Python初級數據分析員證書

（六）數據分析 及可視化專案

13. 數據分析專案 - Demo 12 – Ecommerce Shipping Data Part 2

Review

- Statistics
- Hypothesis testing
- Algebra
- Linear regression
- Propositional logic
- Python
- R
- SQL

- Pandas, NumPy, SciPy
- Data Visualization, Matplotlib, Seaborn, Plotly
- Dashboard Visualization, Business Intelligence
- Storytelling

Data Recap

The data contains the following information:

- **ID**: ID Number of Customers.
- **Warehouse block**: The Company have big Warehouse which is divided in to block such as A,B,C,D,E.
- **Mode of shipment**: The Company Ships the products in multiple way such as Ship, Flight and Road.
- **Customer care calls**: The number of calls made from enquiry for enquiry of the shipment.
- **Customer rating**: The company has rated from every customer. 1 is the lowest (Worst), 5 is the highest (Best).
- **Cost of the product**: Cost of the Product in US Dollars.

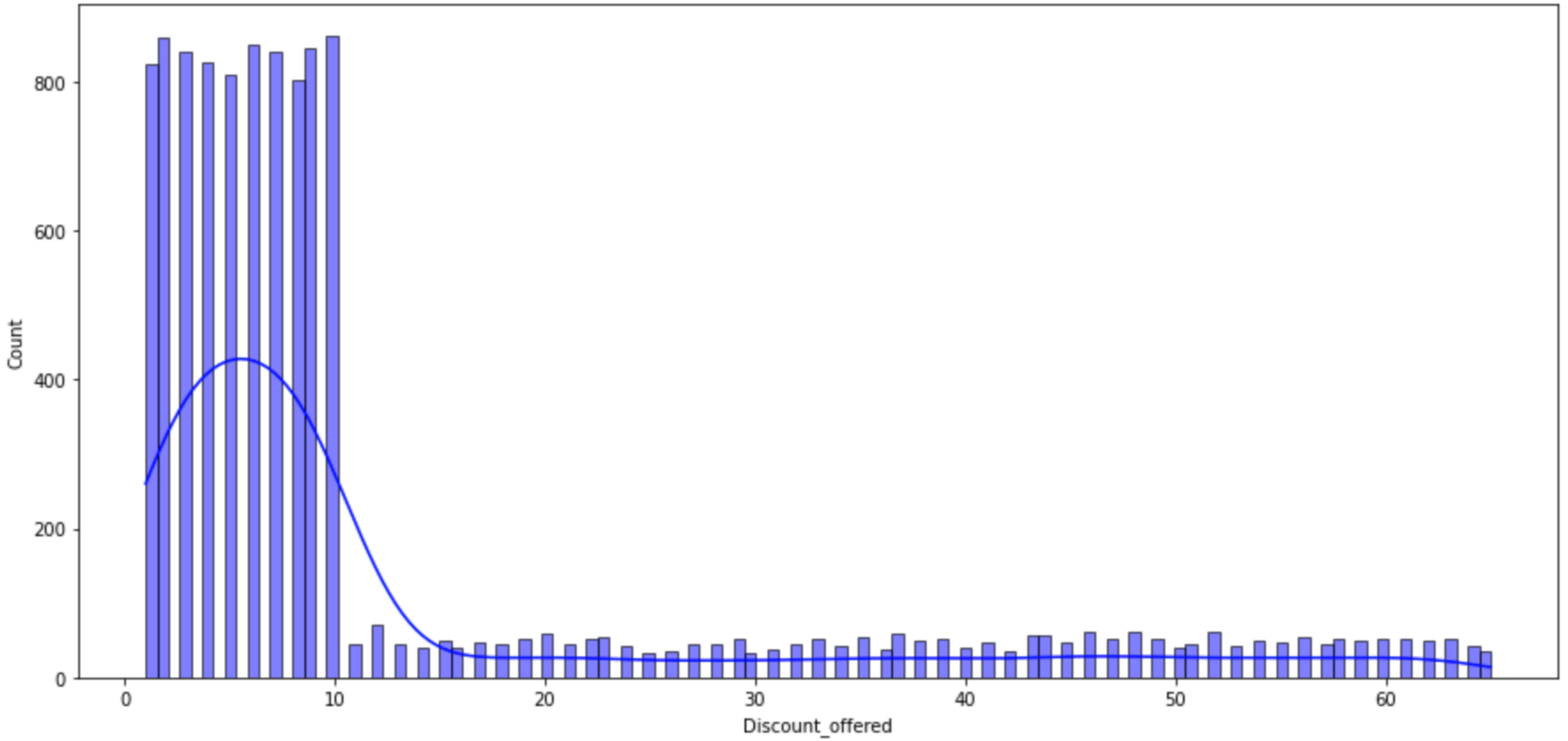
Data Recap

- **Prior purchases**: The Number of Prior Purchase.
- **Product importance**: The company has categorized the product in the various parameter such as low, medium, high.
- **Gender**: Male and Female.
- **Discount offered**: Discount offered on that specific product.
- **Weight in gms**: It is the weight in grams.
- **Reached on time**: It is the **target variable**, where **1** Indicates that the product has NOT reached on time and **0** indicates it has reached on time.

Continue Previous Chapter

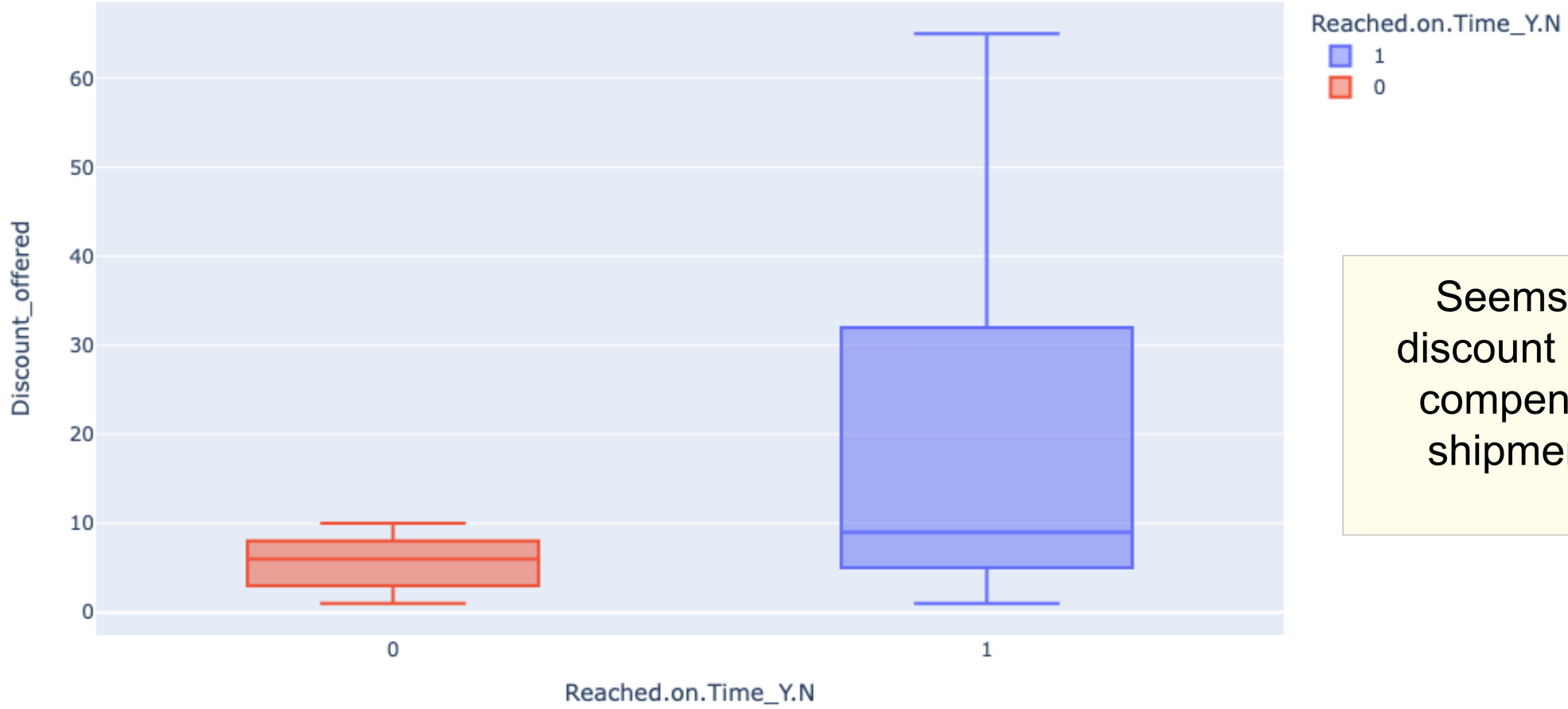
Discount offer distribution

```
1 plt.figure(figsize = (15, 7))
2 ax = sns.histplot(df['Discount_offered'], color = 'b', kde=True)
3 plt.show()
```



Discount offered vs Shipment on time

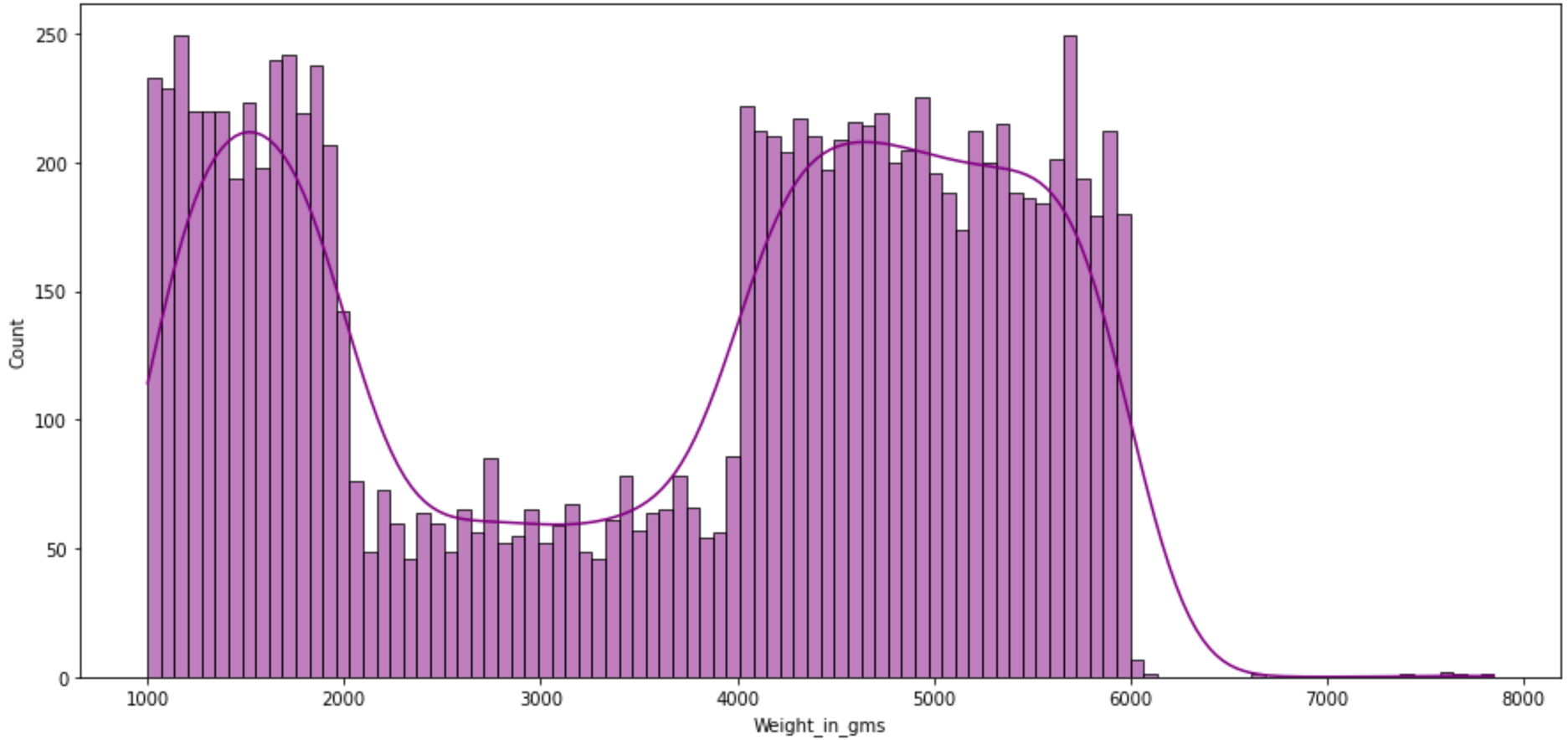
```
1  # 1 : NOT on time and 0: on time
2  px.box(data_frame = df, x = 'Reached.on.Time_Y.N', y = 'Discount_offered',
3         color = 'Reached.on.Time_Y.N')
```



Seems like the discount offered, is compensation of shipment delay.

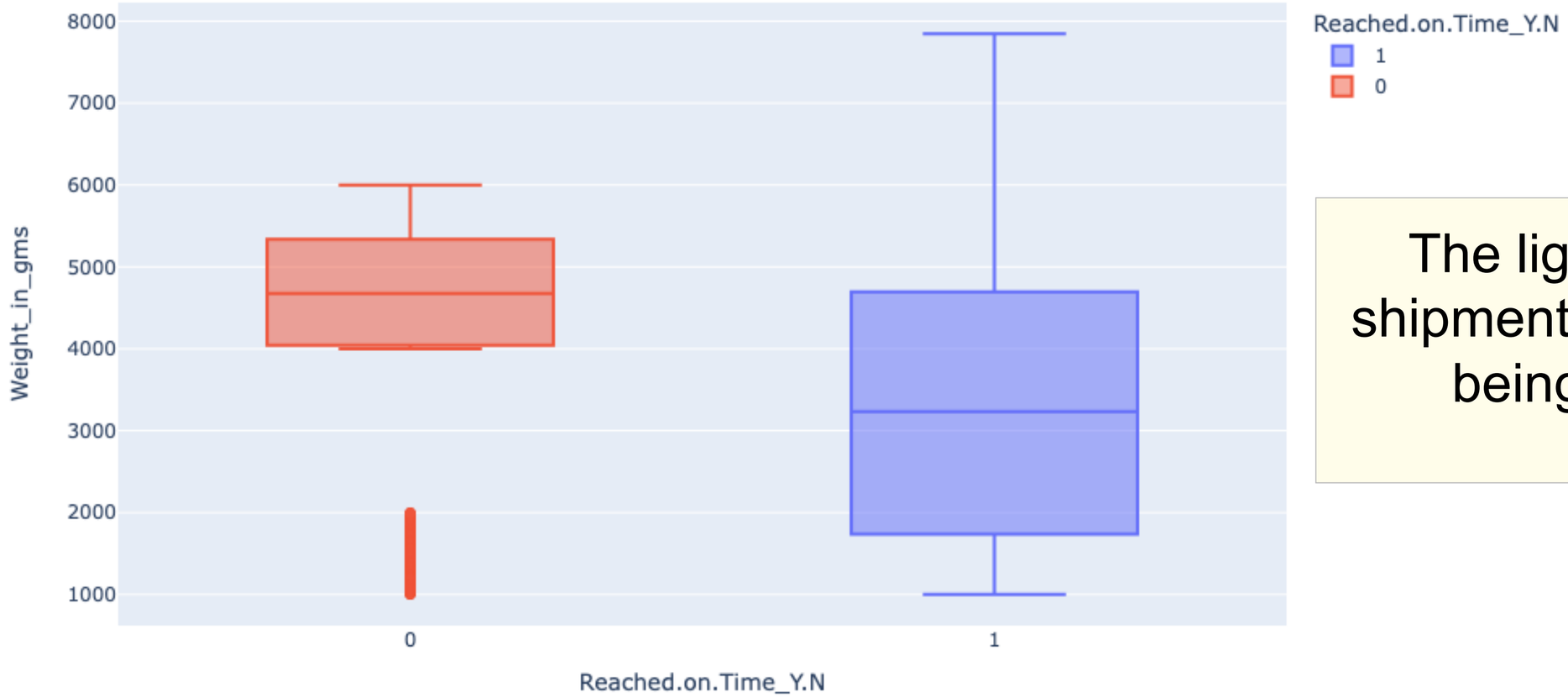
Weight in Grams

```
1 plt.figure(figsize = (15, 7))
2 ax = sns.histplot(df['Weight_in_gms'], bins = 100, color = 'purple', kde=True)
3 plt.show()
```



Weight in Grams vs Shipment on time

```
1 # 1 : NOT on time and 0: on time
2 px.box(data_frame = df, x = 'Reached.on.Time_Y.N', y = 'Weight_in_gms',
3         color = 'Reached.on.Time_Y.N', )
```

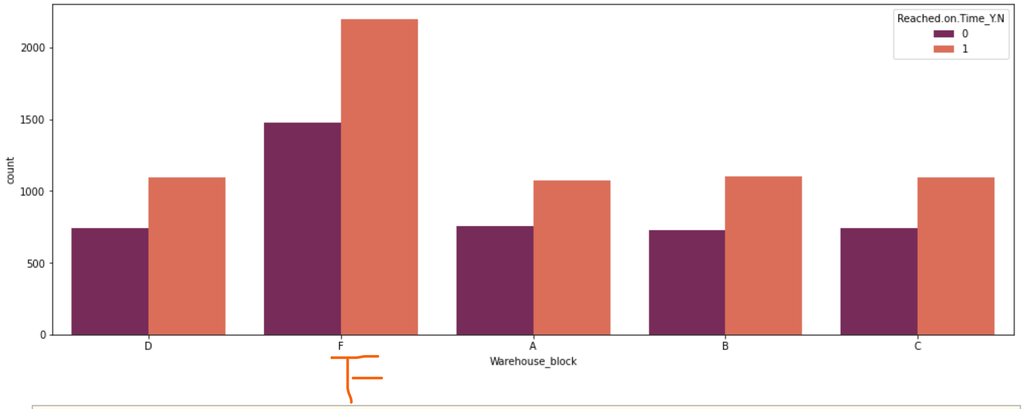


The light weight shipment usually get being delay!

Which warehouse contains most weights?

```
1 ware_block_weight = df.groupby(['Warehouse_block'])['Weight_in_gms'].sum().reset_index()
2 ware_block_weight
```

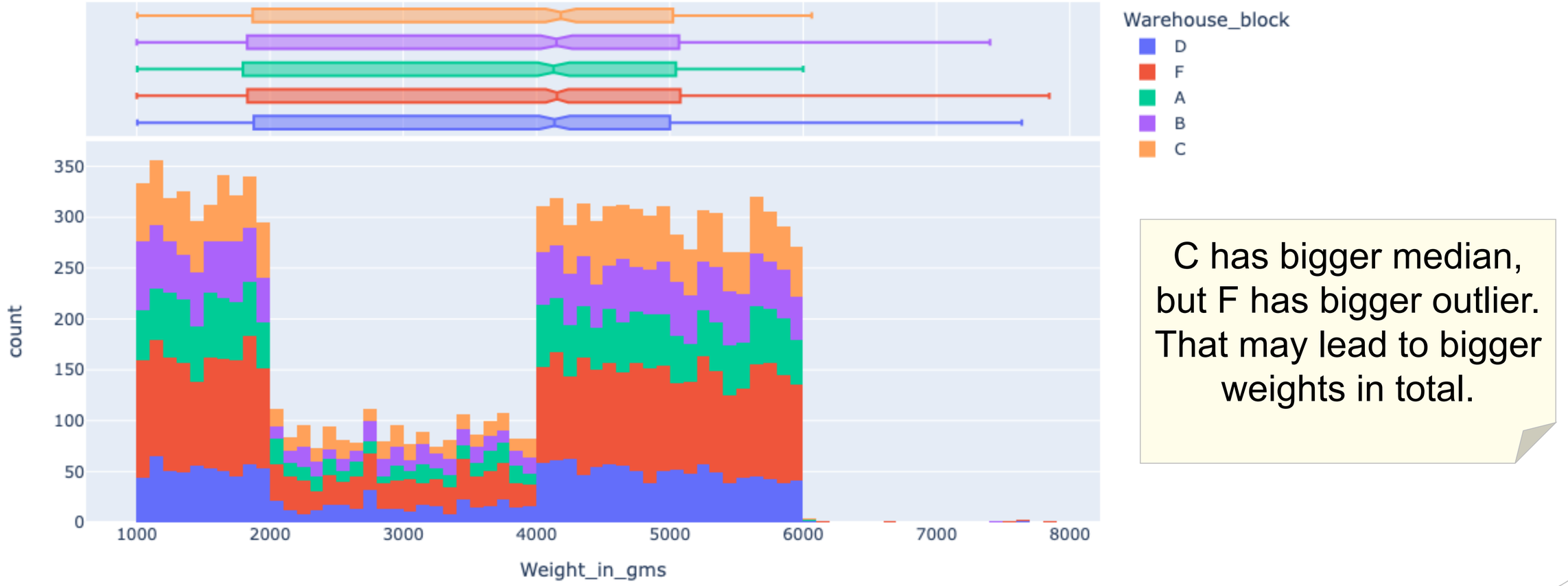
	Warehouse_block	Weight_in_gms
0	A	6627118
1	B	6664240
2	C	6674560
3	D	6655305
4	F	13349327



Refer to previous Warehouse vs Shipment on time plot, does it means F overloaded and made such delay?

Which warehouse contains most weights?

```
1  px.histogram(data_frame = df, x = 'Weight_in_gms', nbins = 100,
2               color = 'Warehouse_block', marginal = 'box')
```



C has bigger median, but F has bigger outlier. That may lead to bigger weights in total.

Mode of shipment vs Weight in grams

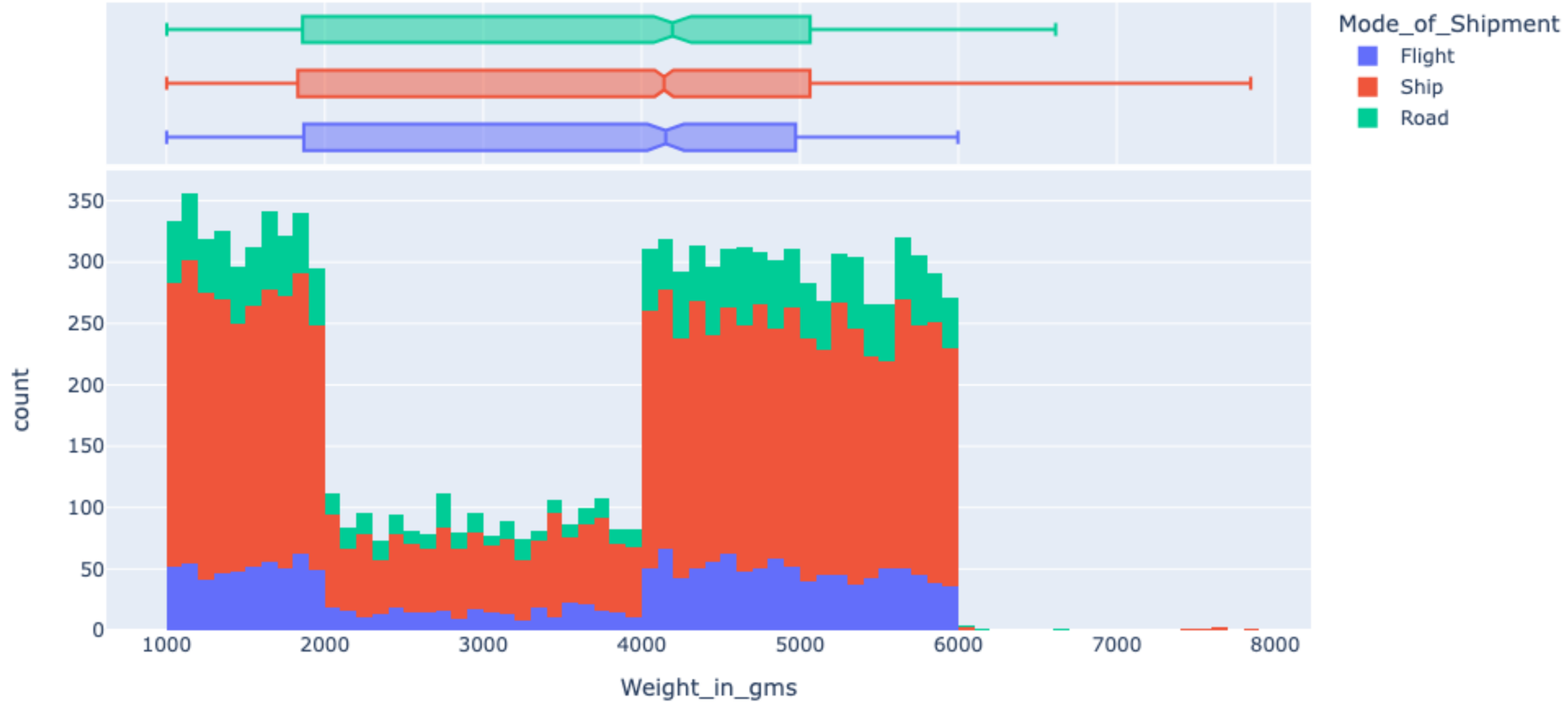
Most of the parcels are transferred by ships.

```
shipment_mode_weight = df.groupby(['Mode_of_Shipment'])['Weight_in_gms'].sum().reset_index()
shipment_mode_weight
```

	Mode_of_Shipment	Weight_in_gms
0	Flight	6449405
1	Road	6423209
2	Ship	27097936

Mode of shipment vs Weight in grams

```
1  px.histogram(data_frame = df, x = 'Weight_in_gms', nbins = 100,
2               color = 'Mode_of_Shipment', marginal = 'box')
```



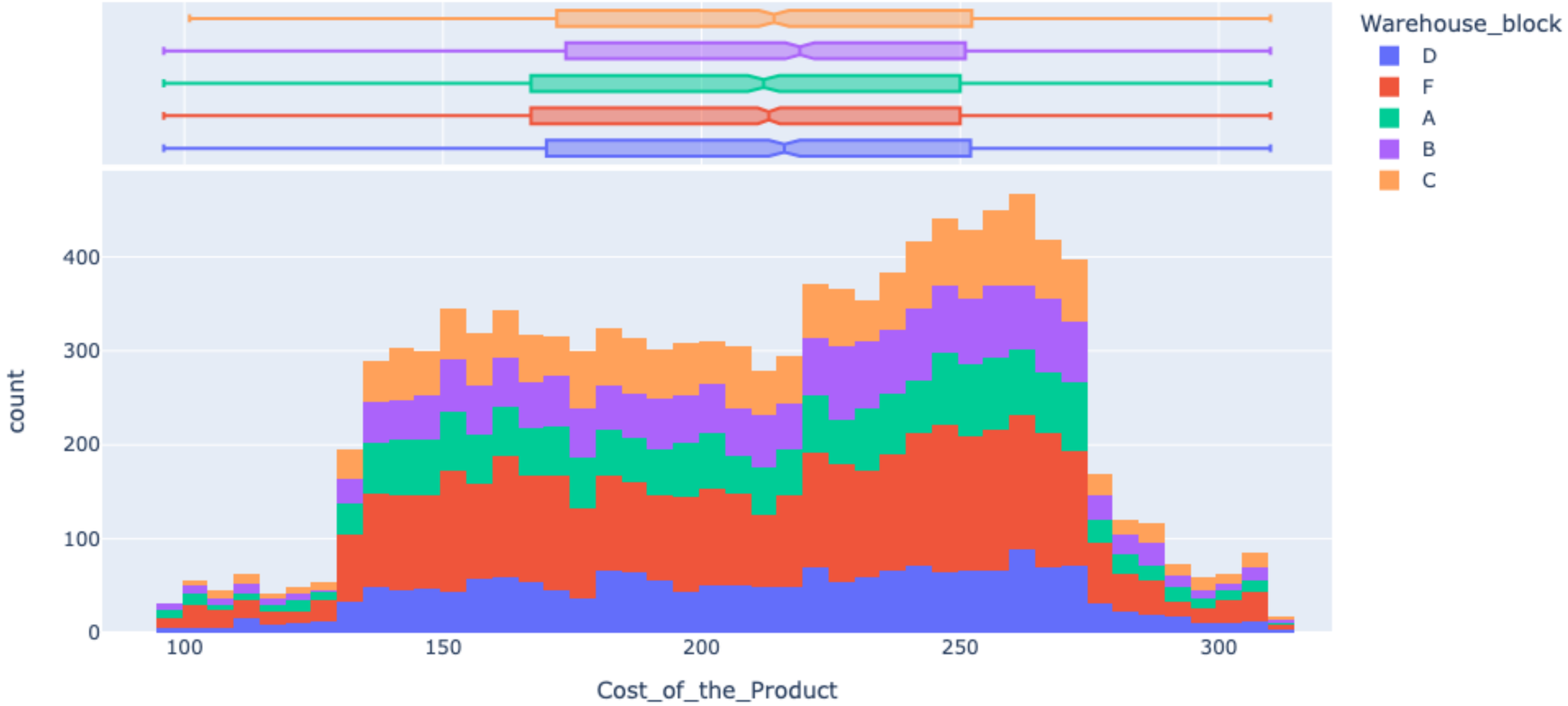
Warehouse vs Cost of product

```
warehouse_weight = df.groupby(['Warehouse_block'])['Cost_of_the_Product'].sum().reset_index()
warehouse_weight
```

	Warehouse_block	Cost_of_the_Product
0	A	382671
1	B	388888
2	C	387114
3	D	386805
4	F	766477

Warehouse vs Cost of product

```
px.histogram(data_frame = df, x = 'Cost_of_the_Product', nbins = 100,
             color = 'Warehouse_block', marginal = 'box')
```



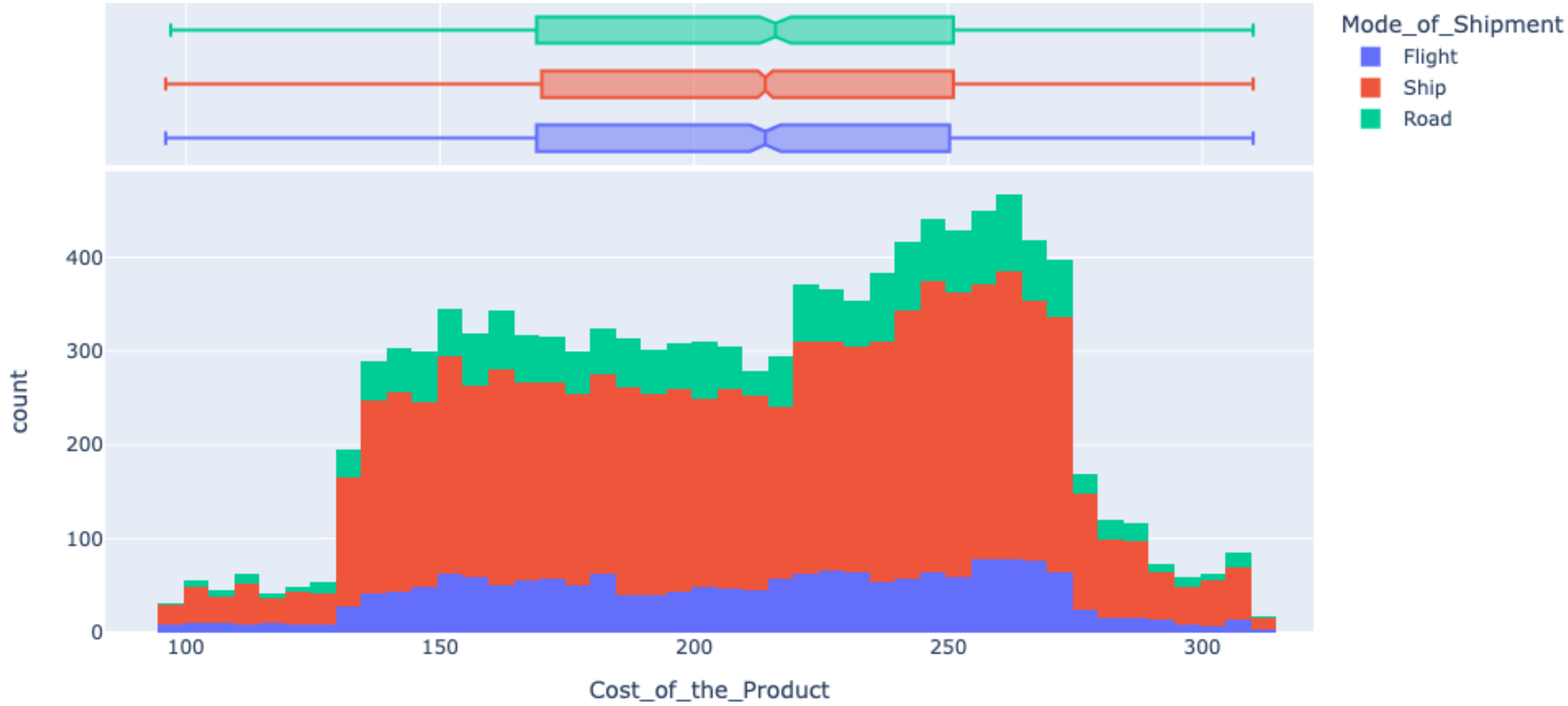
Cost of product vs Shipment mode

```
mode_shipment_cost = df.groupby(['Mode_of_Shipment'])['Cost_of_the_Product'].sum().reset_index()
mode_shipment_cost
```

	Mode_of_Shipment	Cost_of_the_Product
0	Flight	371938
1	Road	370437
2	Ship	1569580

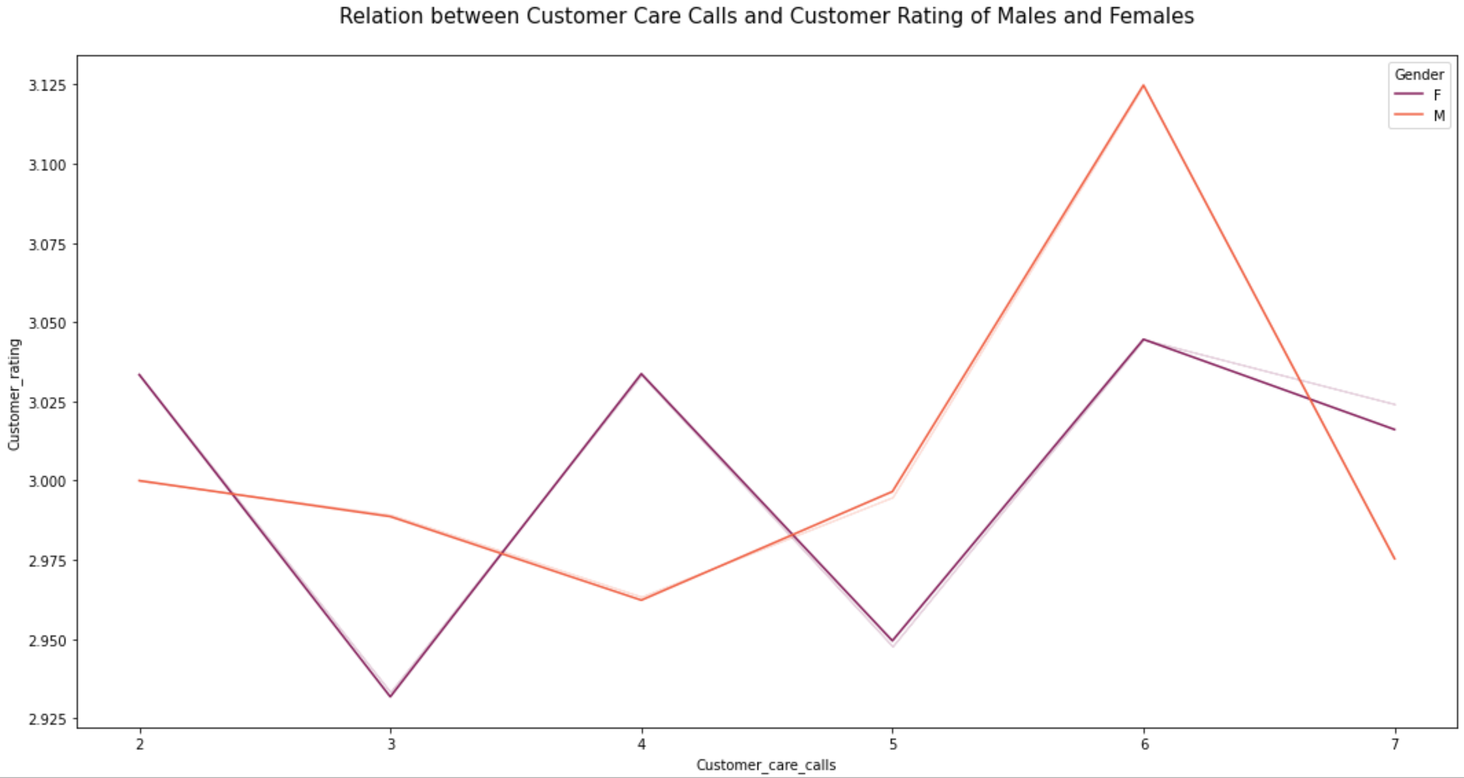
Cost of product vs Shipment mode

```
px.histogram(data_frame = df, x = 'Cost_of_the_Product', nbins = 100,
             color = 'Mode_of_Shipment',  marginal = 'box')
```



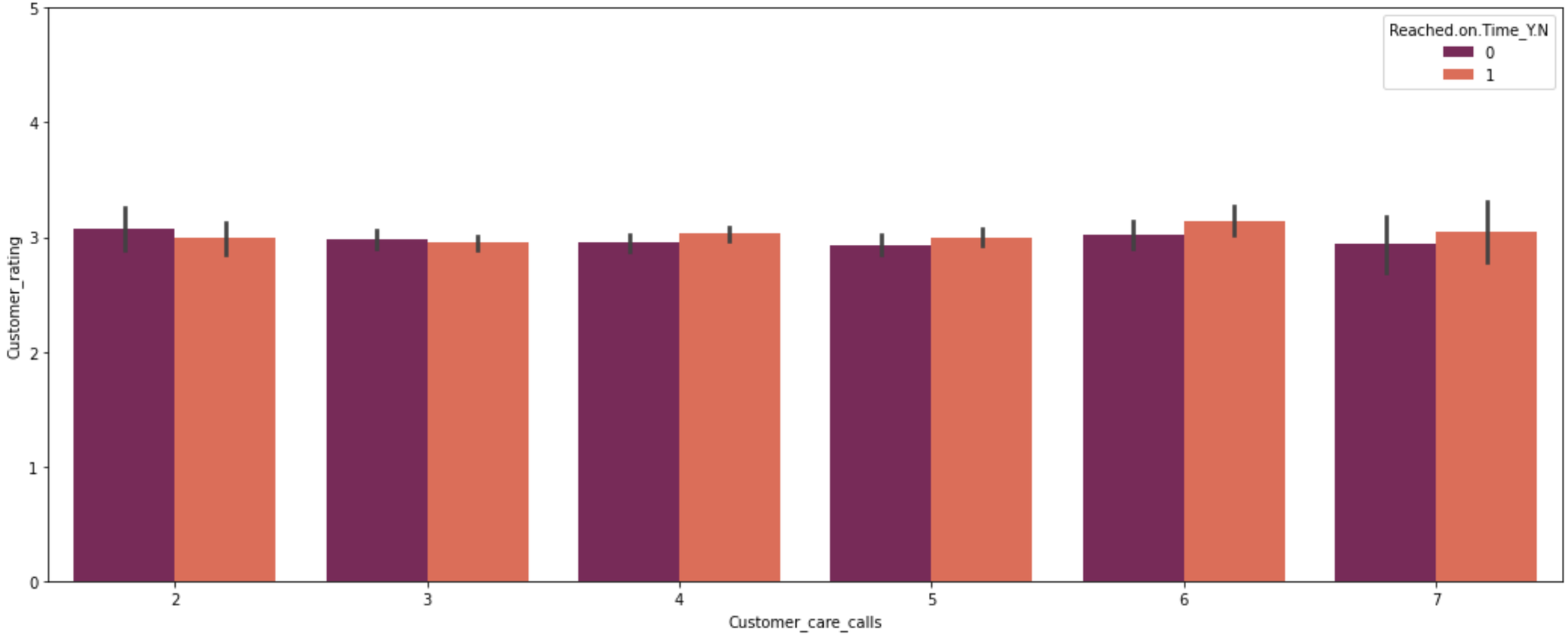
Customer call effects on Rating

```
plt.figure(figsize = (18, 9))
sns.lineplot(x = 'Customer_care_calls', y = 'Customer_rating', hue = 'Gender', data = df,
             palette = 'rocket', errorbar=('ci', 0))
plt.title('Relation between Customer Care Calls and Customer Rating of Males and Females\n',
          fontsize = 15)
plt.show()
```



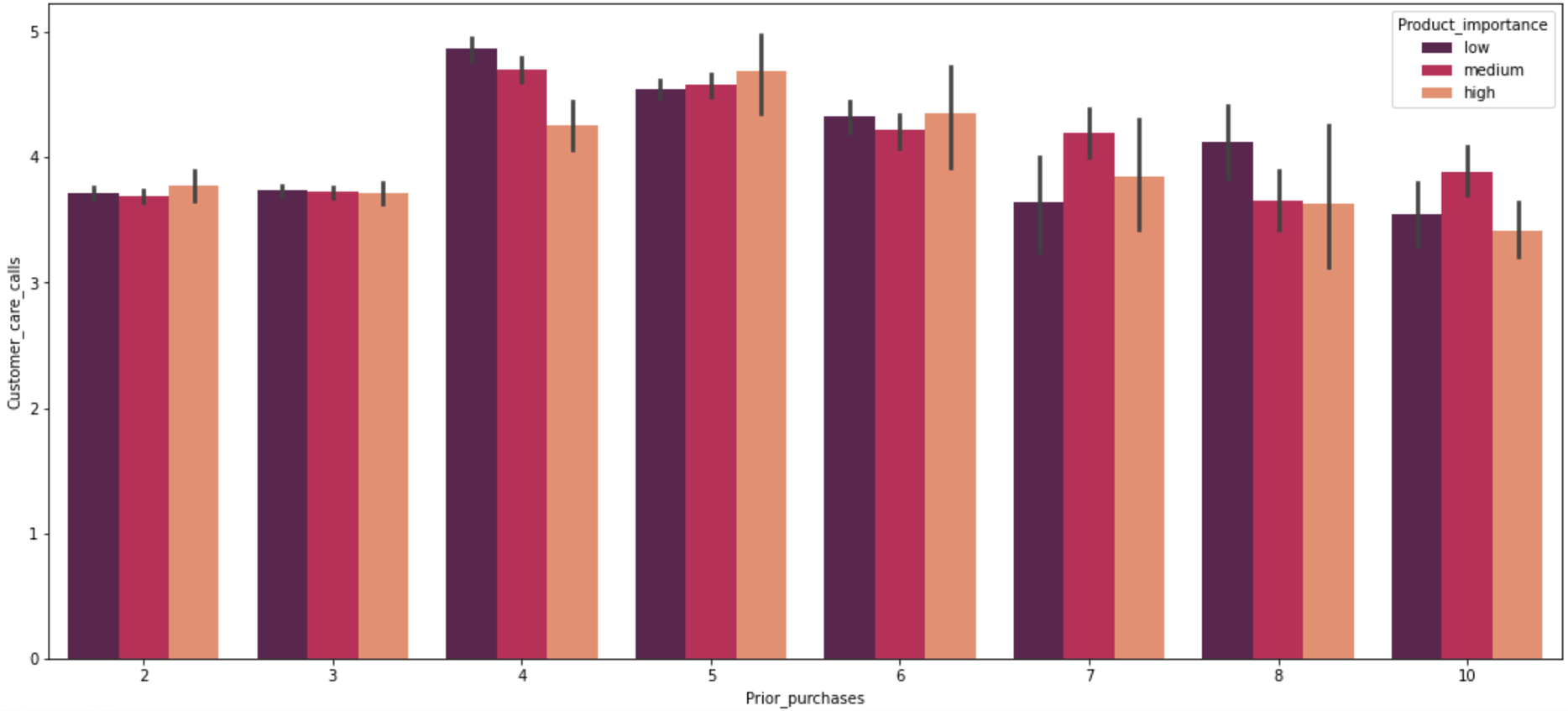
Customer's rating, call and shipment on time

```
1 plt.figure(figsize = (18, 7))
2 sns.barplot(x = 'Customer_care_calls', y = 'Customer_rating',
3             hue = 'Reached.on.Time_Y.N', data = df, palette = 'rocket')
4 plt.ylim(0, 5)
5 plt.show()
```



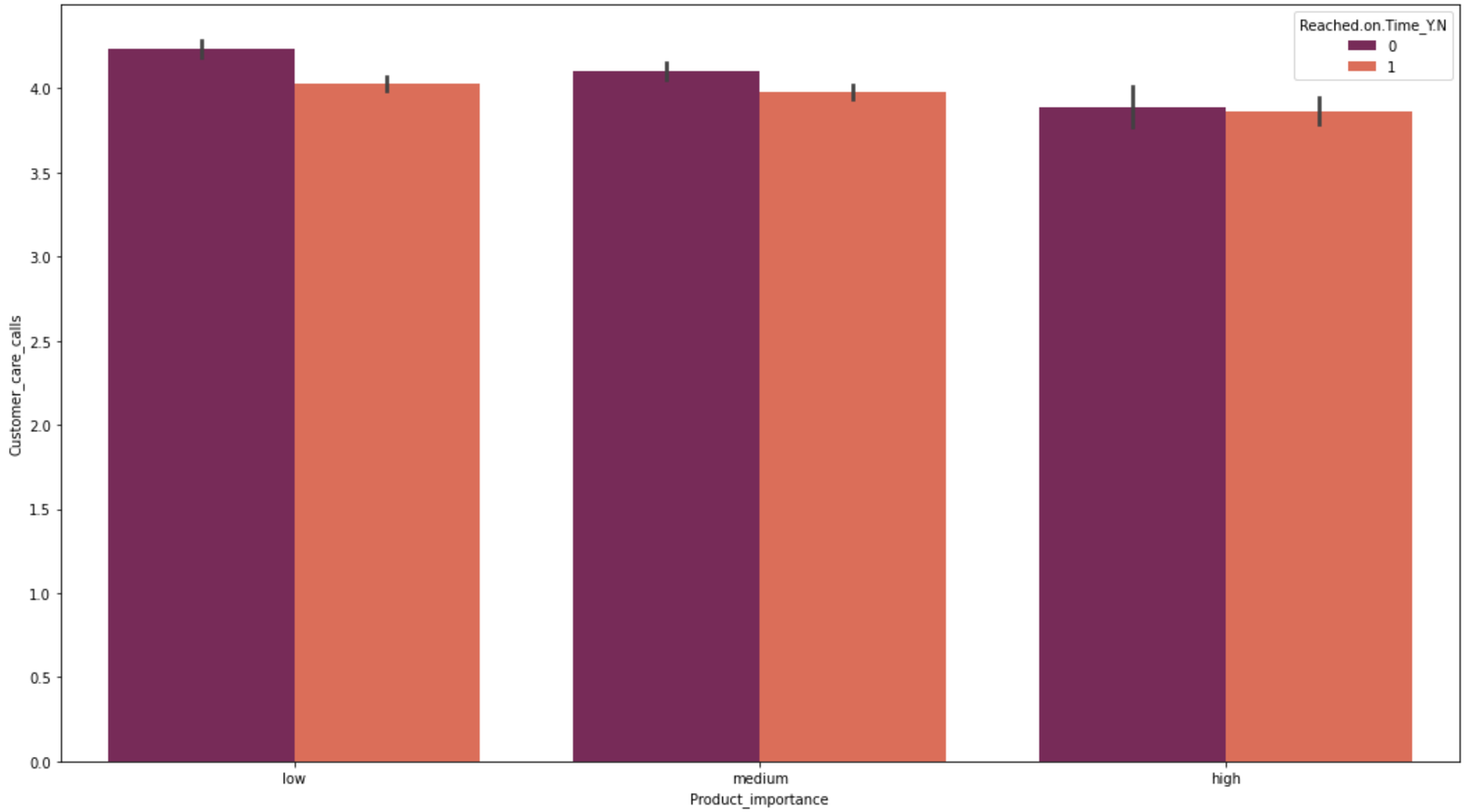
Customer calls, Prior purchase and Product importance

```
1 plt.figure(figsize = (18, 8))
2 sns.barplot(x = 'Prior_purchases', y = 'Customer_care_calls', data = df,
3               hue = 'Product_importance', palette = 'rocket')
4 plt.show()
```



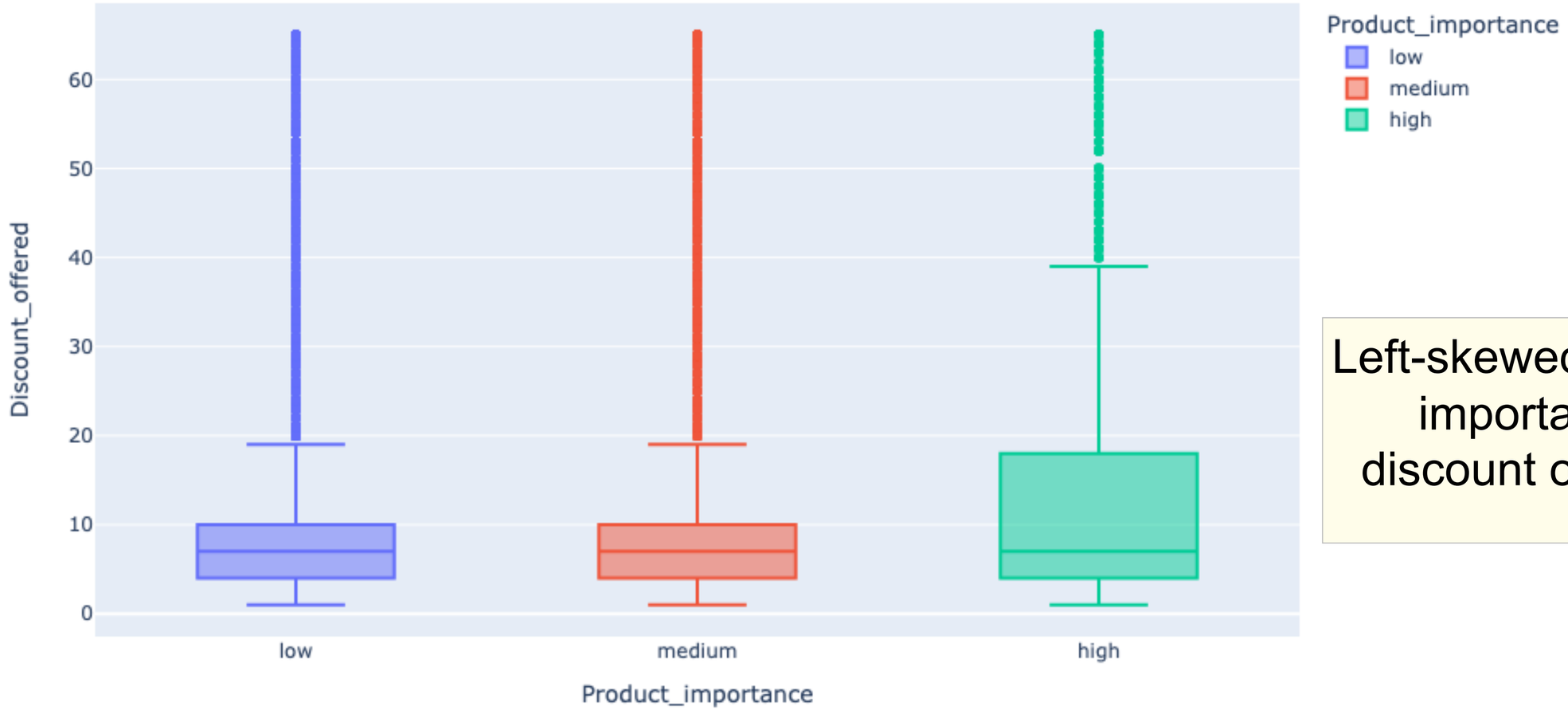
Product importance vs Customer call on Shipment on time

```
1 plt.figure(figsize = (18, 10))
2 sns.barplot(x='Product_importance', y = 'Customer_care_calls',
3              hue = 'Reached.on.Time_Y.N', data = df, palette = 'rocket')
4 plt.show()
```



Product importance and Discount offered

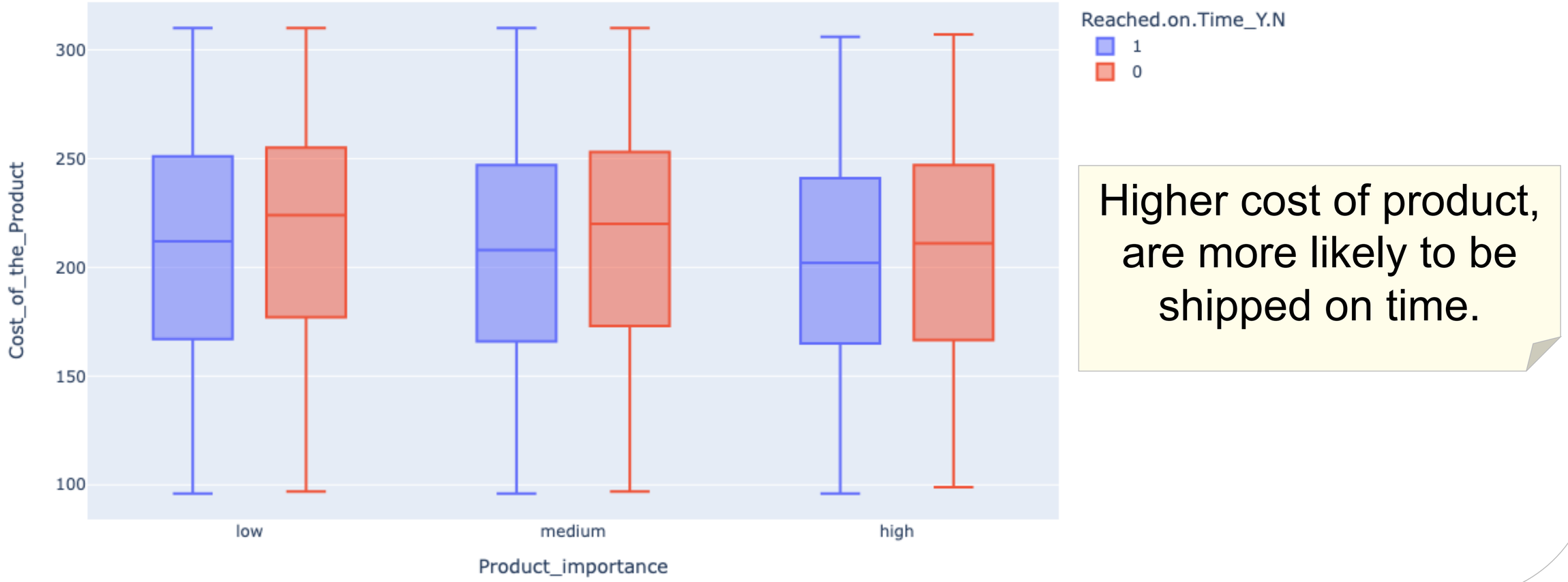
```
px.box(data_frame = df, x = 'Product_importance', y ='Discount_offered',
       color = 'Product_importance')
```



Left-skewed in high importance discount offered.

Cost of product vs importance on Shipment on time

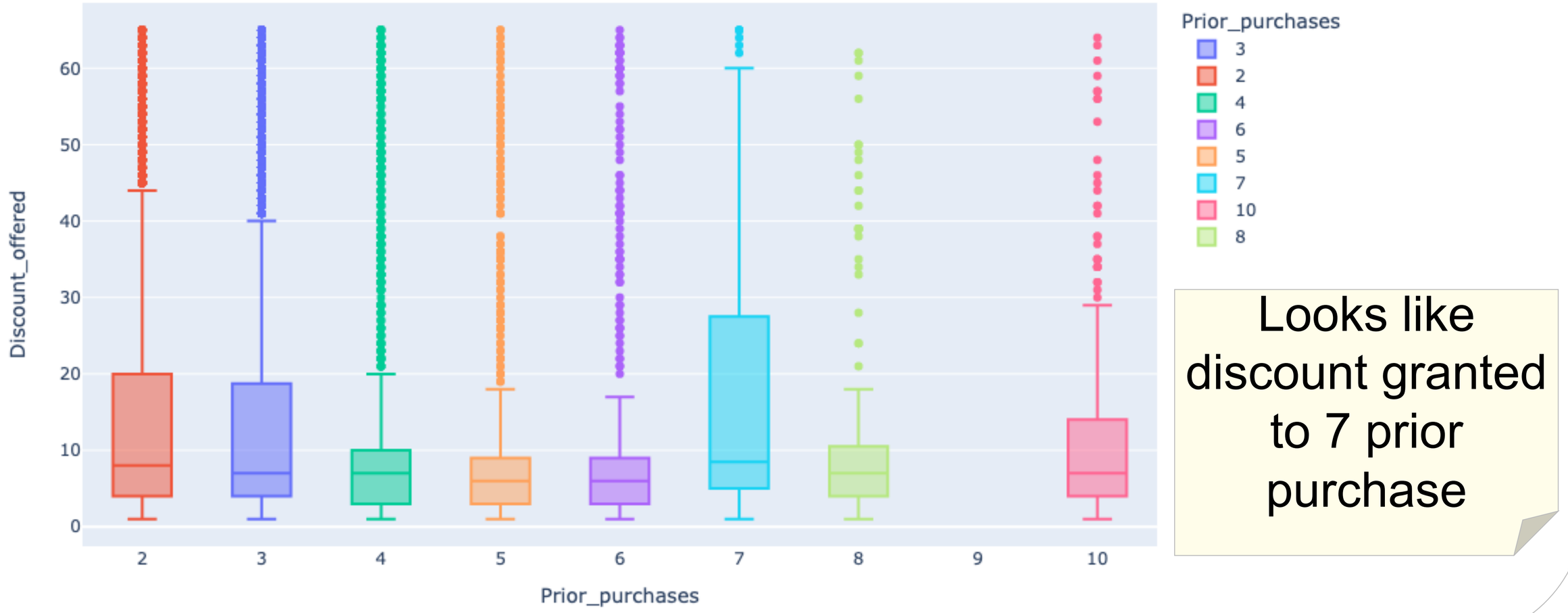
```
1  # 1 : NOT on time and 0: on time
2  px.box(data_frame = df, x = 'Product_importance', y ='Cost_of_the_Product',
3         color = 'Reached.on.Time_Y.N')
```



Higher cost of product, are more likely to be shipped on time.

Relation of Prior_purchases and Discount

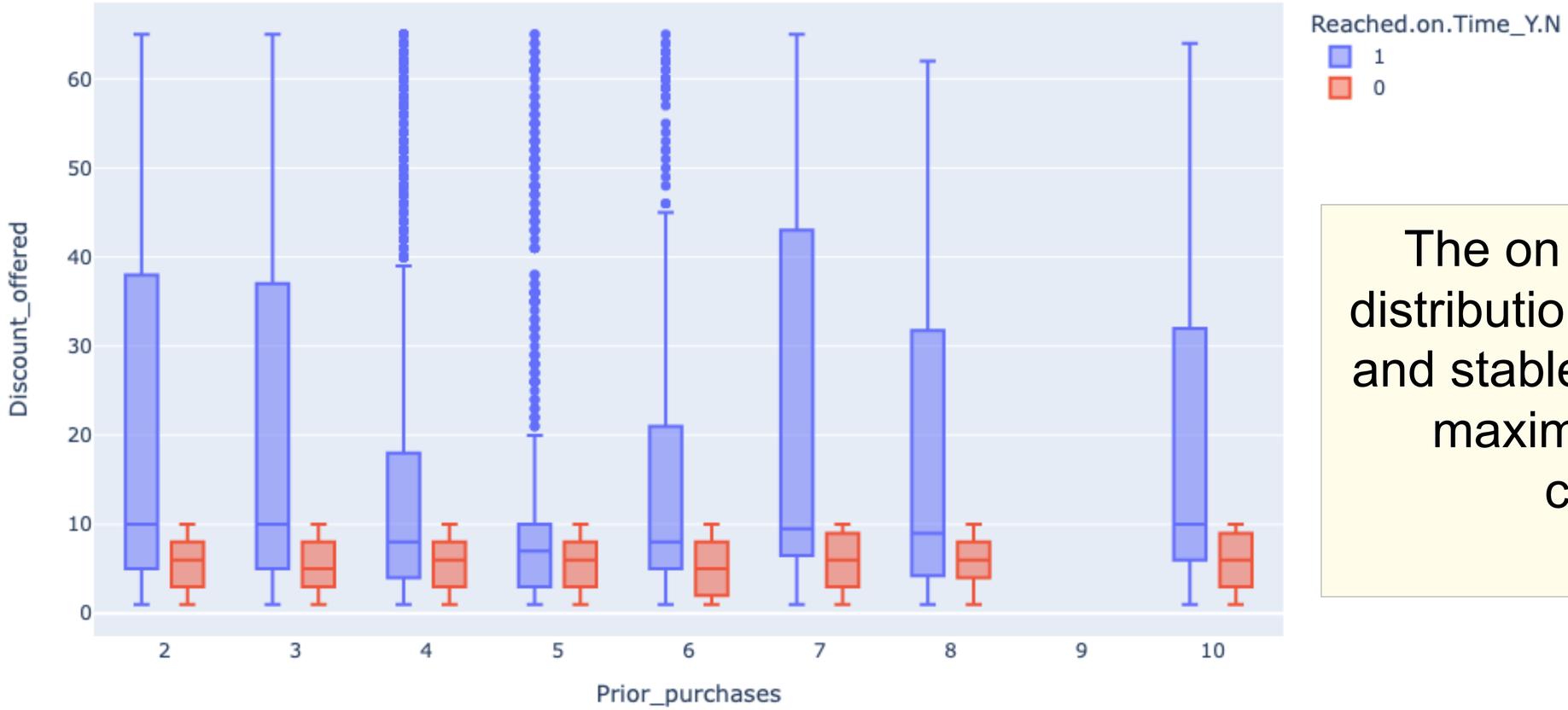
```
1  px.box(x = 'Prior_purchases', y = 'Discount_offered', data_frame = df,
2         color = 'Prior_purchases')
```



Looks like discount granted to 7 prior purchase

Prior_purchases and Discount Offered and Shipment on time

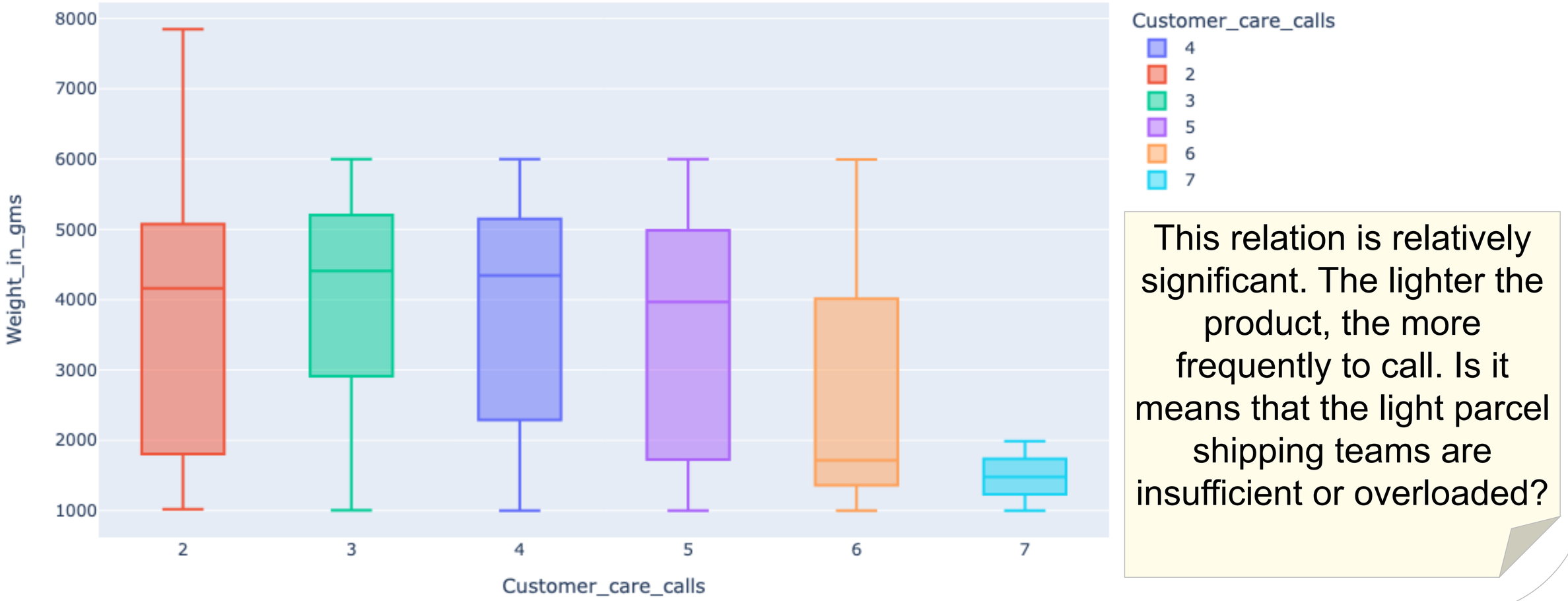
```
1  # 1 : NOT on time and 0: on time
2  px.box(x = 'Prior_purchases', y = 'Discount_offered', data_frame = df,
3         color = 'Reached.on.Time_Y.N')
```



The on time ratio and distribution is quite normal and stable. It might be the maximum shipping capacity.

Customer call and Weight in grams

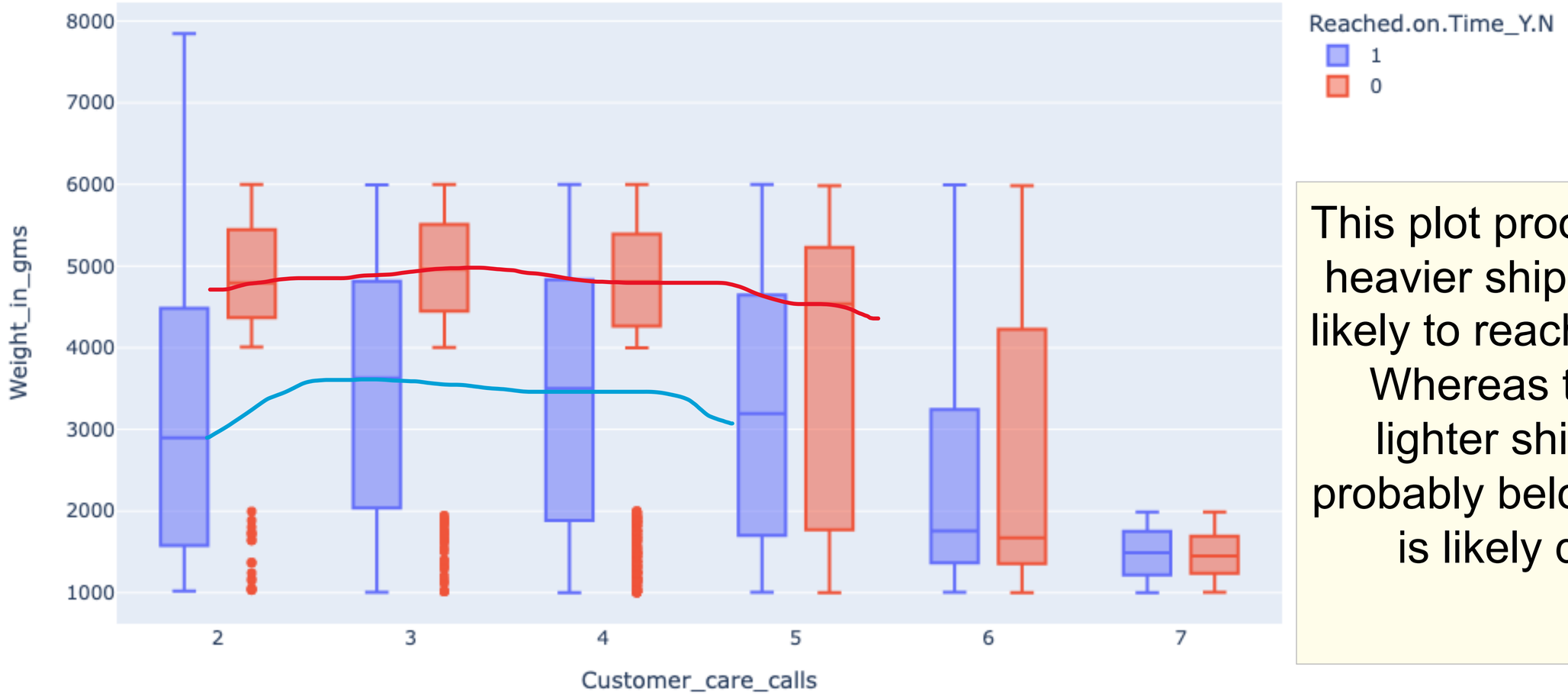
```
px.box(x = 'Customer_care_calls', y = 'Weight_in_gms', data_frame = df,
       color = 'Customer_care_calls')
```



This relation is relatively significant. The lighter the product, the more frequently to call. Is it means that the light parcel shipping teams are insufficient or overloaded?

Calls and Weight on Shipment on time

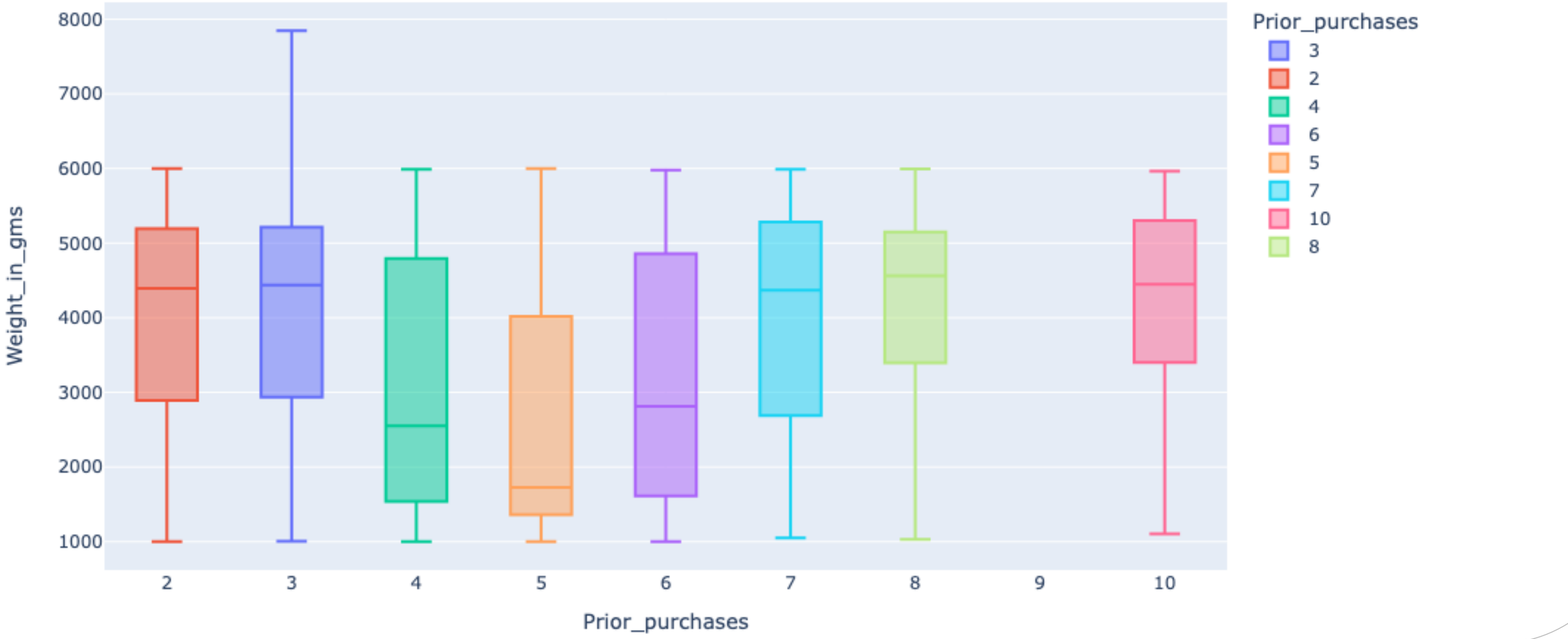
```
# 1 : NOT on time and 0: on time
px.box(x = 'Customer_care_calls', y = 'Weight_in_gms', data_frame = df,
       color = 'Reached.on.Time_Y.N')
```



This plot proof that the heavier shipment are likely to reach on time. Whereas the the lighter shipment probably below 3.5Kg, is likely delay.

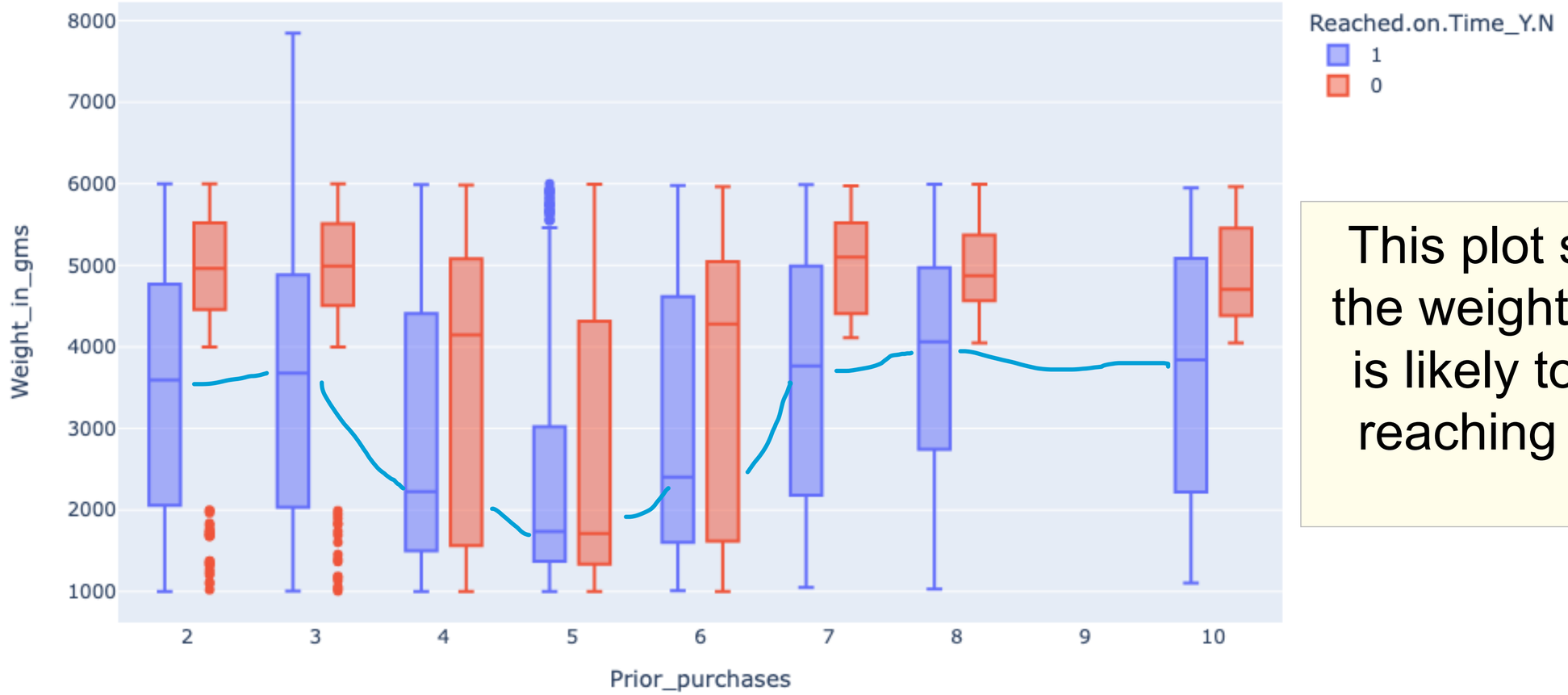
Relation of Prior purchase and Weight

```
px.box(x = 'Prior_purchases', y = 'Weight_in_gms', data_frame = df,
       color = 'Prior_purchases')
```



Prior purchases, Weights and Shipment on time

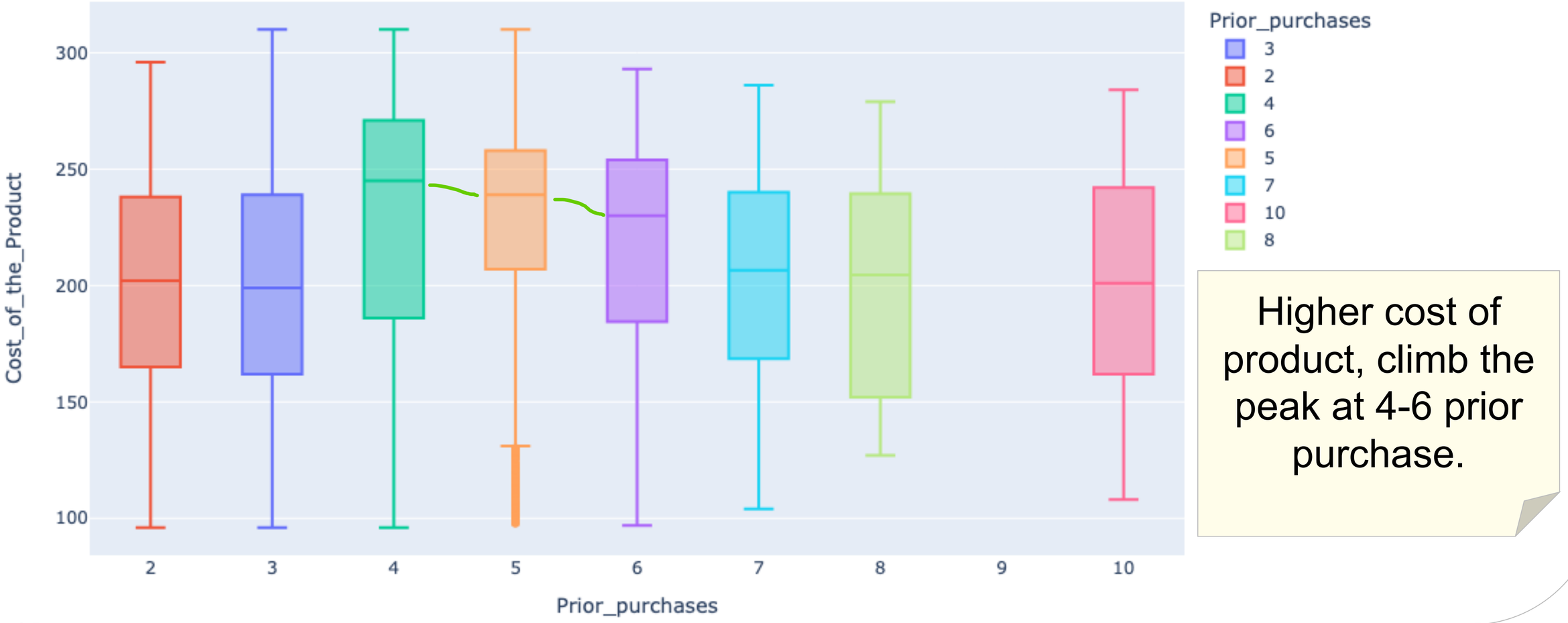
```
# 1 : NOT on time and 0: on time
px.box(x = 'Prior_purchases', y = 'Weight_in_gms', data_frame = df,
       color = 'Reached.on.Time_Y.N')
```



This plot shows that the weight under 4Kg is likely to be late in reaching shipment.

Relation of prior purchases and cost of the products

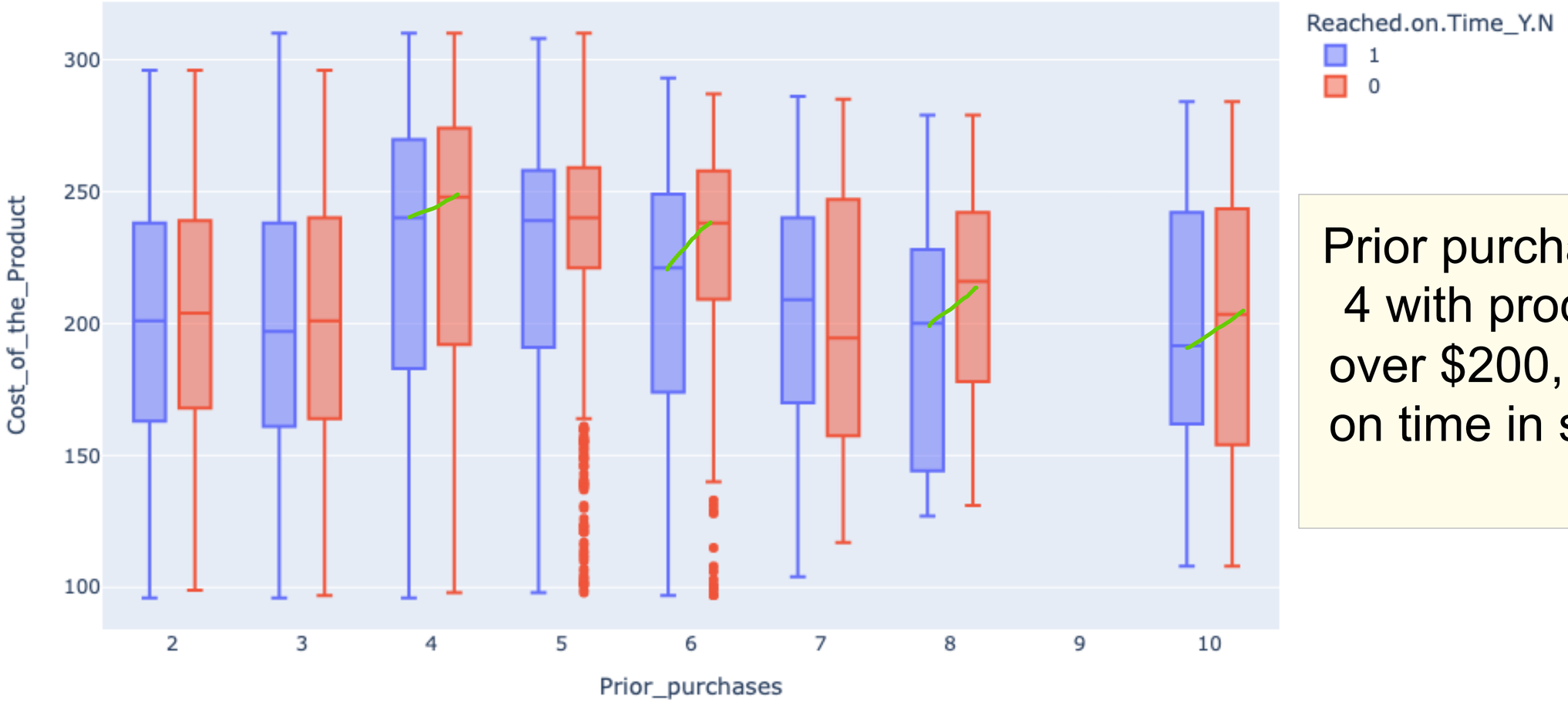
```
1  px.box(x = 'Prior_purchases', y = 'Cost_of_the_Product', data_frame = df,
2         color = 'Prior_purchases')
```



Higher cost of product, climb the peak at 4-6 prior purchase.

Prior purchases and Products cost on Shipment on time

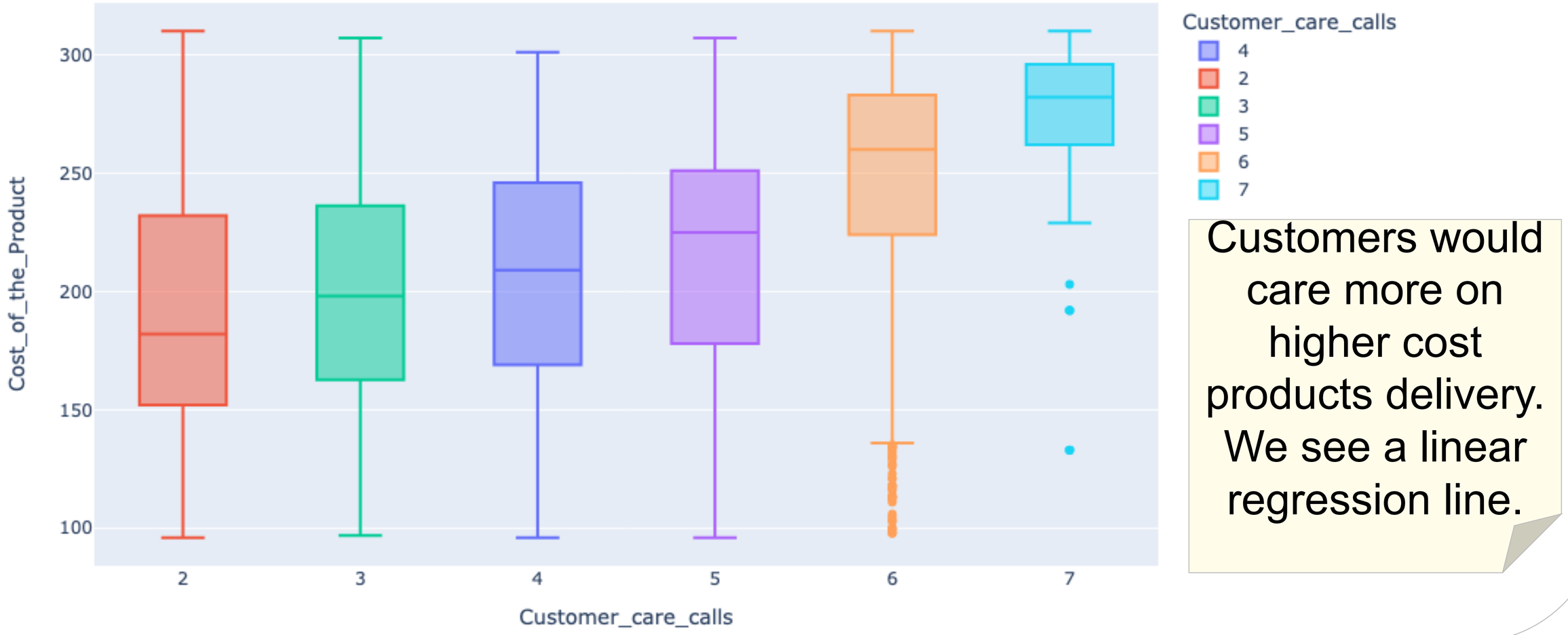
```
# 1 : NOT on time and 0: on time
px.box(x = 'Prior_purchases', y = 'Cost_of_the_Product', data_frame = df,
       color = 'Reached.on.Time_Y.N')
```



Prior purchases over 4 with product cost over $200, are likely on time in shipment.

Cost of the products and customer care calls

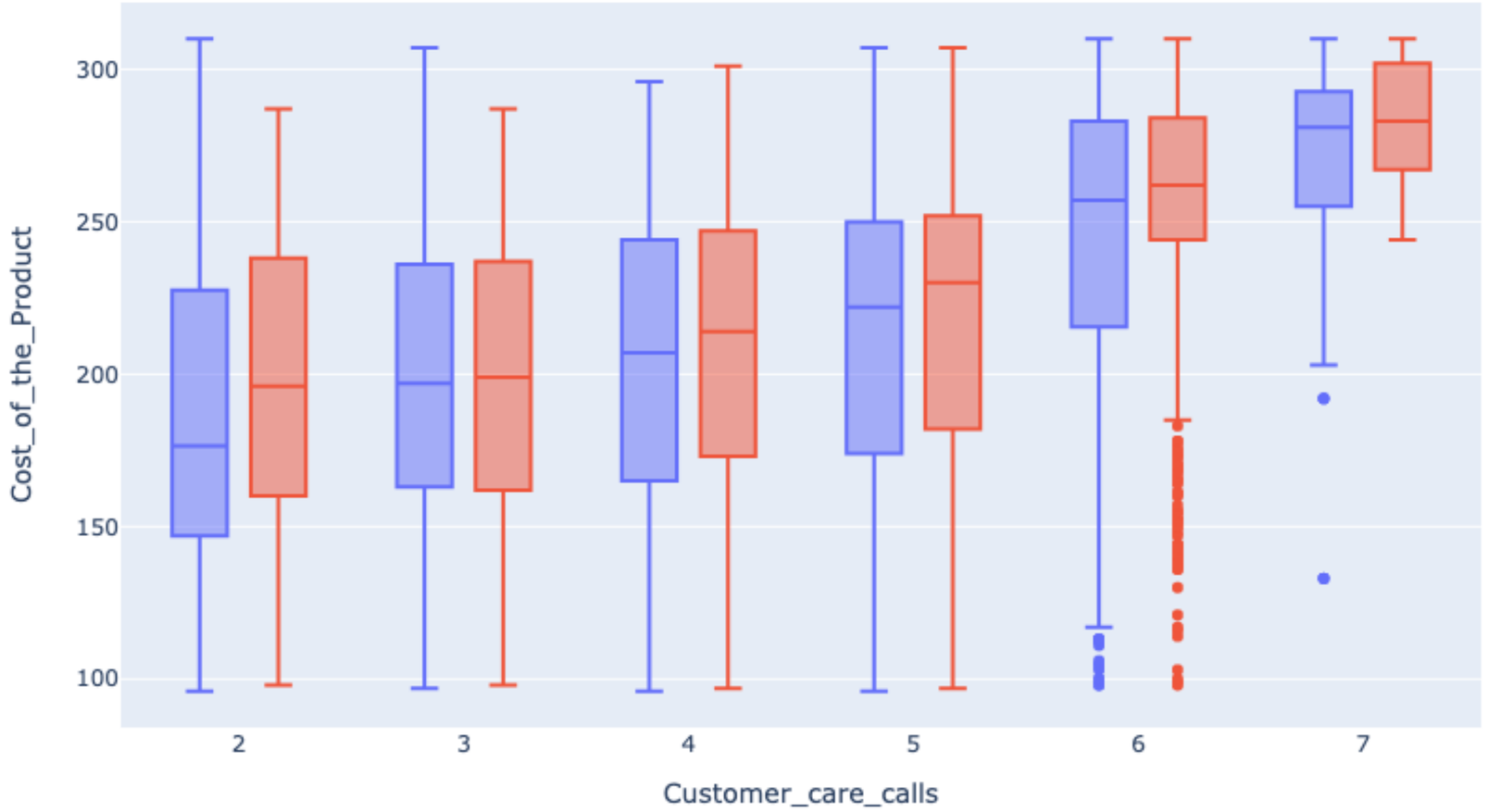
```
px.box(x = 'Customer_care_calls', y = 'Cost_of_the_Product', data_frame = df,
       color = 'Customer_care_calls')
```



Customers would care more on higher cost products delivery. We see a linear regression line.

Product cost and Customer call on Shipment on time

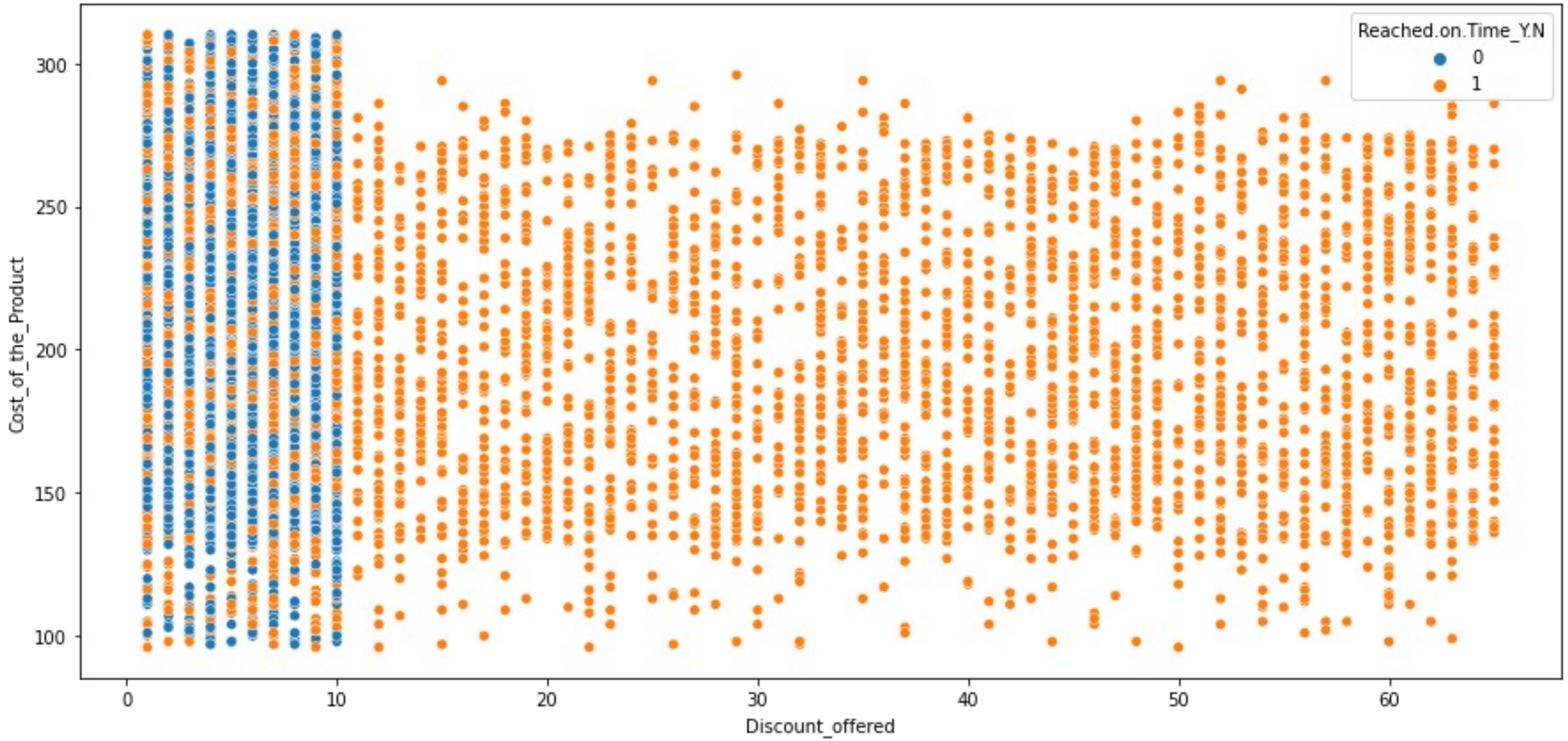
```
1  # 1 : NOT on time and 0: on time
2  px.box(x = 'Customer_care_calls', y = 'Cost_of_the_Product', data_frame = df,
3         color = 'Reached.on.Time_Y.N')
```



It seems that if product cost is low, no need to call, because there is no difference.
If the product cost is higher than $200, customer need to call 6 times to ensure the shipment on time.

Product Cost and Discount on Shipment on time

```
# 1 : NOT on time and 0: on time
plt.figure(figsize = (15, 7))
sns.scatterplot(x='Discount_offered', y='Cost_of_the_Product',
                data=df, hue='Reached.on.Time_Y.N')
plt.show()
```



For discount over 10%, the shipment is 100% late. We are not sure the 10% or more discount is the compensate and/or part of the reason being late as implied accepted fact.

Chapter Wrap Up

In these chapter we demonstrate lots of visualisation. Some of the statistics like distribution, or histogram are easy to convince audience by visuals.

Correlation of two variables could be positive or negative related. A heatmap of df.corr() will have the whole picture at a glimpse.

Sometime the target may correlate to more than 2 variables at the same time. This might need to elaborate in machine learning or regression analysis.

Reference & Resources

Official Website:

https://plotly.com/python/

Plotly Graph Objects:

https://plotly.com/python/graph-objects/

Seaborn:

https://seaborn.pydata.org/examples/index.html

LogisticRegression:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html