

Supply Chain Data Analysis

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
```

In [] :

```
data = pd.read_csv('supply_chain_data.csv')
data
```

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times	Order quantities	...	Location	Lead time	Production volumes	Manufacturing lead time	Manufacturing costs	Inspection results	Defect rates	Transportation modes	Routes	Costs
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58	7	96	...	Mumbai	29	215	29	46.279879	Pending	0.226410	Road	Route B	187.752075
1	skincare	SKU1	14.843523	95	736	7460.900055	Female	53	30	37	...	Mumbai	23	517	30	33.616769	Pending	4.854068	Road	Route B	503.065579
2	haircare	SKU2	11.319683	34	8	9577.749626	Unknown	1	10	88	...	Mumbai	12	971	27	30.688019	Pending	4.580593	Air	Route C	141.920282
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23	13	59	...	Kolkata	24	937	18	35.624741	Fail	4.746649	Rail	Route A	254.776159
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5	3	56	...	Delhi	5	414	3	92.065161	Fail	3.145590	Air	Route A	923.440632
...
95	haircare	SKU95	77.902927	65	672	7386.363944	Unknown	15	14	26	...	Mumbai	18	450	26	58.890666	Pending	1.210882	Air	Route A	778.864241
96	cosmetics	SKU96	24.423131	29	324	7698.424766	Non-binary	67	2	32	...	Mumbai	28	648	28	17.893756	Pending	3.872048	Road	Route A	188.742141
97	haircare	SKU97	3.526111	56	62	4370.916580	Male	46	19	4	...	Mumbai	10	535	13	65.765156	Fail	3.376238	Road	Route B	540.132423
98	skincare	SKU98	19.754605	43	913	8525.952560	Female	53	1	27	...	Chennai	28	581	9	5.604691	Pending	2.908122	Rail	Route B	862.198864
99	haircare	SKU99	68.517833	17	627	9185.185629	Unknown	55	8	59	...	Chennai	29	921	2	38.072899	Fail	0.346027	Rail	Route B	210.743009

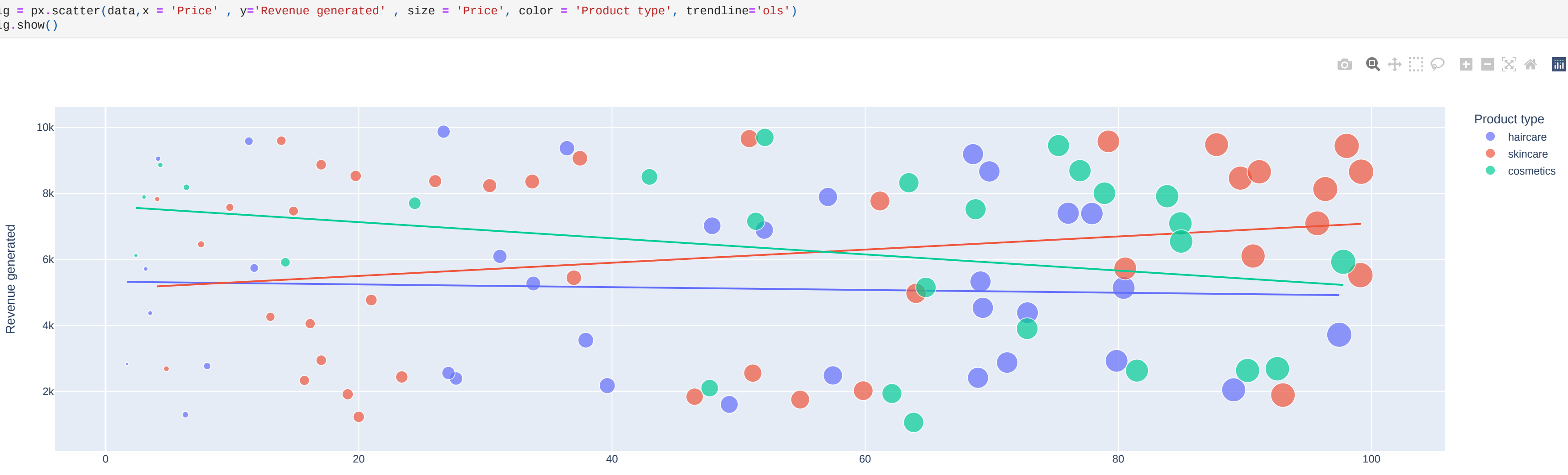
100 rows × 24 columns

In [3] :

```
data.describe()
```

	Price	Availability	Number of products sold	Revenue generated	Stock levels	Lead times	Order quantities	Shipping times	Shipping costs	Lead time	Production volumes	Manufacturing lead time	Manufacturing costs	Defect rates	Costs
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000
mean	49.462461	49.400000	460.960000	5776.048197	47.770000	15.960000	49.220000	5.750000	5.548149	17.080000	567.840000	14.770000	47.266693	2.277158	539.245792
std	31.189193	30.743317	303.780074	2732.841744	31.969372	8.785901	26.784429	2.724283	2.451376	8.846251	263.048681	8.91243	28.982841	1.461366	258.301086
min	1.699976	1.000000	6.000000	1061.613523	0.000000	1.000000	1.000000	1.000000	1.013487	1.000000	104.000000	1.000000	1.085069	0.018608	59.3016248
25%	19.597823	22.750000	184.250000	2812.847151	16.750000	6.000000	26.000000	3.750000	3.502428	10.000000	352.000000	7.000000	22.983259	1.009650	318.778455
50%	53.239631	43.500000	392.500000	6006.352023	47.500000	17.000000	52.000000	6.000000	5.320534	18.000000	568.500000	14.000000	45.905622	2.141963	520.439444
75%	77.196228	75.000000	704.250000	8253.970921	73.000000	24.000000	71.250000	8.000000	7.401695	25.000000	797.000000	23.000000	68.621026	3.563995	763.078231
max	99.171329	100.000000	996.000000	9866.465458	100.000000	30.000000	96.000000	10.000000	9.929816	30.000000	985.000000	30.000000	99.466109	4.939255	997.413450

```
In [4]: fig = px.scatter(data, x = 'Price', y = 'Revenue generated', size = 'Price', color = 'Product type', trendLine='ols')
fig.show()
```



Insights

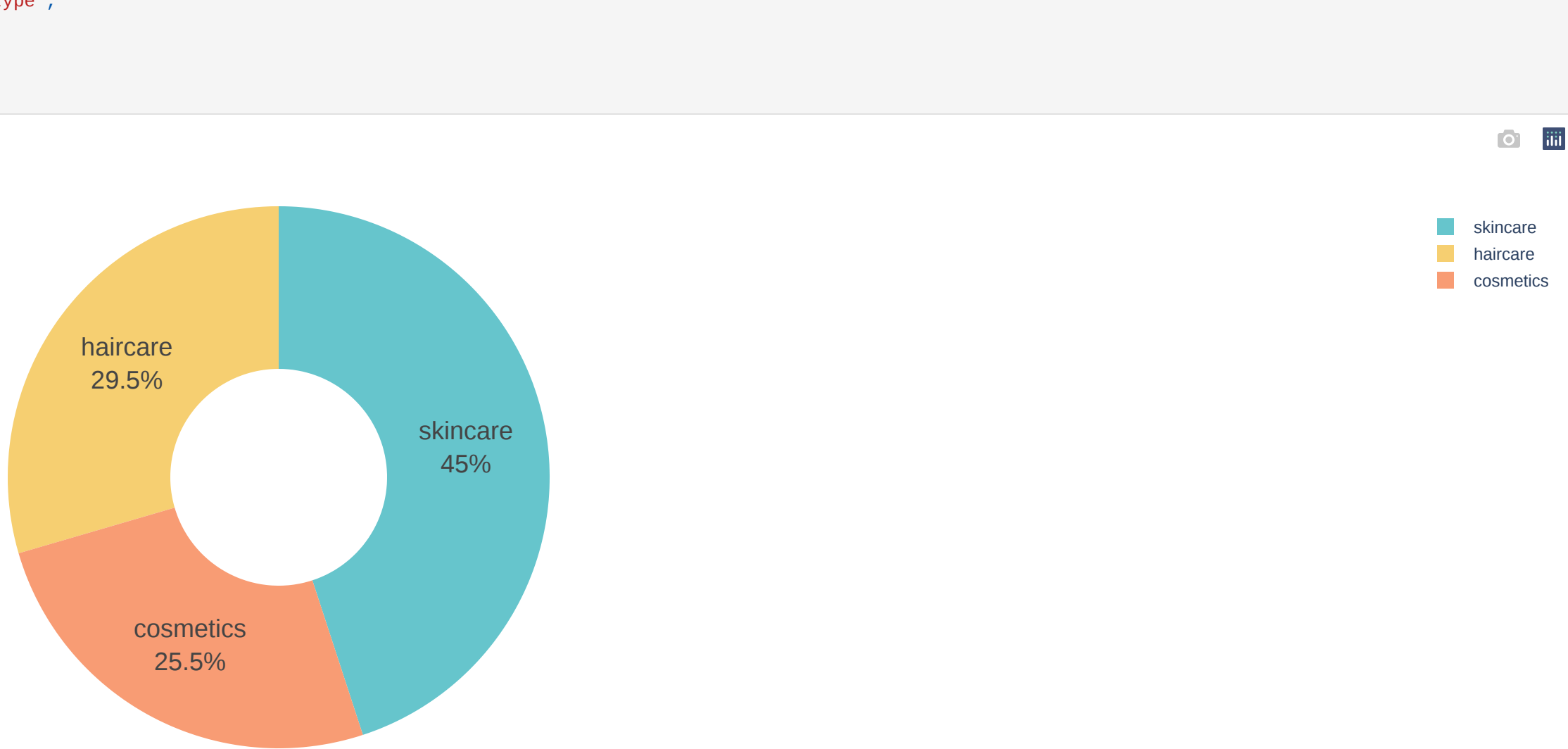
- The best resourse of revenue is Skincare Products
- The lowest resours of revenue is Cosmetics Products

In [] :

2 Analyzing the sales by product type

```
In [6]: product_sales = data.groupby('Product type')['Number of products sold'].sum().reset_index()
fig2 = px.pie(product_sales, values = 'Number of products sold', names = 'Product type',
             hole = 0.4, color_discrete_sequence = px.colors.qualitative.Pastel)
fig2.update_layout(title = 'Sale By Product Type')
fig2.update_traces(textinfo = 'percent+label', textfont_size = 18)
fig2.show()
```

Sale By Product Type



In [7] :

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times	Order quantities	...	Location	Lead time	Production volumes	Manufacturing lead time	Manufacturing costs	Inspection results	Defect rates	Transportation modes	Routes	Costs
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58	7	96	...	Mumbai	29	215	29	46.279879	Pending	0.226410	Road	Route B	187.752075
1	skincare	SKU1	14.843523	95	736	7460.900055	Female	53	30	37	...	Mumbai	23	517	30	33.616769	Pending	4.854068	Road	Route B	503.065579
2	haircare	SKU2	11.319683	34	8	9577.749626	Unknown	1	10	88	...	Mumbai	12	971	27	30.688019	Pending	4.580593	Air	Route C	141.920282
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23	13	59	...	Kolkata	24	937	18	35.624741	Fail	4.746649	Rail	Route A	254.776159
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5	3	56	...	Delhi	5	414	3	92.065161	Fail	3.145590	Air	Route A	923.440632

5 rows × 24 columns

Q3 Revenue generated by Carrier Shipping

```
In [22]: revenue = data.groupby('Shipping carriers')['Revenue generated'].sum().reset_index()
revenue
```

	Shipping carriers	Revenue generated
0	Carrier A	142629.994507
1	Carrier B	250094.646988
2	Carrier C	184880.177143

```
In [26]: fig3 = go.Figure(go.Bar(x = revenue['Shipping carriers'],
                             y = revenue['Revenue generated']))
fig3.update_layout(title = 'Revenue generated by shipping carrier',
                  xaxis_title = 'Shipping Carriers', yaxis_title = 'Revenue generated')
fig3.show()
```

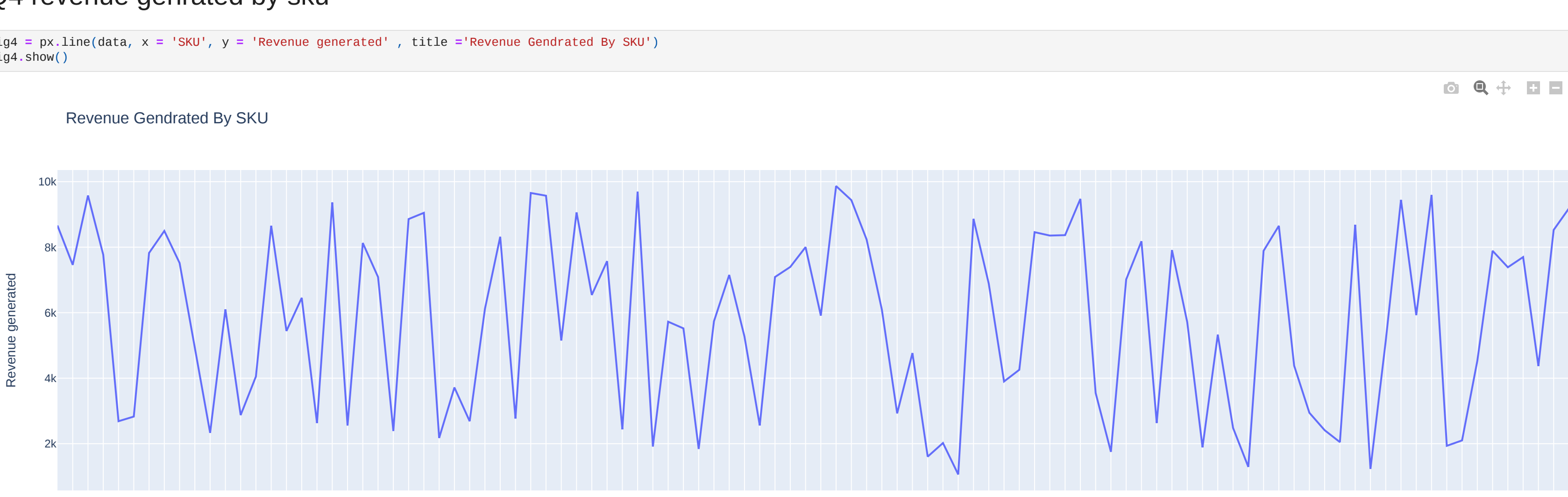
Revenue generated by shipping carrier



Q4 revenue genrated by sku

```
In [28]: fig4 = px.line(data, x = 'SKU', y = 'Revenue generated', title = 'Revenue Genrated By SKU')
fig4.show()
```

Revenue Genrated By SKU



In [29] :

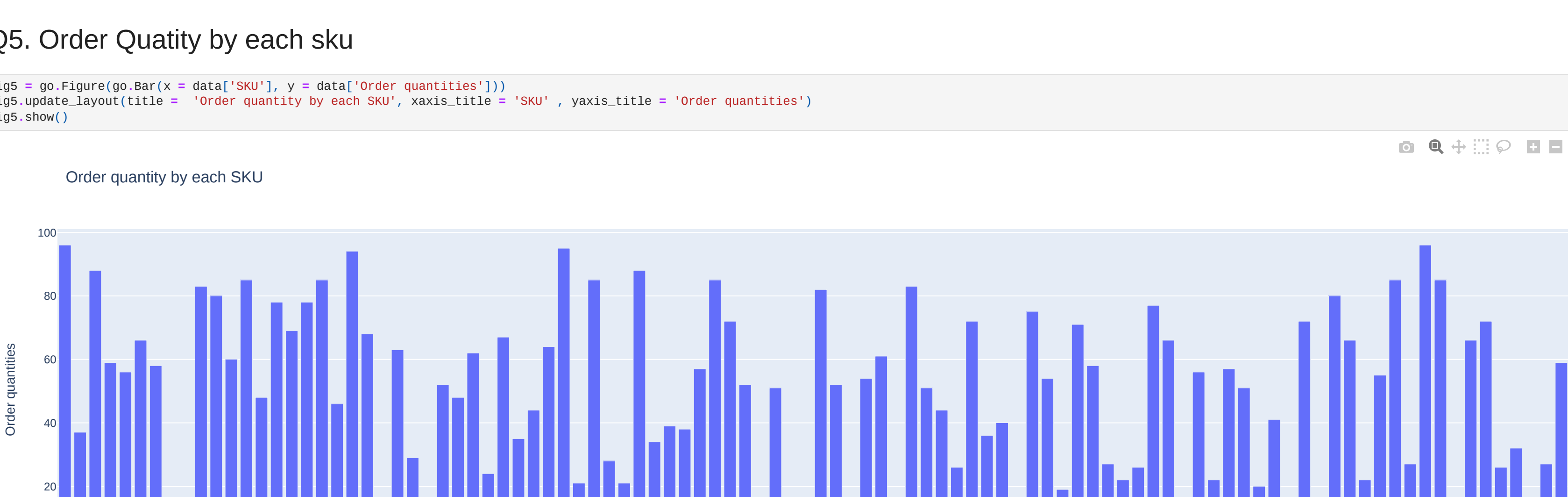
	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times	Order quantities	...	Location	Lead time	Production volumes	Manufacturing lead time	Manufacturing costs	Inspection results	Defect rates	Transportation modes	Routes	Costs
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58	7	96	...	Mumbai	29	215	29	46.279879	Pending	0.226410	Road	Route B	187.752075
1	skincare	SKU1	14.843523	95	736	7460.900055	Female	53	30	37	...	Mumbai	23	517	30	33.616769	Pending	4.854068	Road	Route B	503.065579
2	haircare	SKU2	11.319683	34	8	9577.749626	Unknown	1	10	88	...	Mumbai	12	971	27	30.688019	Pending	4.580593	Air	Route C	141.920282
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23	13	59	...	Kolkata	24	937	18	35.624741	Fail	4.746649	Rail	Route A	254.776159
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5	3	56	...	Delhi	5	414	3	92.065161	Fail	3.145590	Air	Route A	923.440632

5 rows × 24 columns

Q5. Order Quaty by each sku

```
In [36]: fig5 = go.Figure(go.Bar(x = data['SKU'], y = data['Order quantities']))
fig5.update_layout(title = 'Order quantity by each SKU', xaxis_title = 'SKU', yaxis_title = 'Order quantities')
fig5.show()
```

Order quantity by each SKU



In [37] :

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times	Order quantities	...	Location	Lead time	Production volumes	Manufacturing lead time	Manufacturing costs	Inspection results	Defect rates	Transportation modes	Routes	Costs
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58	7	96	...	Mumbai	29	215	29	46.279879	Pending	0.226410	Road	Route B	187.752075
1	skincare	SKU1	14.843523	95	736	7460.900055	Female	53	30	37	...	Mumbai	23	517	30	33.616769	Pending	4.854068	Road	Route B	503.065579
2	haircare	SKU2	11.319683	34	8	9577.749626	Unknown	1	10	88	...	Mumbai	12	971	27	30.688019	Pending	4.580593	Air	Route C	141.920282
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23	13	59	...	Kolkata	24	937	18	35.624741	Fail	4.746649	Rail	Route A	254.776159
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5	3	56	...	Delhi	5	414	3	92.065161	Fail	3.145590	Air	Route A	923.440632

5 rows × 24 columns

Q6. Shipping coast by carriers

```
In [44]: cost = data.groupby('Shipping carriers')['Shipping costs'].sum().reset_index()
cost
```

	Shipping carriers	Shipping costs
0	Carrier A	155.537831
1	Carrier B	236.897620
2	Carrier C	162.379457

```
In [47]: fig7 = px.bar(cost, x = 'Shipping carriers', y = 'Shipping costs', title = 'Shipping coast by carrier')
fig7.show()
```

Shipping coast by carrier



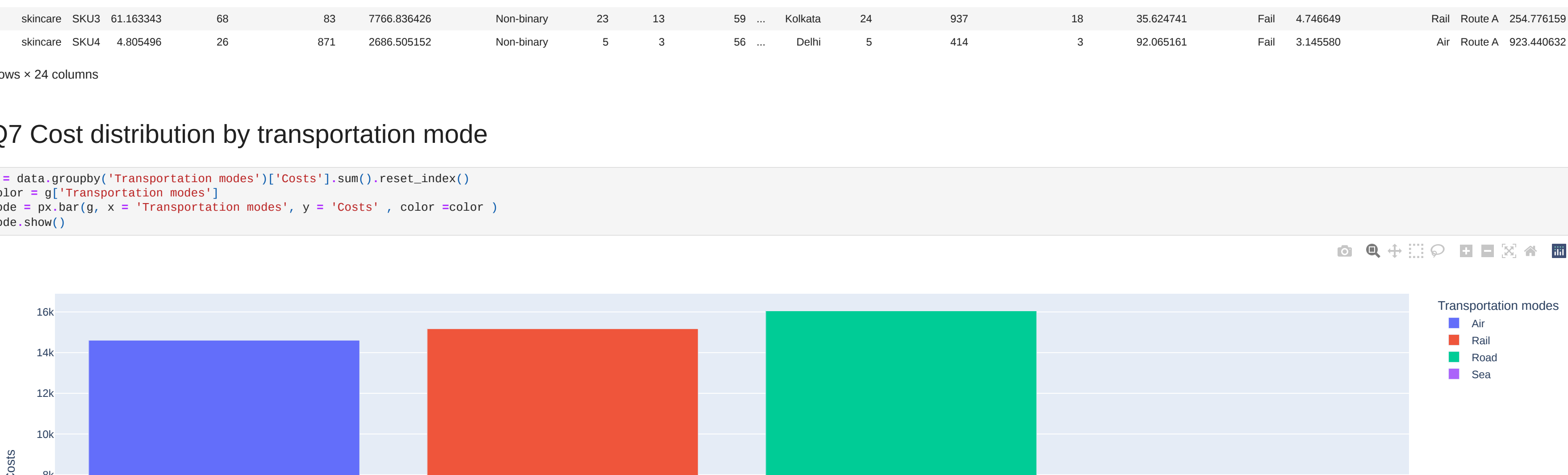
In [48] :

	Product type	SKU	Price	Availability	Number of products sold	Revenue generated	Customer demographics	Stock levels	Lead times	Order quantities	...	Location	Lead time	Production volumes	Manufacturing lead time	Manufacturing costs	Inspection results	Defect rates	Transportation modes	Routes	Costs
0	haircare	SKU0	69.808006	55	802	8661.996792	Non-binary	58	7	96	...	Mumbai	29	215	29	46.279879	Pending	0.226410	Road	Route B	187.752075
1	skincare	SKU1	14.843523	95	736	7460.900055	Female	53	30	37	...	Mumbai	23	517	30	33.616769	Pending	4.854068	Road	Route B	503.065579
2	haircare	SKU2	11.319683	34	8	9577.749626	Unknown	1	10	88	...	Mumbai	12	971	27	30.688019	Pending	4.580593	Air	Route C	141.920282
3	skincare	SKU3	61.163343	68	83	7766.836426	Non-binary	23	13	59	...	Kolkata	24	937	18	35.624741	Fail	4.746649	Rail	Route A	254.776159
4	skincare	SKU4	4.805496	26	871	2686.505152	Non-binary	5	3	56	...	Delhi	5	414	3	92.065161	Fail	3.145590	Air	Route A	923.440632

5 rows × 24 columns

Q7 Cost distribution by transportation mode

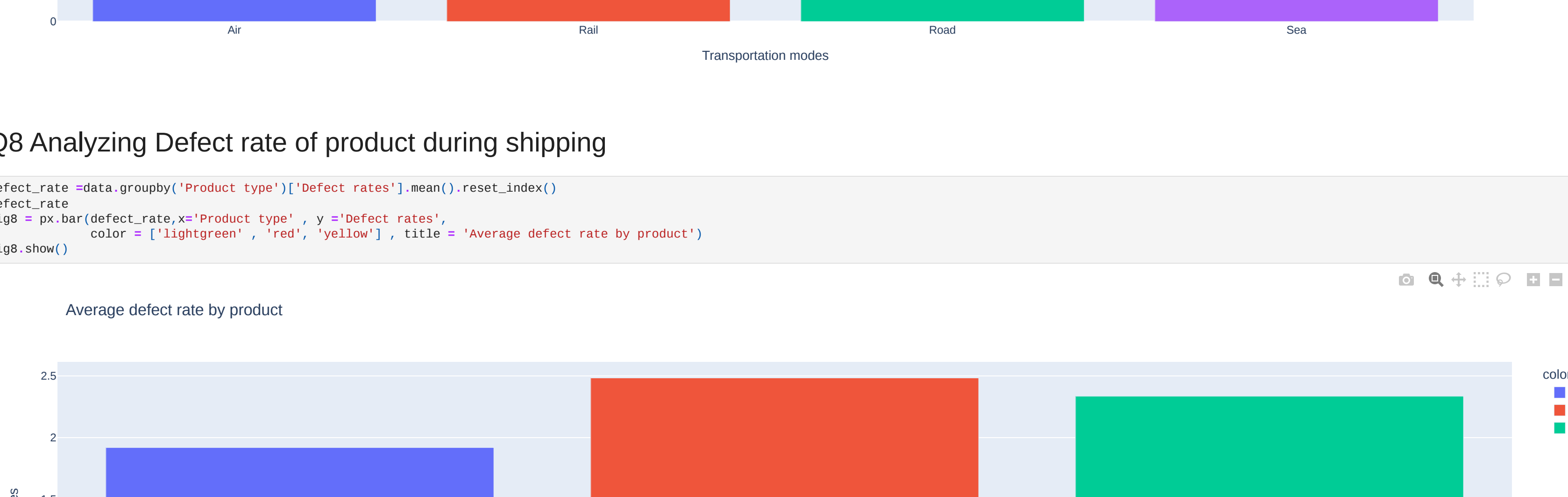
```
In [58]: g = data.groupby('Transportation modes')['Costs'].sum().reset_index()
color = g['Transportation modes'].color
mode = px.bar(g, x = 'Transportation modes', y = 'Costs', color = color)
mode.show()
```



Q8 Analyzing Defect rate of product during shipping

```
In [68]: defect_rate = data.groupby('Product type')['Defect rates'].mean().reset_index()
defect_rate
fig8 = px.bar(defect_rate, x = 'Product type', y = 'Defect rates',
              color = ['lightgreen', 'red', 'yellow'], title = 'Average defect rate by product')
fig8.show()
```

Average defect rate by product



In [] :

In [] :

In [] :

In [] :