In [1]:

```
# Importing all necessary python libraries for this project and the Data Set
import pandas as pd
import plotly.express as px
import plotly.io as pio
import plotly.graph_objects as go
pio.templates.default = "plotly_dark"
Fashion_Beauty_Data=pd.read_csv(r'C:\Users\OKONKWO HENRY\Downloads\supply_chain_data.csv')
print(Fashion_Beauty_Data.head())
```

```
  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]
```

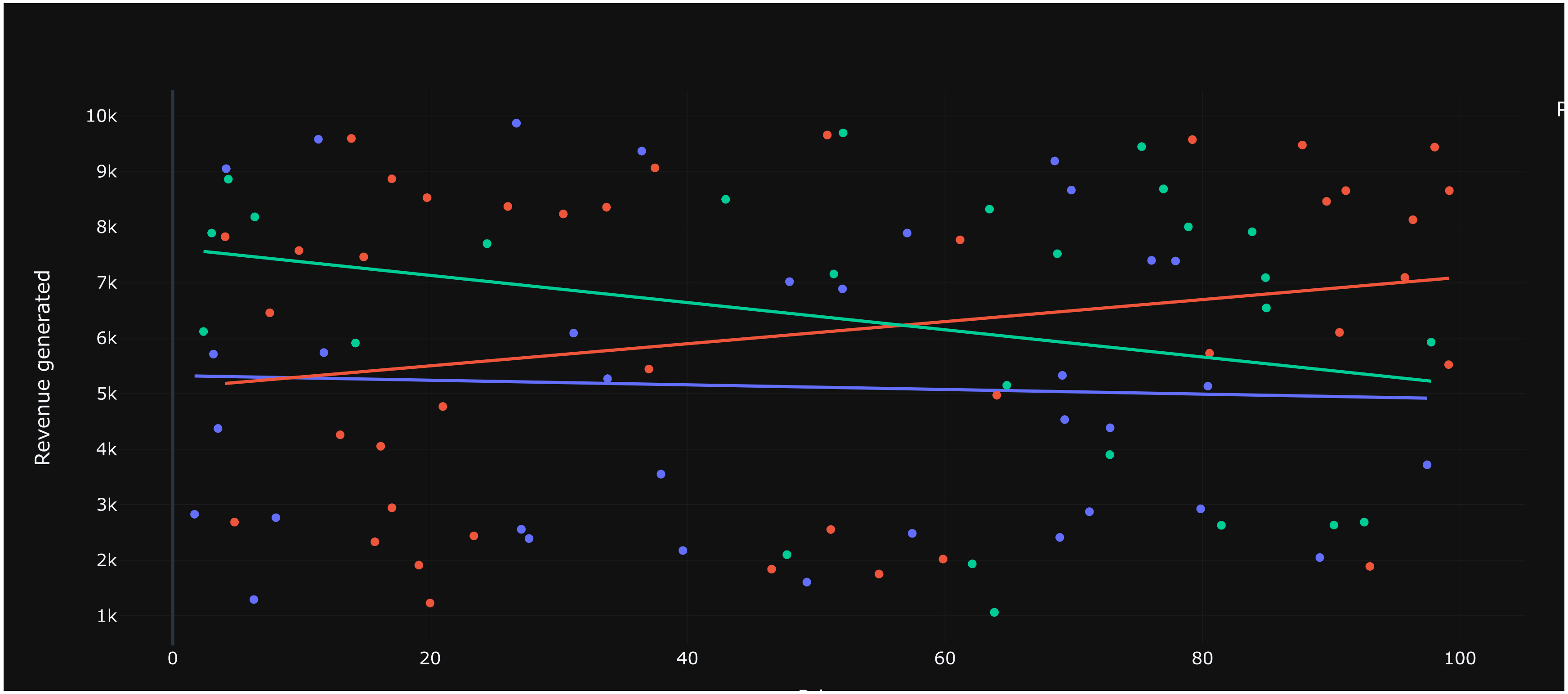
In [2]:

```
#Let’s have a look at the descriptive statistics of the dataset:
print(Fashion_Beauty_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%    51.239830     43.500000               392.500000        6006.352023   
75%    77.198228     75.000000               704.250000        8253.976920   
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 [3]:

```python
#Begin analysis the Supply Chain
#  The relationship between the price of the products and the revenue generated by them
Price_Revenue=px.scatter(Fashion_Beauty_Data, x='Price', y='Revenue generated', color='Product type',
                hover_data=['Number of products sold'], trendline="ols")
Price_Revenue.show()
```

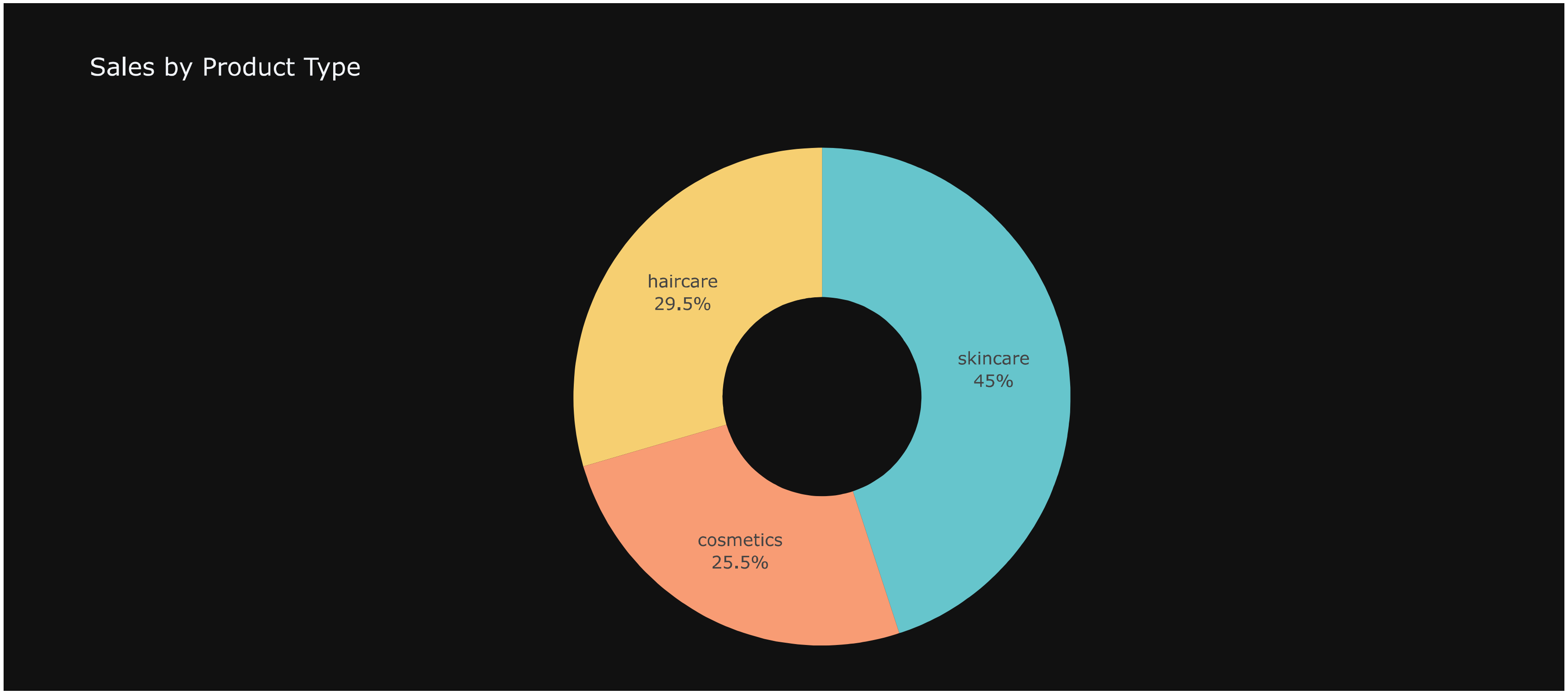


In [4]:

```
# DATA INSIGHT 1
# Thus, the company derives more revenue from skincare products, and the higher the price of skincare products, the more revenue they generate.
```

In [5]:

```
# Analysis of Sales by product type
sales_product = Fashion_Beauty_Data.groupby('Product type')['Number of products sold'].sum().reset_index()
pie_chart = px.pie(sales_product, values='Number of products sold', names='Product type',
                   title='Sales by Product Type',
                   hover_data=['Number of products sold'],
                   hole=0.4,
                   color_discrete_sequence=px.colors.qualitative.Pastel)

pie_chart.update_traces(textposition='inside', textinfo='percent+label')
pie_chart.show()
```



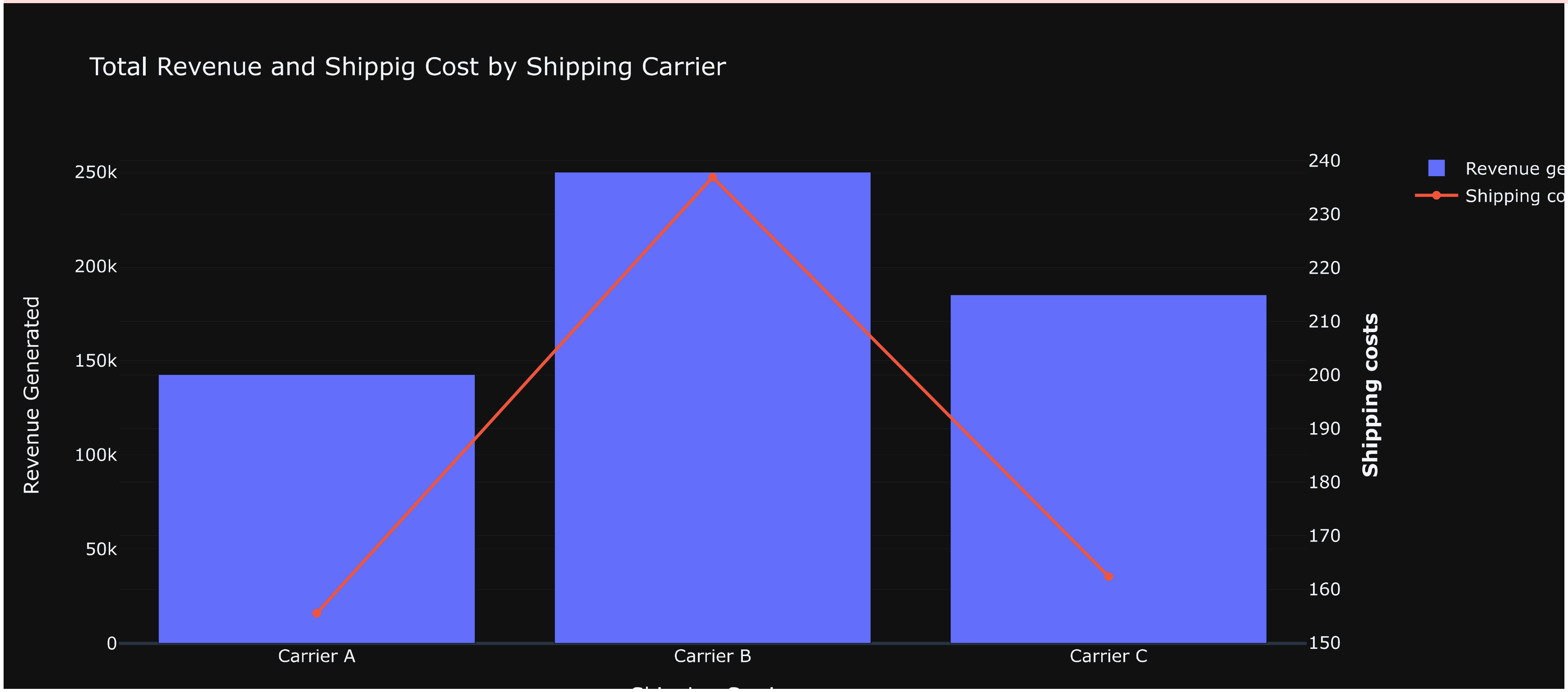
In [6]:

```
# DATA INSIGHT 2
# It can be Shown that 45% of the business comes from skincare products, 29.5% from haircare, and 25.5% from cosmetics.
```

In [7]:

```
# Analysis of Shipping Carriers to Revenue and Cost
from plotly.subplots import make_subplots
total_revenue = Fashion_Beauty_Data.groupby('Shipping carriers')['Revenue generated','Shipping costs'
].sum().reset_index()
fig = go.Figure()
# use specs parameter in make_subplots function
# to create secondary y-axis
fig = make_subplots(specs=[[{"secondary_y": True}]])
fig.add_trace(go.Bar(x=total_revenue['Shipping carriers'],
                     y=total_revenue['Revenue generated'],name='Revenue generated Values'),secondary_y=False)
# Use add_trace function and specify secondary_y axes = True.
fig.add_trace(
    go.Scatter(x=total_revenue['Shipping carriers'],y= total_revenue['Shipping costs'], name="Shipping costs values"),
    secondary_y=True,)
fig.update_layout(title='Total Revenue and Shippig Cost by Shipping Carrier',
                  xaxis_title='Shipping Carrier',)
# Naming y-axes
fig.update_yaxes(title_text='Revenue Generated', secondary_y=False)
fig.update_yaxes(title_text="<b>Shipping costs</b>", secondary_y=True)
fig.show()
```

```
C:\Users\OKONKWO HENRY\AppData\Local\Temp\ipykernel_20556\2994962219.py:3: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
```



```
In [8]: # DATA INSIGHT 3
        # The company is using three carriers for transportation, and Carrier B helps the company in generating more revenue
        # but with higher shipping Cost
```

```
In [9]: # Analysis of average lead time and Average Manufacturing Costs for all products of the company:
        avg_lead_time = Fashion_Beauty_Data.groupby('Product type')['Lead time'].mean().reset_index()
        avg_manufacturing_costs = Fashion_Beauty_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)
```

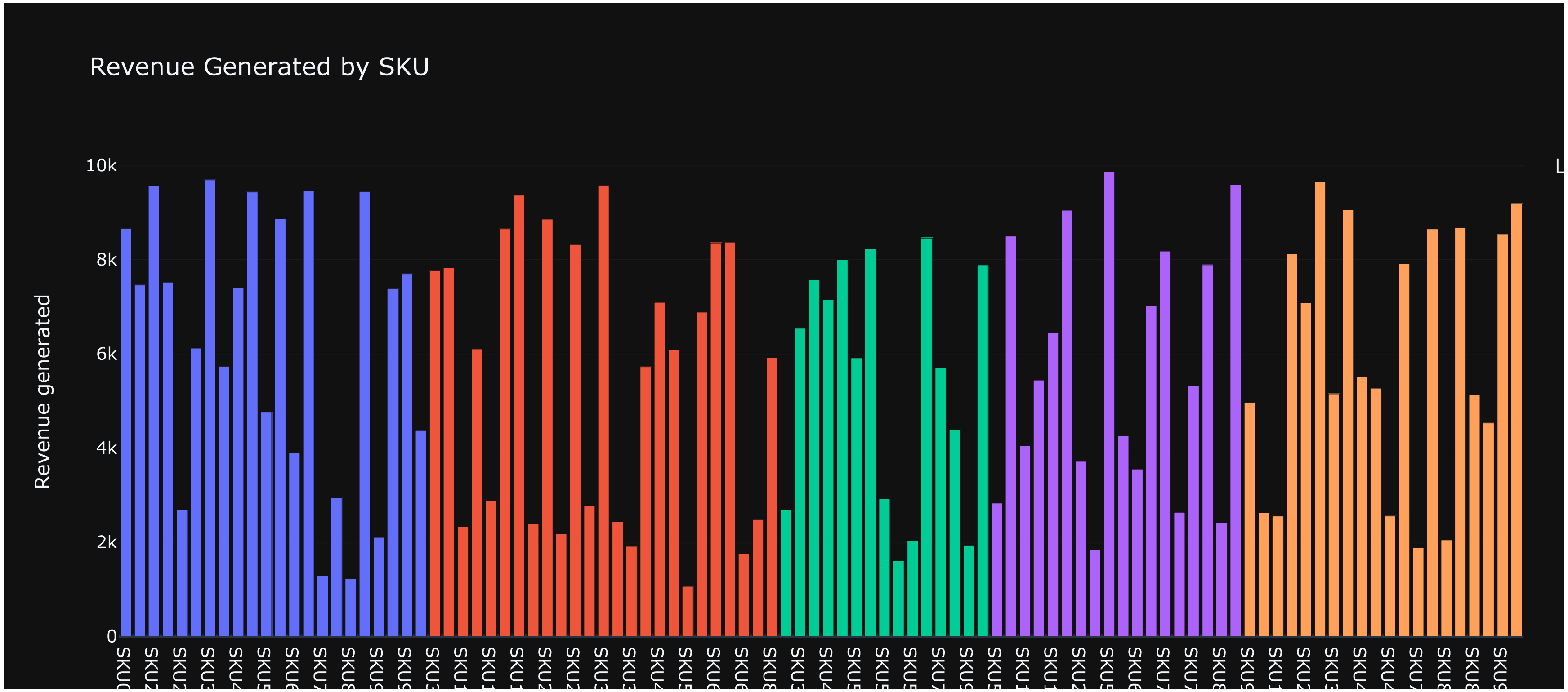
```
  Product type  Average Lead Time  Average Manufacturing Costs
0    cosmetics          13.538462                    43.052740
1     haircare          18.705882                    48.457993
2     skincare          18.000000                    48.993157
```

In [10]:

```
# DATA INSIGHT 4
# The Cosmetics Product has the Least average Lead time and average Lowest Maufacturing Cost
```

In [11]:

```
# Analysis of SKU(Store Keeping Unit) by Revenue Generated with Location
revenue_chart = px.bar(Fashion_Beauty_Data, x='SKU',
                        y='Revenue generated', color ="Location",
                        title='Revenue Generated by SKU')
revenue_chart.show()
```

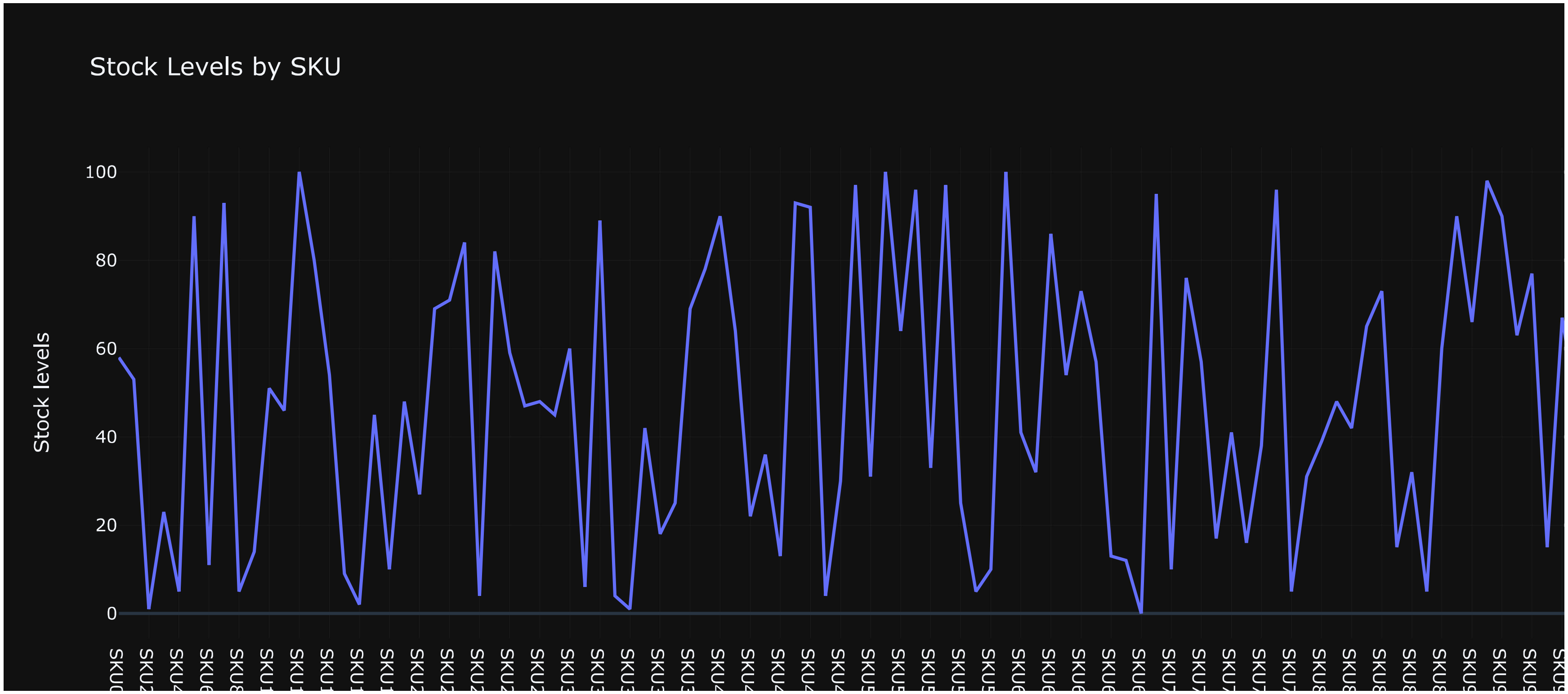


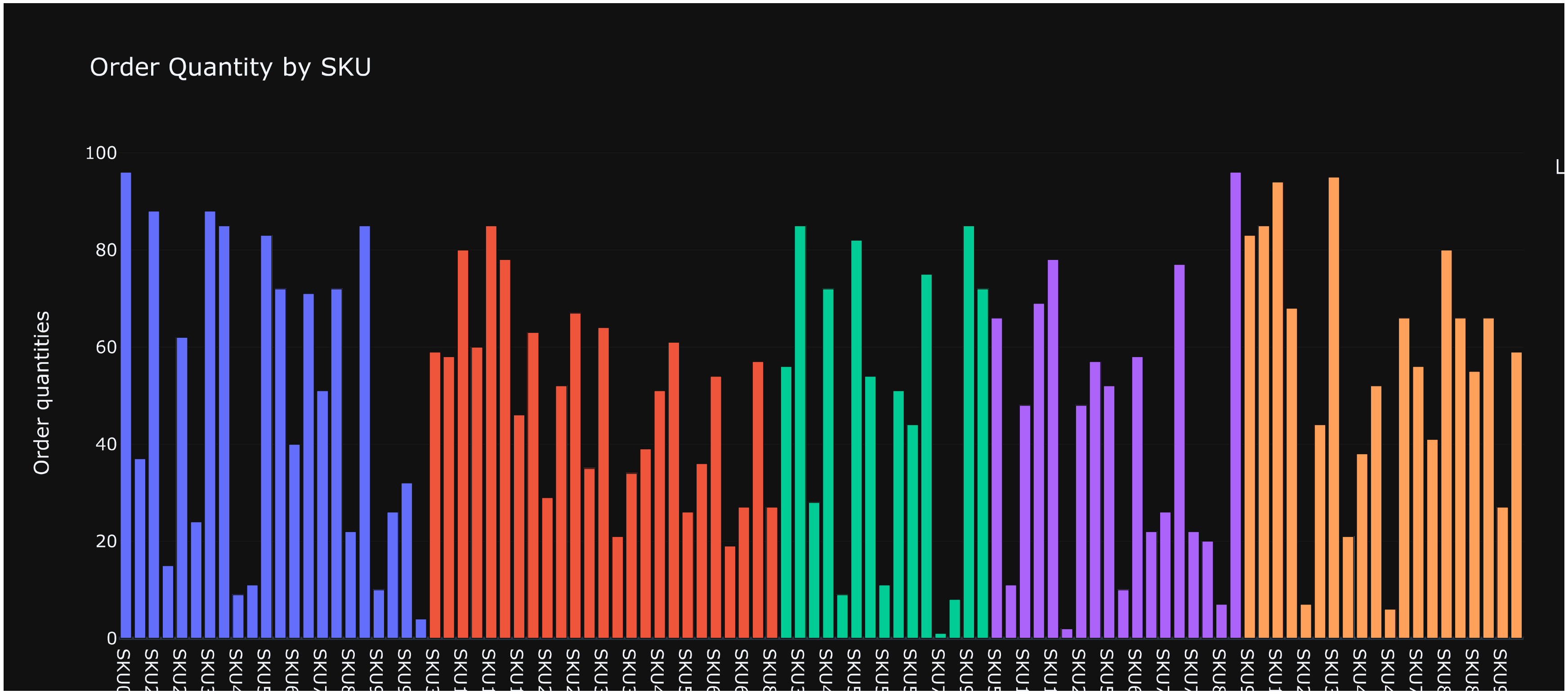
In [12]:

```
# DATA INSIGHT 5
    # The SKU with the highest Reveue generated with Location at Bangalore
```

In [13]:

```
# Analysis of SKU(Store Keeping Unit) by Stock Levels and Ordered Quantities
# Stock levels refer to the number of products a store or business has in its inventory.
stock_chart = px.line(Fashion_Beauty_Data, x='SKU',
                      y='Stock levels',
                      title='Stock Levels by SKU')
stock_chart.show()
order_quantity_chart = px.bar(Fashion_Beauty_Data, x='SKU',
                              y='Order quantities',color ="Location",
                              title='Order Quantity by SKU')
order_quantity_chart.show()
```


Stock Levels by SKU
Stock levels
100
80
60
40
20
0
SKU0
SKU2
SKU4
SKU6
SKU8
SKU1
SKU1
SKU1
SKU1
SKU1
SKU2
SKU2
SKU2
SKU2
SKU2
SKU3
SKU3
SKU3
SKU3
SKU3
SKU4
SKU4
SKU4
SKU4
SKU4
SKU5
SKU5
SKU5
SKU5
SKU5
SKU6
SKU6
SKU6
SKU6
SKU6
SKU7
SKU7
SKU7
SKU7
SKU7
SKU8
SKU8
SKU8
SKU8
SKU8
SKU9
SKU9
SKU9
SKU9

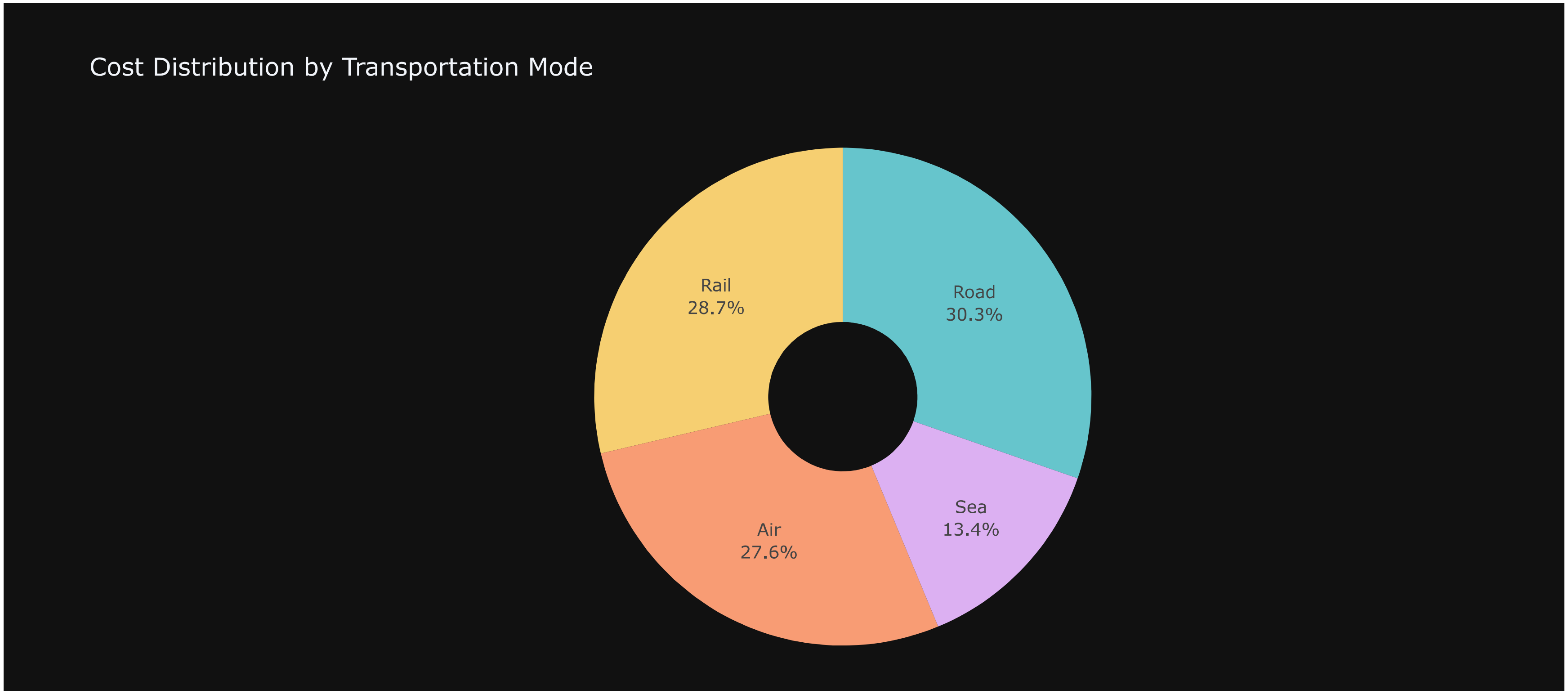


In [14]:

```
# DATA INSIGHT 6
# The SKU with the Highest Stock Level are SKU 51 and SKU 59 with 100 while
# The SKU with the Highest Order Quantities is SKU 0 and SKU 90
```

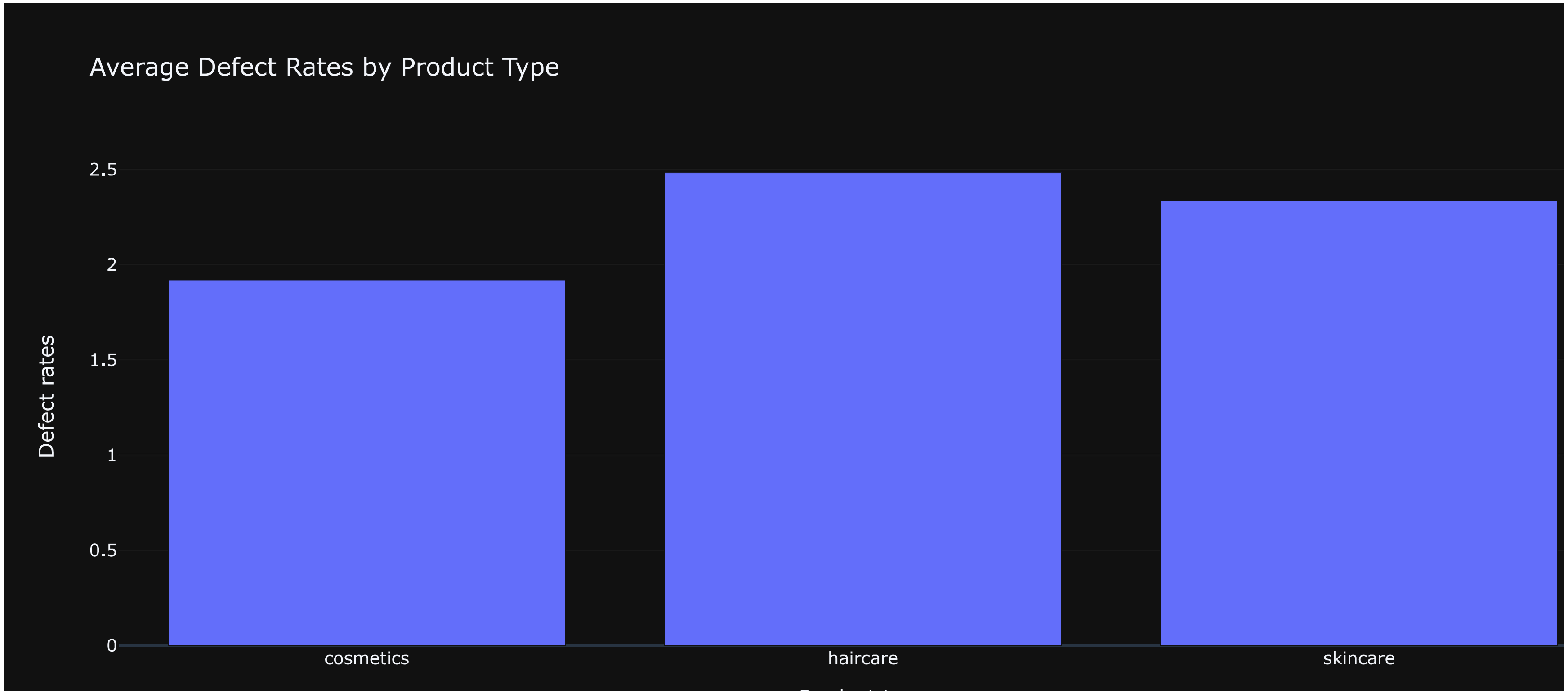
In [15]:

```
# Cost Analysis Distribution by mode of Transportation
transportation_chart = px.pie(Fashion_Beauty_Data, values='Costs', names='Transportation modes',
                              title='Cost Distribution by Transportation Mode',
                              hole=0.3,
                              color_discrete_sequence=px.colors.qualitative.Pastel)
transportation_chart.update_traces(textposition='inside', textinfo='percent+label')
transportation_chart.show()
```



In [16]:

```
# Analysis of Defect rate by Product Type
defect_rates_by_product = Fashion_Beauty_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()
```



In [17]:

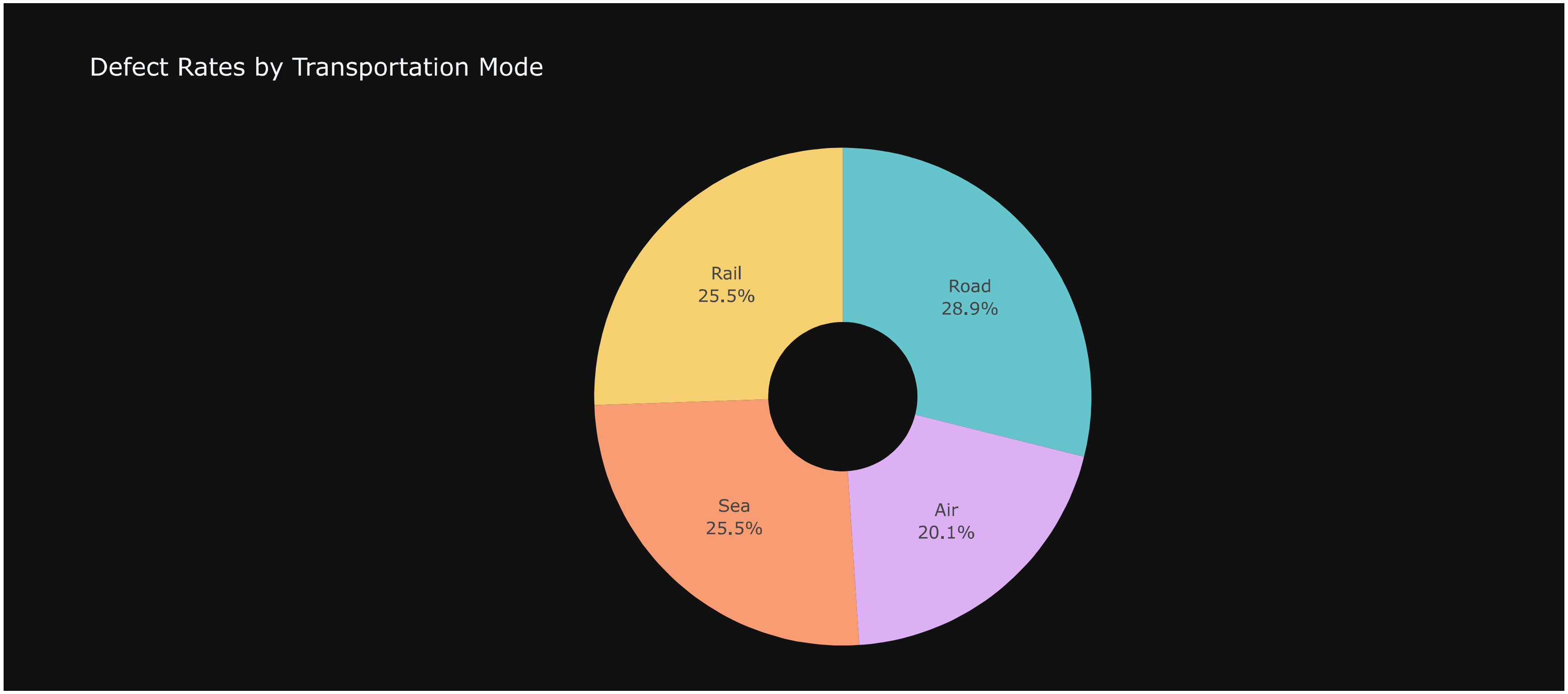
```
# DATA INSIGHT 7
# The company spends more on Road and Rail modes of transportation for the transportation of Goods
```

In [18]:

```
# DATA INSIGHT 8
# So the defect rate of haircare products is higher
```

In [19]:

```
# Analysis on defect rate by Transportation Mode
pivot_table = pd.pivot_table(Fashion_Beauty_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=0.3,
                              color_discrete_sequence=px.colors.qualitative.Pastel)
transportation_chart.update_traces(textposition='inside', textinfo='percent+label')
transportation_chart.show()
```



```
In [20]: # DATA INSIGHT 9
         # Road transportation results in a higher defect rate, and Air transportation has the lowest defect rate.
```