

```
In [ ]: 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	69.808806	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]

```
In [ ]: import plotly.io as pio
pio.renderers.default = 'notebook'
```

```
In [ ]: 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

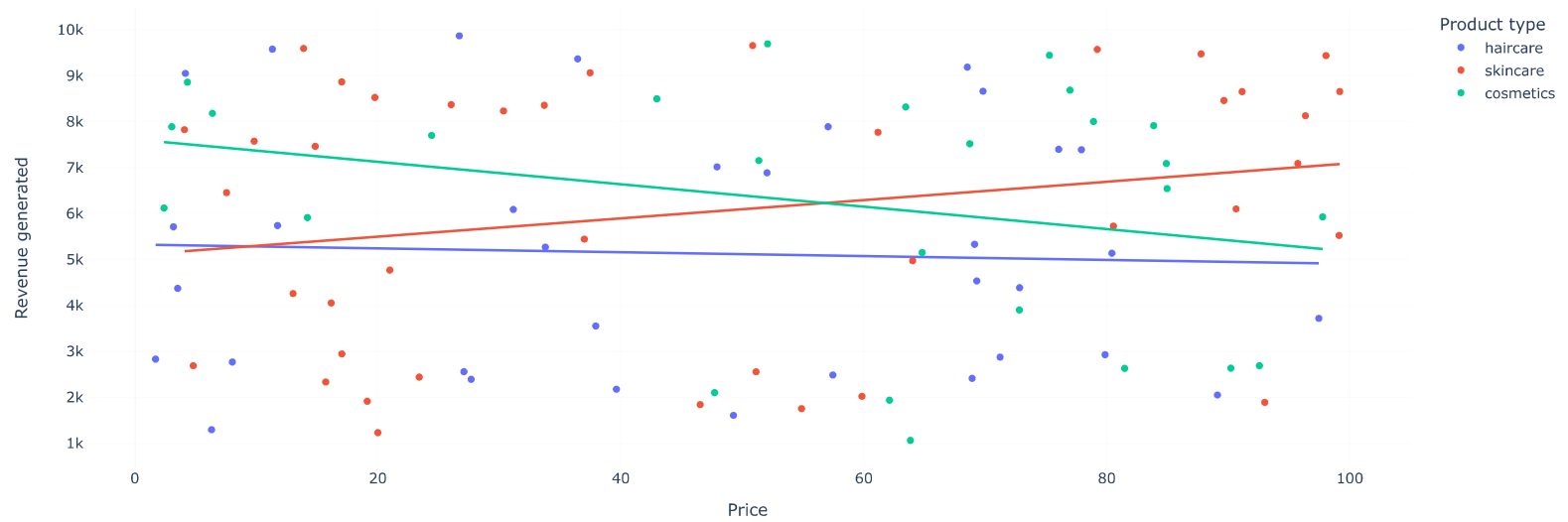
	Shipping costs	Lead time	Production volumes \
count	100.000000	100.000000	100.000000
mean	5.548149	17.080000	567.840000
std	2.651376	8.846251	263.046861
min	1.013487	1.000000	104.000000
25%	3.540248	10.000000	352.000000
50%	5.320534	18.000000	568.500000
75%	7.601695	25.000000	797.000000
max	9.929816	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

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

)

fig.show()
```



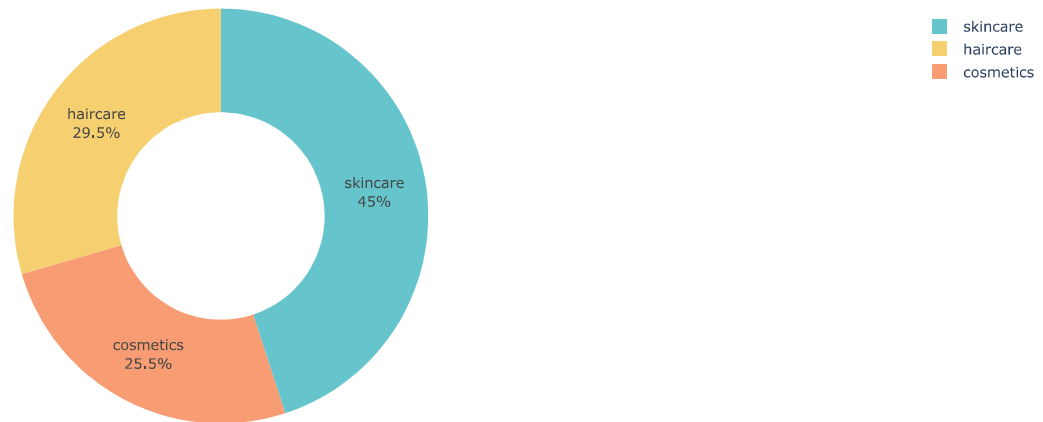
Thus, the company derives more revenue from skincare products, and the higher the price of skincare products, the more revenue they generate.

The next line of code looks at the sales by product type:

```
In [ ]: 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()
```

Sales by Product Type

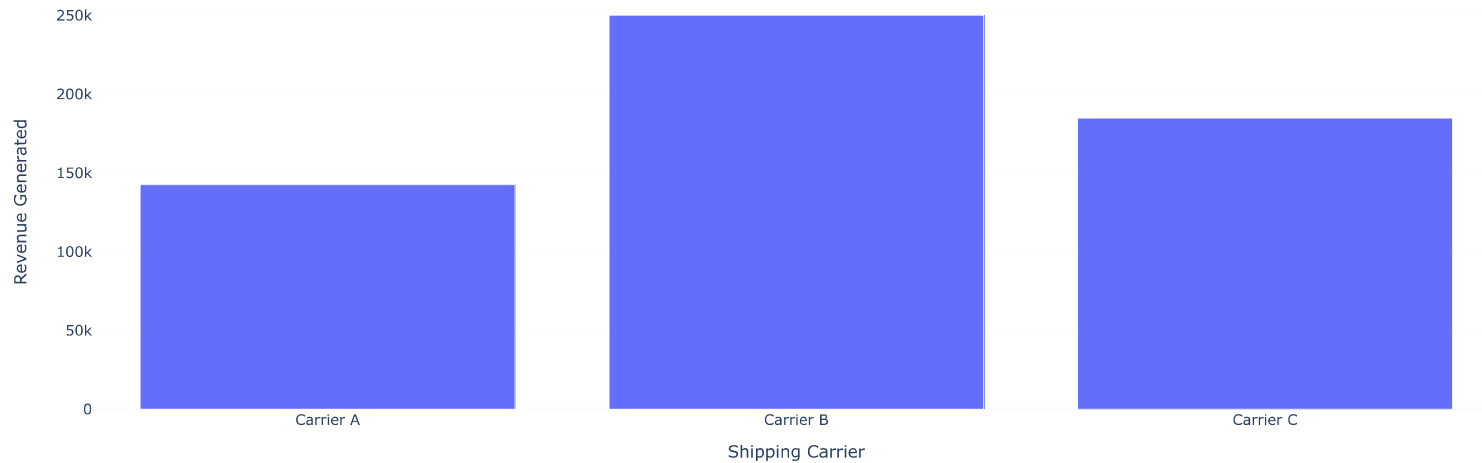


45% of the business comes from skincare products, 29.5% from haircare, and 25.5% from cosmetics.

What if we want to know the total revenue from shipping carriers?

```
In [ ]: 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()
```

Total Revenue by shipping carrier



The bar chat shows us that the company uses three carriers for transportation, and Carrier B helps the company in generating more revenue.

What is the average lead time and the average manufacturing cost for all products of the company

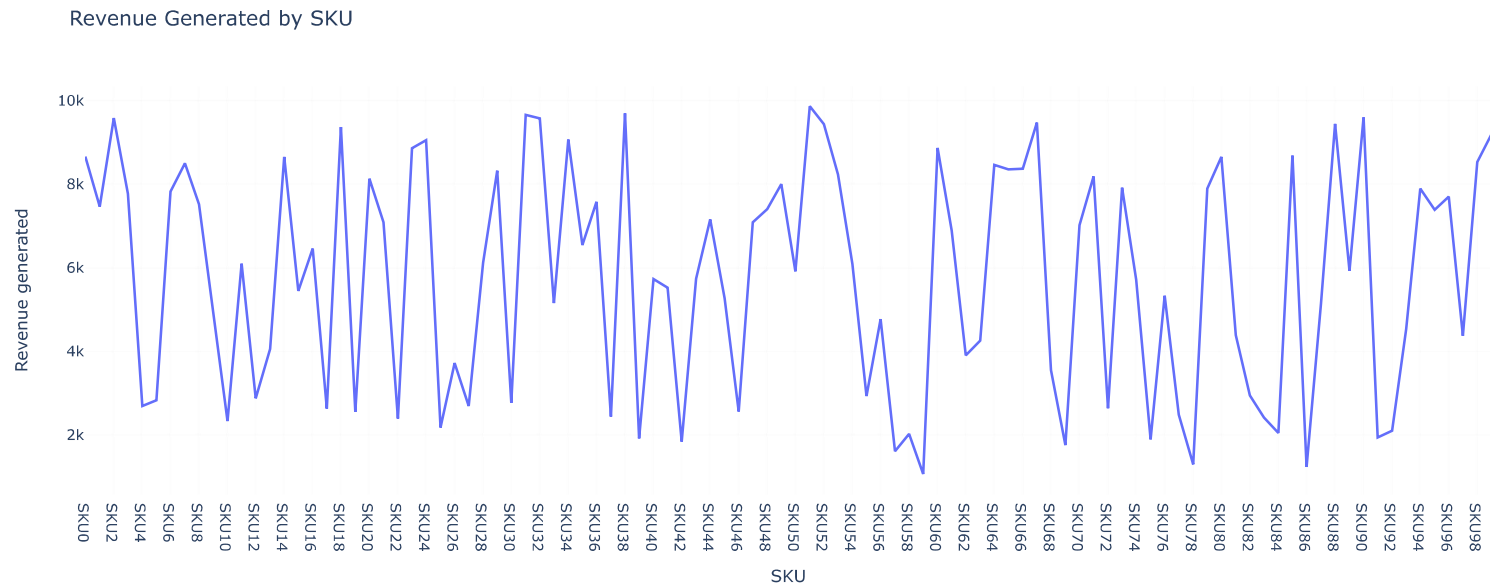
```
In [ ]: avg_lead_time = data.groupby('Product type')['Lead time'].mean().reset_index()
avg_manufacturing_costs = data.groupby('Product type')['Manufacturing costs'].mean().reset_index()
result = pd.merge(avg_lead_time, avg_manufacturing_costs, on='Product type')
result.rename(columns={'lead time': 'Average Lead Time', 'Manufacturing costs': 'Average Manufacturing Costs'}, inplace=True)
print(result)
```

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

SKU's are generally being used for keeping track of the amount of stock/merchandise readily available in a company.

let's track the revenue each SKU produced

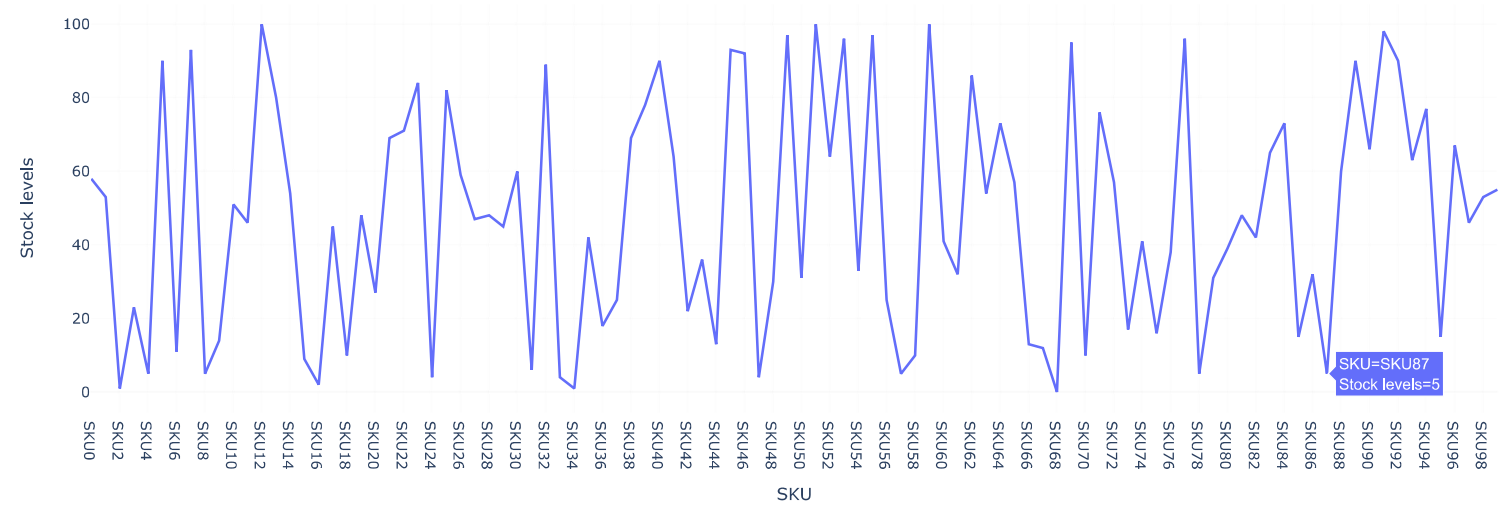
```
In [ ]: revenue_chart = px.line(data, x='SKU',
                                y= 'Revenue generated',
                                title='Revenue Generated by SKU')
revenue_chart.show()
```



We could also track the Stock level which is the number of products a store or business has in its inventory.

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

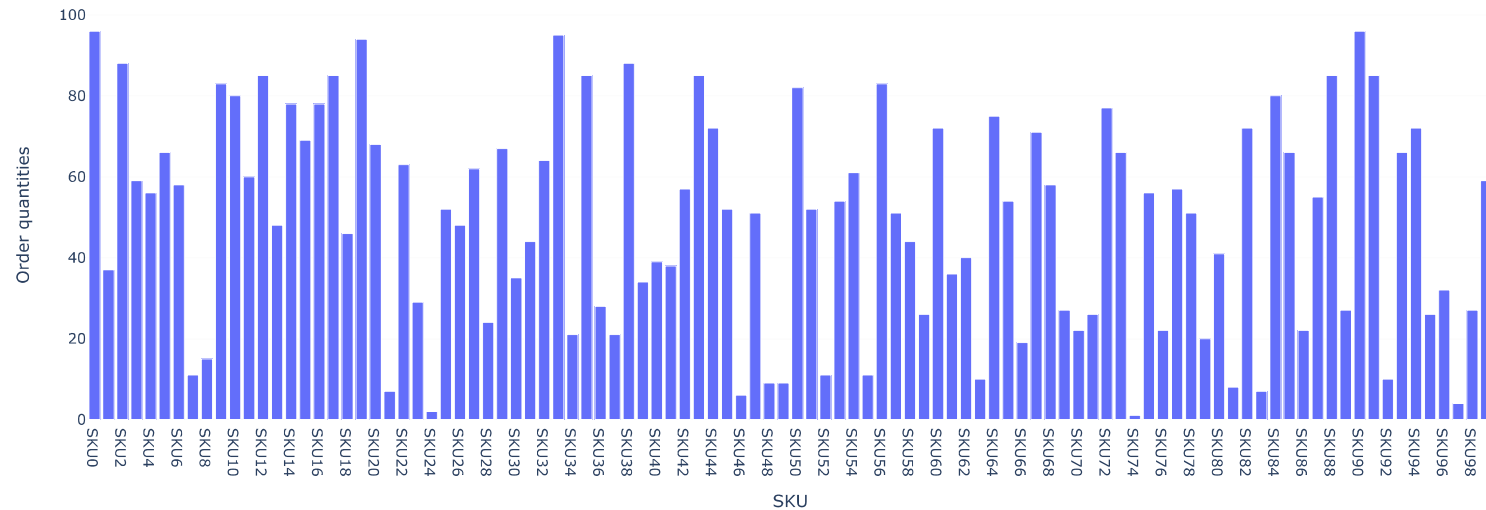
Stock Levels by SKU



What is the order quantity for each SKU:

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

Order Quantity by SKU



What if we want to know the shipping cost of each carrier

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


Shipping Costs by Carrier

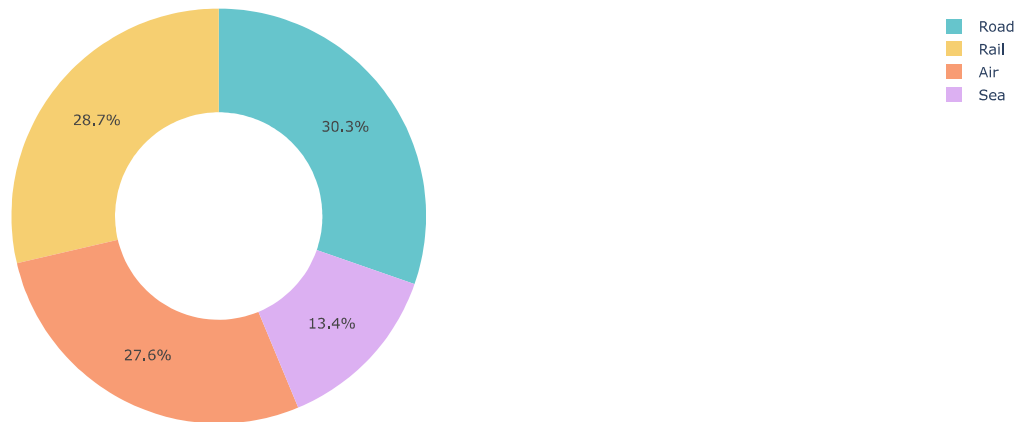


In one of the above charts we realized carrier B helps the company more in revenue generation. However the same carrier cost the company the most of the 3.

What is the cost of distribution by transportation mode:

```
In [ ]: 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()
```

Cost Distribution by transportation Mode



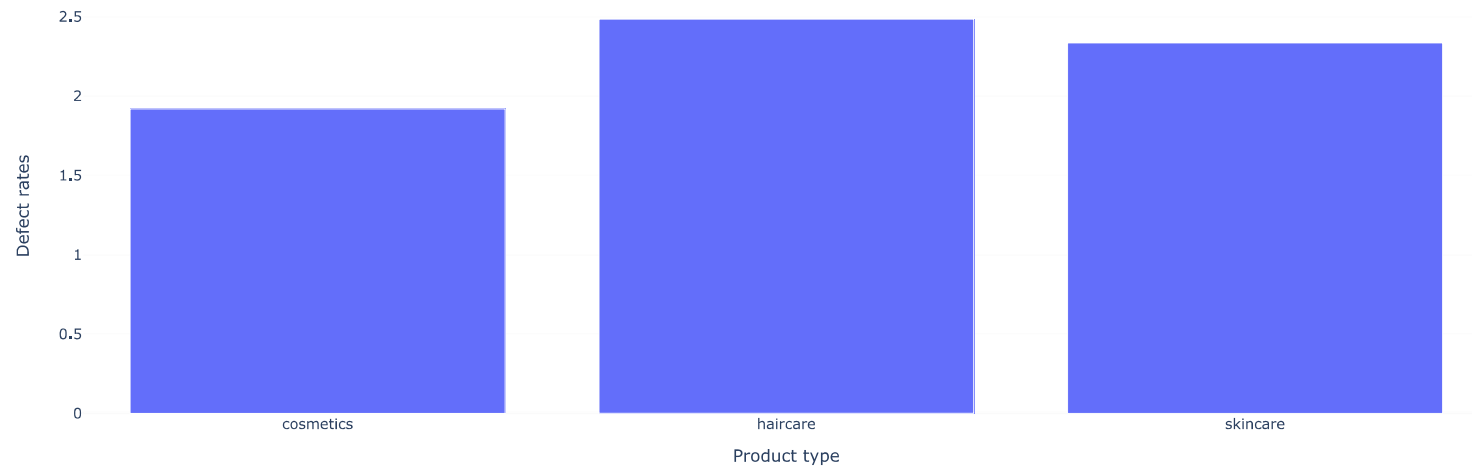
The company spends more on Road and Rail mode of transportation

Defect rate analysis, showing the percentage that have something wrong or are found broken after shipping.

what is the average defect rate of all product types:

```
In [ ]: 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()
```

Average Defect Rates by Product Type



Hair products have the highest defect rate.

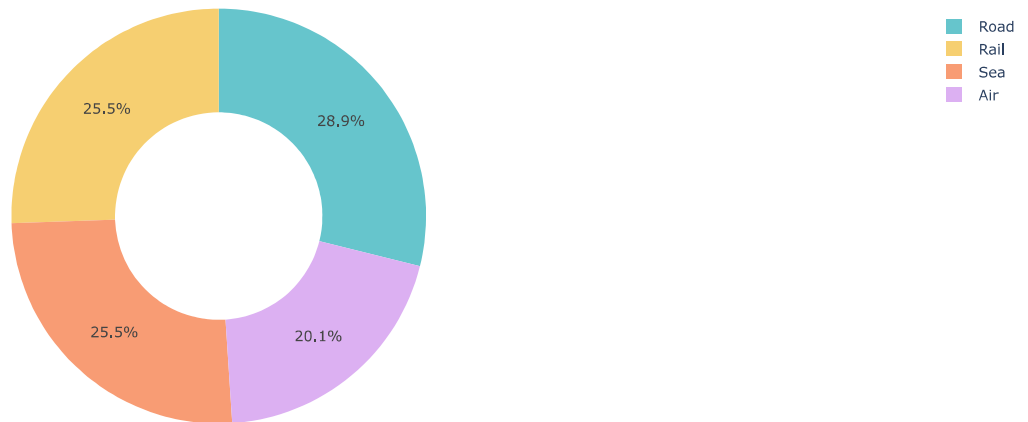
What is the defect rate by mode of transportation:

```
In [ ]: 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()
```

Defecr Rates by Transportation Mode



Road transportation has the highest defect rate, and Air transportation has the lowest defect rate.

All analysis done with python programming

In summary the purpose of this exploration was to analyze various components of the Supply Chain to understand how to improve the effectiveness of the Supply Chain to create more value for customers.

In []: