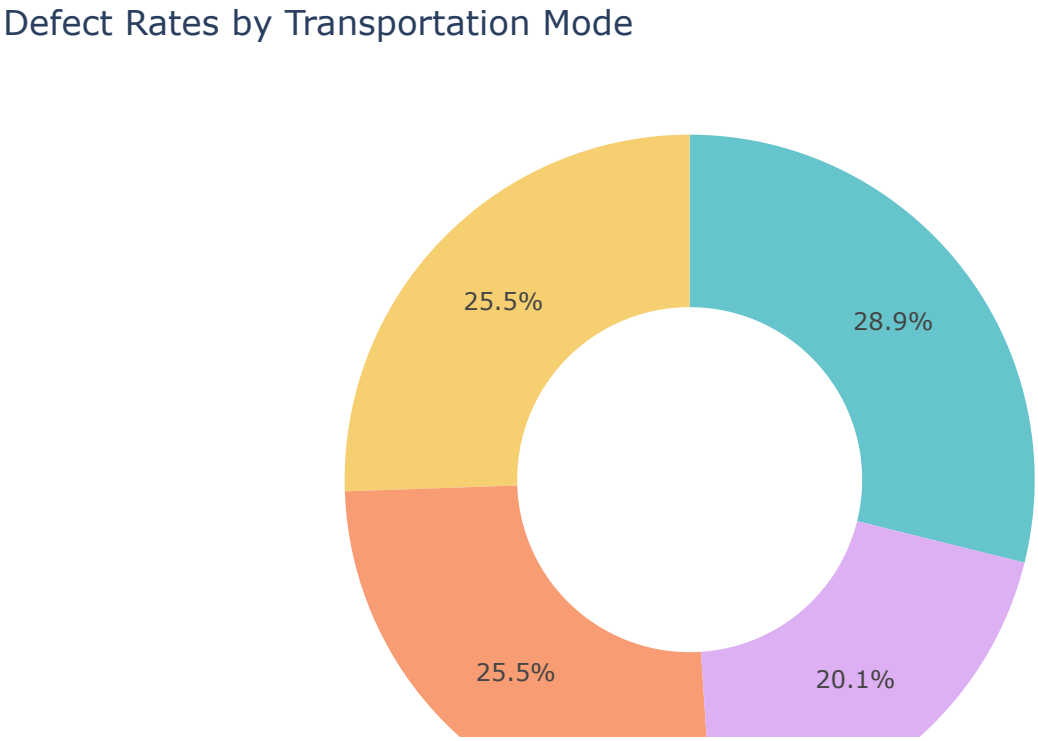
The defect rate in the supply chain refers to the percentage of products that have something wrong or are found broken after shipping. Let,s have a look at the average defect rate of all product types:

In [48]:
```
defect_rates_by_product = df.groupby('Product type')['Defect rates'].mean().res
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 modes of transportation:

In [49]:
```
pivot_table = pd.pivot_table(df, 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.Pa
transportation_chart.show()
```



In [ ]: