PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of degree of

INT216 - PYTHON PROJECT

Dataset: Data analysis (smart supply chain)

SUBMITTED BY

Name of student: - Manish Kumar Das

Registration Number: - 12017595

Roll Number: - RRE267A06

Section: - RE267

SUBMITTED TO

Lovely Professional University

(School of Computer Science and Engineering)

Index

Dataset-name: Data analysis (smart supply chain)

What dataset am I working with the dataset?

I am working with a dataset of ***supply chain by Dataco Global Company*** for my EDA project/Analysis.

Dataset link: https://www.kaggle.com/code/sanketchavan5595/data-analysis-smart-supply-chain/data

Why did I choose this dataset?

The supply chain is an overriding part of any online business. The supply chain starts with vendors on one side followed by a manufacturer and finally distribution of goods to consumers.

After the analysis, we will understand the importance of the supply chain in the present business scenario. What are the challenges of supply chain management? It will be very easy to analyze what steps can be taken to solve those challenges.

We can do several types of analysis like, consumer purchase rates, payment methods consumers prefer, which products cost the most, country vs. purchase ratio, etc.

Dataset dimension?

Row × Columns = **180519 × 53**

Dataset Information

Colum's Description:

COLUMN	DESCRIPTION
Type	*Type of transaction made*
Days for shipping (real)	*Actual shipping days of the purchased product*

Days for shipment (scheduled)	*Days of scheduled delivery of the purchased product*
Benefit per order	*Earnings per order placed*
Sales per customer	*Total sales per customer made per customer*
Delivery Status	*Delivery status of orders: Advance shipping , Late delivery , Shipping canceled , Shipping on time*
Late_delivery_risk	*Categorical variable that indicates if sending is late (1), it is not late (0).*
Category Id	*Product category code*
Category Name	*Description of the product category*
Customer City	*City where the customer made the purchase*
Customer Country	*Country where the customer made the purchase*
Customer Email	*Customer's email*
Customer Fname	*Customer name*
Customer Id	*Customer ID*
Customer Lname	*Customer lastname*
Customer Password	*Masked customer key*
Customer Segment	*Types of Customers: Consumer , Corporate , Home Office*
Customer State	*State to which the store where the purchase is registered belongs*
Customer Street	*Street to which the store where the purchase is registered belongs*
Customer Zipcode	*Customer Zipcode*
Department Id	*Department code of store*
Department Name	*Department name of store*
Latitude	*Latitude corresponding to location of store*
Longitude	*Longitude corresponding to location of store*
Market	*Market to where the order is delivered : Africa, Europe, LATAM, Pacific Asia, USCA*
Order City	*Destination city of the order*
Order Country	*Destination country of the order*
Order Customer Id	*Customer order code*
order date (DateOrders)	*Date on which the order is made*
Order Id	*Order code*
Order Item Cardprod Id	*Product code generated through the RFID reader*
Order Item Discount	*Order item discount value*
Order Item Discount Rate	*Order item discount percentage*
Order Item Id	*Order item code*
Order Item Product Price	*Price of products without discount*
Order Item Profit Ratio	*Order Item Profit Ratio*

Order Item Quantity	*Number of products per order*
Sales	*Value in sales*
Order Item Total	*Total amount per order*
Order Profit Per Order	*Order Profit Per Order*
Order Region	*Region of the world where the order is delivered : Southeast Asia, South Asia, Oceania, Eastern Asia, West Asia, West of USA, US Center, West Africa, Central Africa, North Africa, Western Europe, Northern, Caribbean, South America, East Africa, Southern Europe, East of USA, Canada, Southern Africa, Central Asia, Europe, Central America, Eastern Europe, South of USA*
Order State	*State of the region where the order is delivered*
Order Status	*Order Status : COMPLETE, PENDING, CLOSED, PENDING_PAYMENT, CANCELED, PROCESSING, SUSPECTED_FRAUD, ON_HOLD, PAYMENT_REVIEW*
Product Card Id	*Product code*
Product Category Id	*Product category code*
Product Description	*Product Description*
Product Image	*Link of visit and purchase of the product*
Product Name	*Product Name*
Product Price	*Product Price*
Product Status	*Status of the product stock :If it is 1 not available , 0 the product is available*
Shipping date (DateOrders)	*Exact date and time of shipment*
Shipping Mode	*The following shipping modes are presented : Standard Class , First Class , Second Class , Same Day*

Column's Datatype:

```
#   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   Type                          180519 non-null  object
 1   Days for shipping (real)      180519 non-null  int64
 2   Days for shipment (scheduled) 180519 non-null  int64
 3   Benefit per order             180519 non-null  float64
 4   Sales per customer            180519 non-null  float64
 5   Delivery Status               180519 non-null  object
 6   Late_delivery_risk            180519 non-null  int64
 7   Category Id                   180519 non-null  int64
 8   Category Name                 180519 non-null  object
 9   Customer City                 180519 non-null  object
```

```
10  Customer Country              180519 non-null  object
11  Customer Email                180519 non-null  object
12  Customer Fname                180519 non-null  object
13  Customer Id                   180519 non-null  int64
14  Customer Lname                180511 non-null  object
15  Customer Password             180519 non-null  object
16  Customer Segment              180519 non-null  object
17  Customer State                180519 non-null  object
18  Customer Street               180519 non-null  object
19  Customer Zipcode              180516 non-null  float64
20  Department Id                 180519 non-null  int64
21  Department Name               180519 non-null  object
22  Latitude                      180519 non-null  float64
23  Longitude                     180519 non-null  float64
24  Market                        180519 non-null  object
25  Order City                    180519 non-null  object
26  Order Country                 180519 non-null  object
27  Order Customer Id             180519 non-null  int64
28  order date (DateOrders)       180519 non-null  object
29  Order Id                      180519 non-null  int64
30  Order Item Cardprod Id        180519 non-null  int64
31  Order Item Discount           180519 non-null  float64
32  Order Item Discount Rate      180519 non-null  float64
33  Order Item Id                 180519 non-null  int64
34  Order Item Product Price      180519 non-null  float64
35  Order Item Profit Ratio       180519 non-null  float64
36  Order Item Quantity           180519 non-null  int64
37  Sales                         180519 non-null  float64
38  Order Item Total              180519 non-null  float64
39  Order Profit Per Order        180519 non-null  float64
40  Order Region                  180519 non-null  object
41  Order State                   180519 non-null  object
42  Order Status                  180519 non-null  object
43  Order Zipcode                 24840 non-null   float64
44  Product Card Id               180519 non-null  int64
45  Product Category Id           180519 non-null  int64
46  Product Description           0 non-null       float64
47  Product Image                 180519 non-null  object
48  Product Name                  180519 non-null  object
49  Product Price                 180519 non-null  float64
50  Product Status                180519 non-null  int64
```

```
51  shipping date (DateOrders)     180519 non-null  object
52  Shipping Mode                  180519 non-null  objec
```

Planning

1. Analysis the quantity and distribution of data.
2. Dataset cleaning like remove duplicate/irrelevant/redundant column
   - We can saw that some columns are duplicate which contain same data. So, we can drop one of the duplicate columns. i.e.: "Order Profit Per Order" & Benefit per order"
3. Handling Attributes w/o Variance
   - As seen in checking for variance in the data, columns that have no variance (e.g. only one value) can also be excluded.
4. Handling NaN values
   - Order_Zipcode
   - Product_Description
5. Aggregating Department Information
   - We can aggregate information available about different apartments to facilitate management decision making.
6. Mapping Supply Chain as a Bipartite Graph
   - We can also map the supply chain as a bipartite graph where one node set represents department stores and the other represents customers (customer regions need to be more specific.)
7. Heatmap for correlation matrix
8. Delvery status according to country
9. Analysis the type of customers. Like: Individual, Corporate, Home office
10. We can analyze that which type of product/which categories product consumers order most
11. We can analyze that which type of product/which categories product consumers order most by region/country wise as well
12. Analyze the numbers of order orders respect to country
13. Which year minimum/maximum sales for a particular product.
14. Various plots/Graph like Barplot, Scatterplot, PieChart, Kde plot, Boxplot, Violinplot, Bargraph, Countplot, Crosstablish, stackedbar, heatmaps
15. Hypothesis testing

Data Cleaning

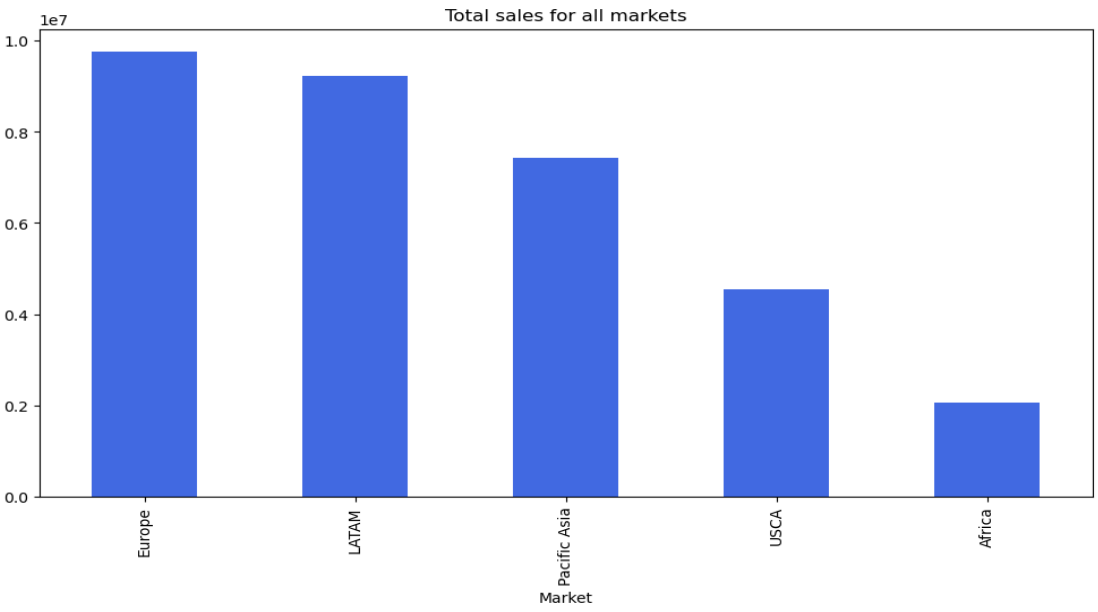
1. Two columns have Invalid Datatype I corrected it.
2. Remove the duplicate columns by writing User-defined-function called "***getDuplicateColumns***".
3. Remove the columns which contains only single values for all of the rows like password & email column etc. For that I used "***getOneDistinctValueCols***" functions
4. I merge two column called "***Customer Fname***" & "***Customer Lname***" and Make a new Column named "**Customer Name**" & Drop those 2 columns.

5. Then I checked that two columns contains lots of null values & this column is "***Customer Zipcode***". These columns are not necessary for out data analysis so I drop these 2 column.
6. Drop some others columns like "Product Image", "Latitude" and "Longitude" columns as it is useless for analysis

Univariate Analysis

1. Which markets has most sales?

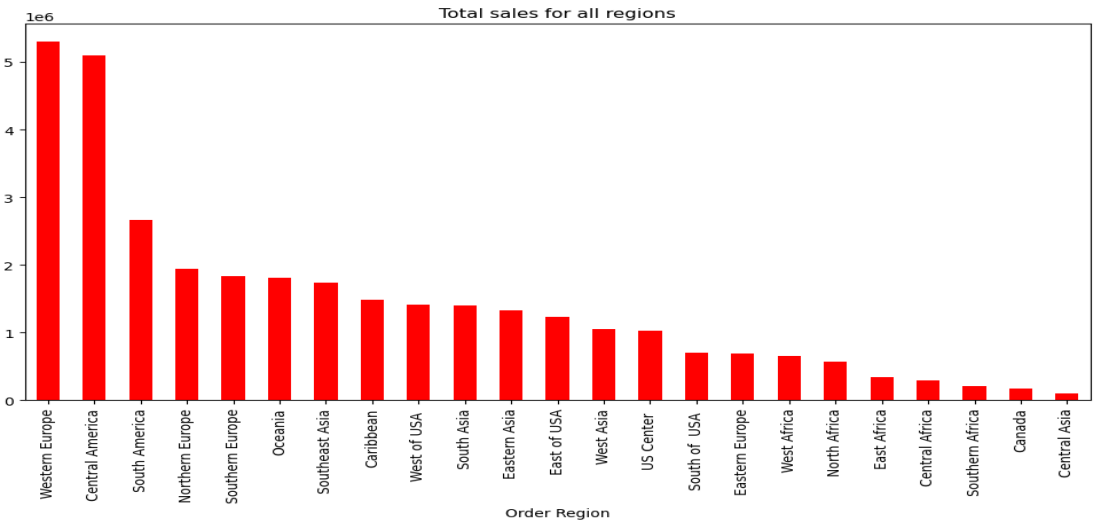
```
market = df.groupby("Market") #Grouping by market
market['Sales per customer'].sum().sort_values(ascending=False).plot.bar(figsize=(12,6), title="Total sales for all
```



Observations: Maximum items sales in **European** market and **African** market is least

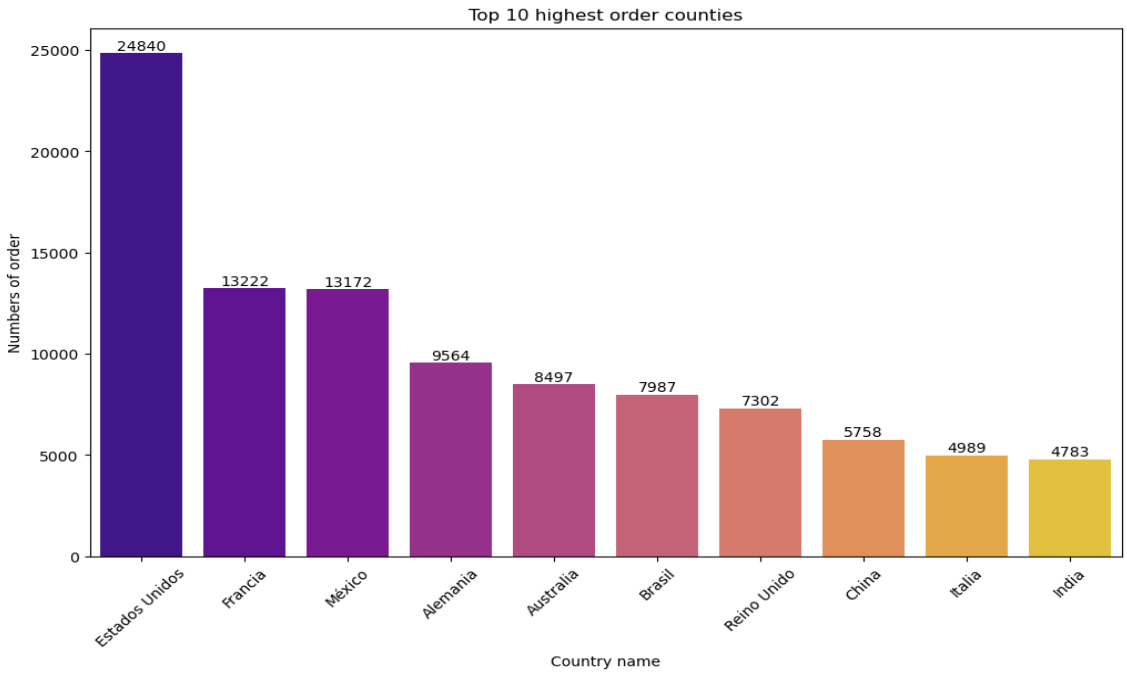
2. Which region has most sales?

```
region = df.groupby("Order Region") # Grouping by Region
region['Sales per customer'].sum().sort_values(ascending=False).plot.bar(figsize=(12,6), title="Total sales for all
```



3. Which country has made heighest orders?

```
# Count Plot
plt.figure(figsize=(12,7))
ax = sns.countplot(data=df, x="Order Country", order=pd.value_counts(df["Order Country"]).iloc[:10].index, palette="
plt.title("Top 10 highest order counties")
plt.xlabel("Country name")
plt.ylabel("Numbers of order")

for container in ax.containers:
    ax.bar_label(container) # showing values on countplot bar

plt.xticks(rotation=45)
plt.show()
```

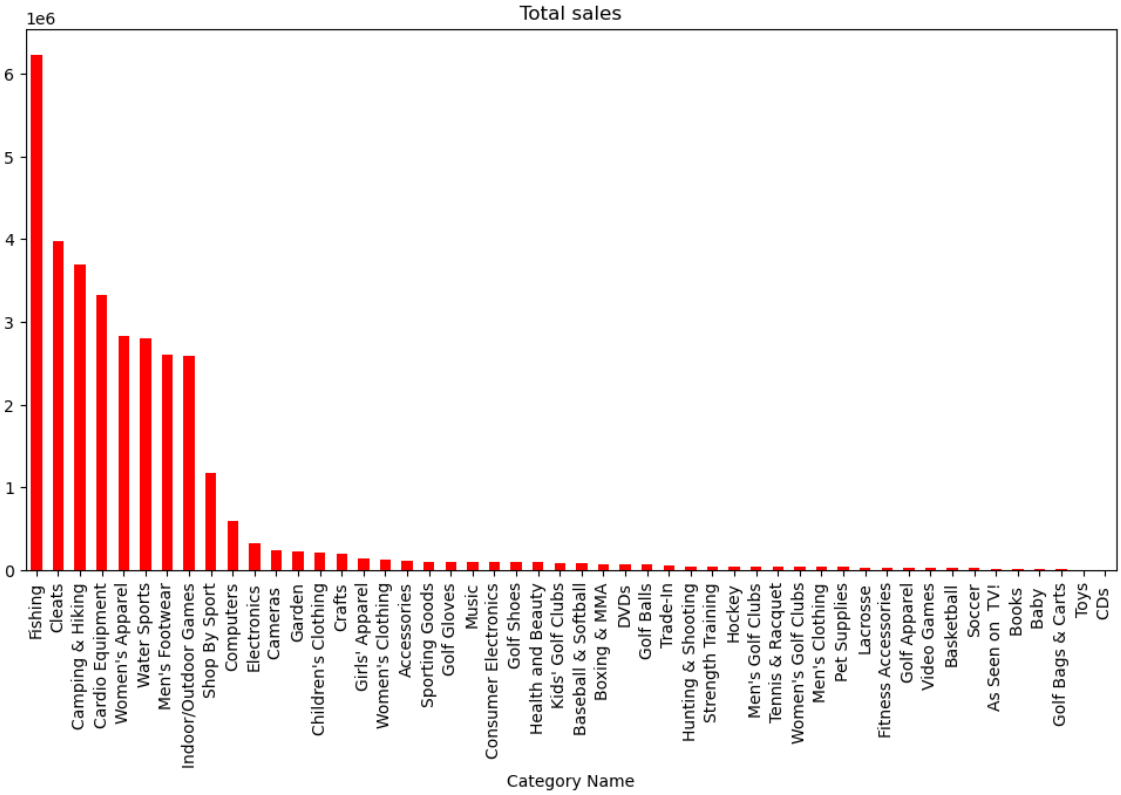


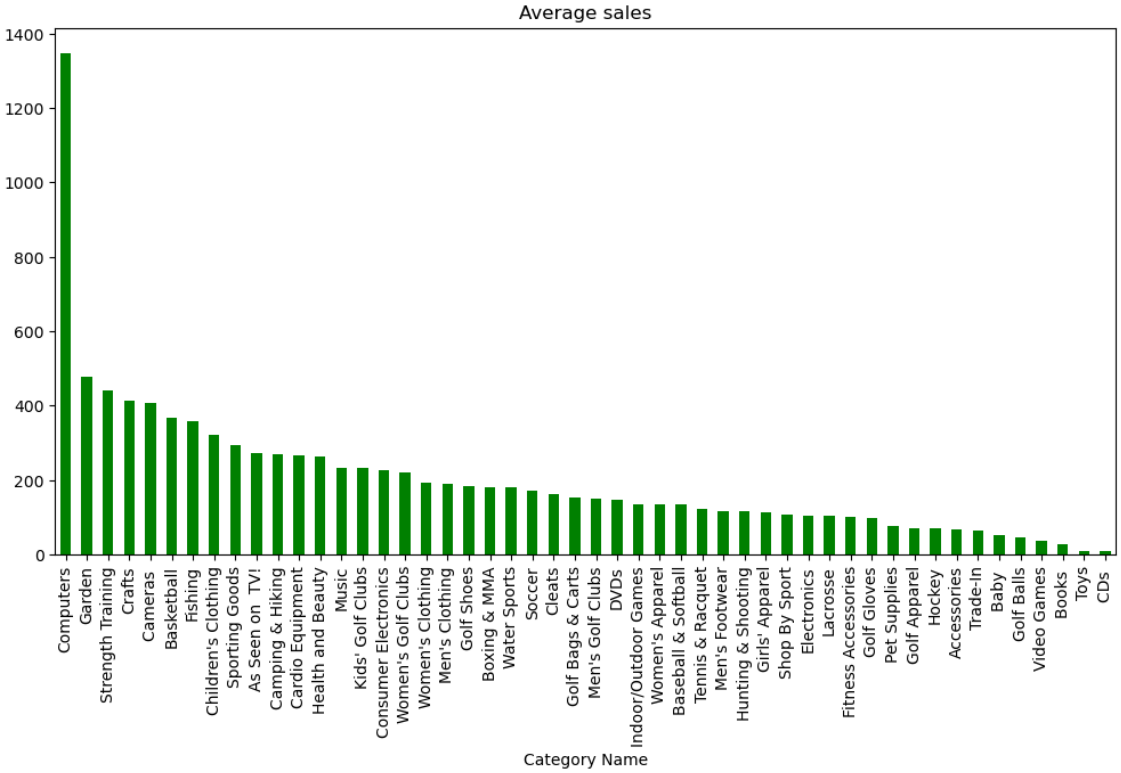
Observation: From the above graph, we can observe that people from ***Estados Unidos*** country have height number of order

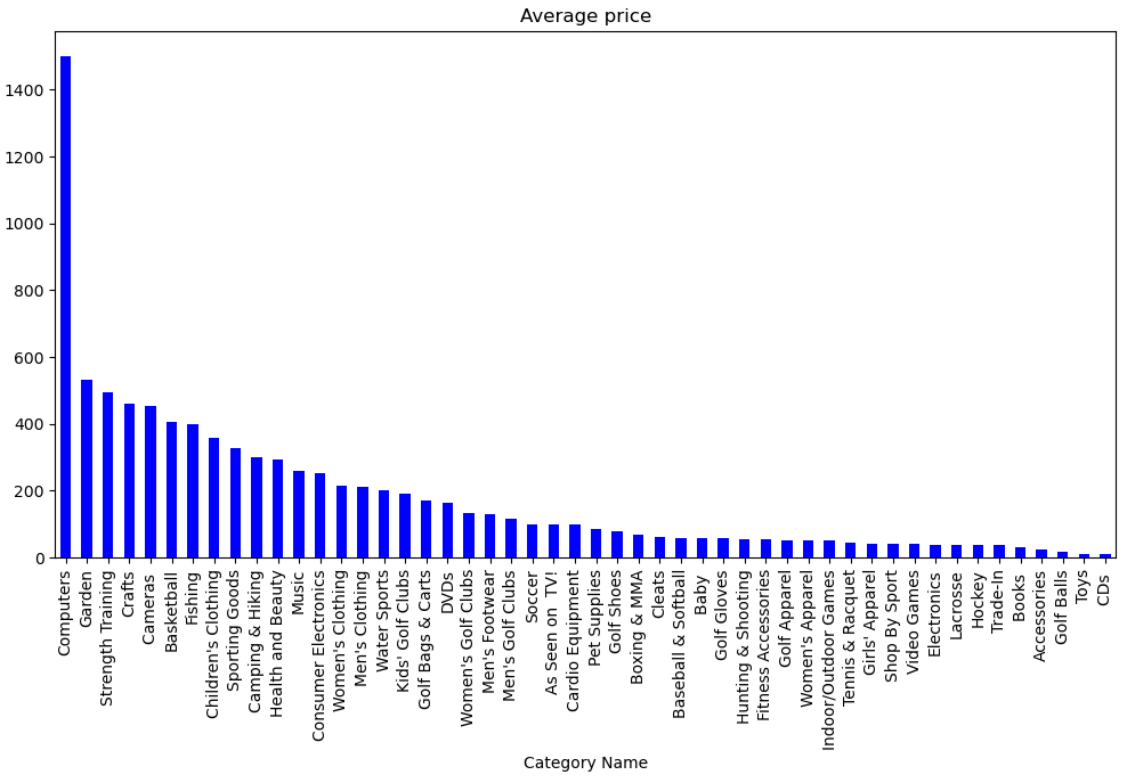
4. Which category of products has highest sales?

The same procedure can be followed here to see the product category with highest sales

```
categories = df.groupby('Category Name') # Grouping by categories

plt.figure(1)
# Total sum of sales for all categories
categories['Sales per customer'].sum().sort_values(ascending=False).plot.bar(figsize=(12,6), title="Total sales", co

# Mean sales for all categories
plt.figure(2)
categories['Sales per customer'].mean().sort_values(ascending=False).plot.bar(figsize=(12,6), title="Average sales",

plt.figure(3)
# Mean prices for all categories
categories['Order Item Product Price'].mean().sort_values(ascending=False).plot.bar(figsize=(12,6), title="Average p
```


Total sales
1e6
0
1
2
3
4
5
6
Fishing
Cleats
Camping & Hiking
Cardio Equipment
Women's Apparel
Water Sports
Men's Footwear
Indoor/Outdoor Games
Shop By Sport
Computers
Electronics
Cameras
Garden
Children's Clothing
Crafts
Girls' Apparel
Women's Clothing
Accessories
Sporting Goods
Golf Gloves
Music
Consumer Electronics
Golf Shoes
Health and Beauty
Kids' Golf Clubs
Baseball & Softball
Boxing & MMA
DVDs
Golf Balls
Trade-In
Hunting & Shooting
Strength Training
Hockey
Men's Golf Clubs
Tennis & Racquet
Women's Golf Clubs
Men's Clothing
Pet Supplies
Lacrosse
Fitness Accessories
Golf Apparel
Video Games
Basketball
Soccer
As Seen on TV!
Books
Baby
Golf Bags & Carts
Toys
CDs
Category Name


Average sales
0
200
400
600
800
1000
1200
1400
Computers
Garden
Strength Training
Crafts
Cameras
Basketball
Fishing
Children's Clothing
Sporting Goods
As Seen on TV!
Camping & Hiking
Cardio Equipment
Health and Beauty
Music
Kids' Golf Clubs
Consumer Electronics
Women's Golf Clubs
Women's Clothing
Men's Clothing
Golf Shoes
Boxing & MMA
Water Sports
Soccer
Cleats
Golf Bags & Carts
Men's Golf Clubs
DVDs
Indoor/Outdoor Games
Women's Apparel
Baseball & Softball
Tennis & Racquet
Men's Footwear
Hunting & Shooting
Girls' Apparel
Shop By Sport
Electronics
Lacrosse
Fitness Accessories
Golf Gloves
Pet Supplies
Golf Apparel
Hockey
Accessories
Trade-In
Baby
Golf Balls
Video Games
Books
Toys
CDs
Category Name



Observations: As we can see from Figure-1 that fishing category had the highest number of sales followed by Cleats. However it is suprising to see that top 8 products with highest price on average are the most sold products on average with computers having almost 1350 sales despite price being 1500$

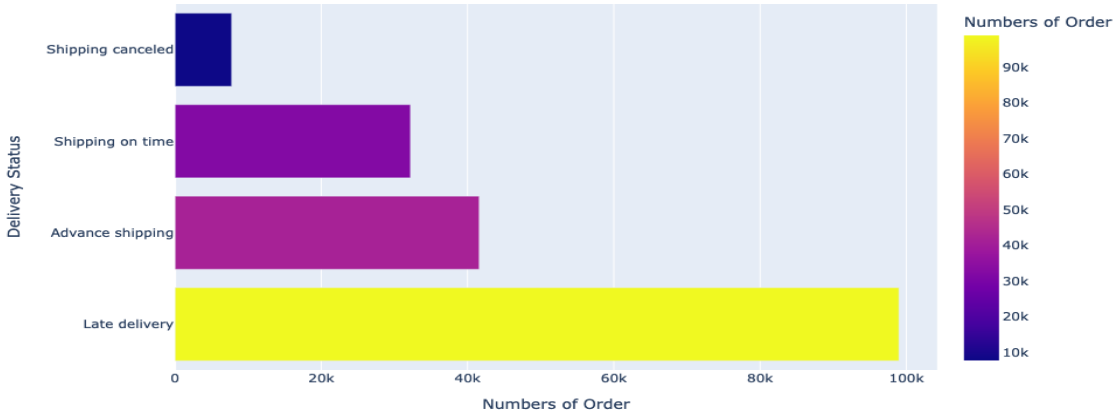
5. Order Number respect to Delivery Status

```
# Let's check Order Number respect to Delivery Status
delivery_status_count = df.groupby(["Delivery Status"]).size().reset_index(name="Numbers of Order").sort_values(by="
delivery_status_count
```

	Delivery Status	Numbers of Order
1	Late delivery	98977
0	Advance shipping	41592
3	Shipping on time	32196
2	Shipping canceled	7754

Observations: As we can see There is a very high number of order in Late delivery. Let's plot in in a graph.

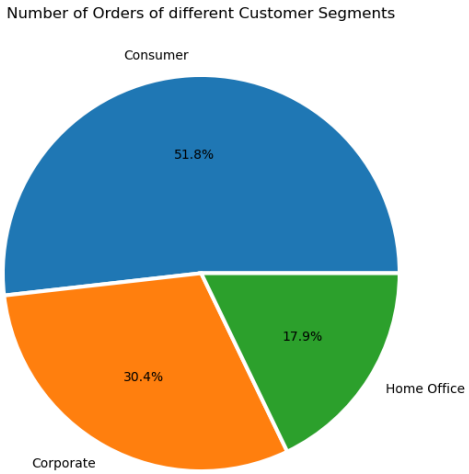
```
# BarPlot
px.bar(delivery_status_count, y="Delivery Status", x="Numbers of Order", color="Numbers of Order")
```



6. Numbers of orders of different customer segments

```
# PieChart - Customer Segments
data_customer_segment = df.groupby(['Customer Segment'])['Order Id'].count().reset_index(name='Number of Orders').so
print(data_customer_segment)

plt.figure(figsize=(12,7))
plt.pie(data_customer_segment["Number of Orders"], labels=data_customer_segment["Customer Segment"], autopct="%.1f%%
plt.title("Number of Orders of different Customer Segments");
plt.show()
```

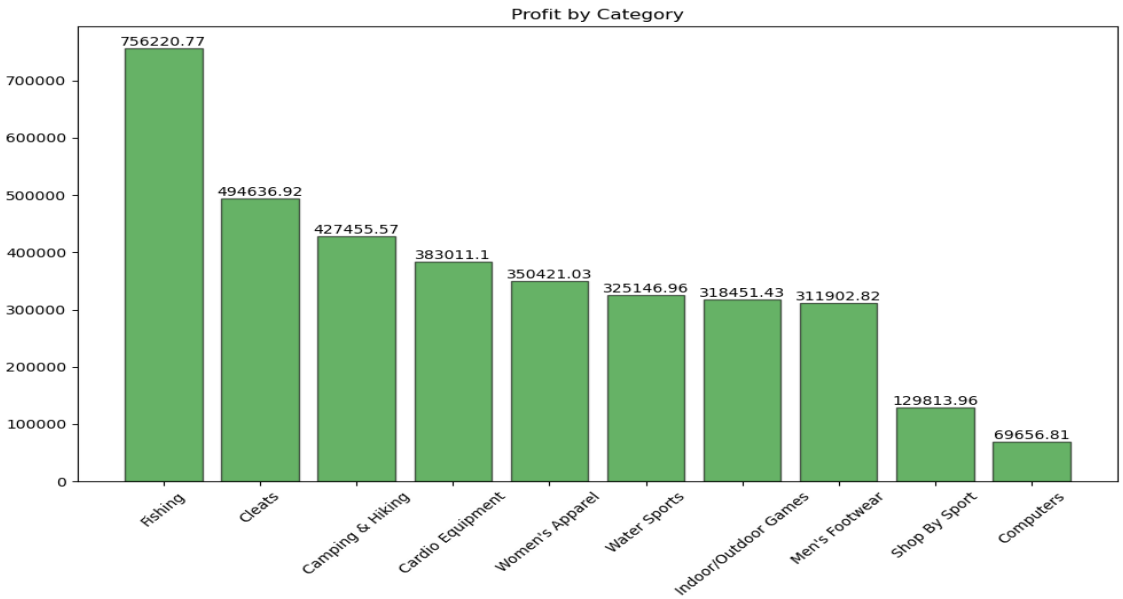
```
  Customer Segment  Number of Orders
0         Consumer             93504
1        Corporate             54789
2      Home Office             32226
```



Observations: Maxumum numbers of customer is Individual customer. Which is approximately 52%

7. Which category generates the most profit?

```
p2 = df.groupby('Category Name')['Benefit per order'].sum().sort_values(ascending = False).head(10)

plt.figure(figsize=(10,7))
plt.bar(p2.index, p2, fc='green', ec='black', alpha = 0.6)
plt.title('Profit by Category')
for i in range (len(p2)) :
    plt.text(i, p2[i], round(p2[i], 2), ha='center', va='bottom')

plt.xticks(rotation=45)
plt.tight_layout()
```



Observations: Fishing is the most profitable category.

8. Which payment methods customer prefer most?

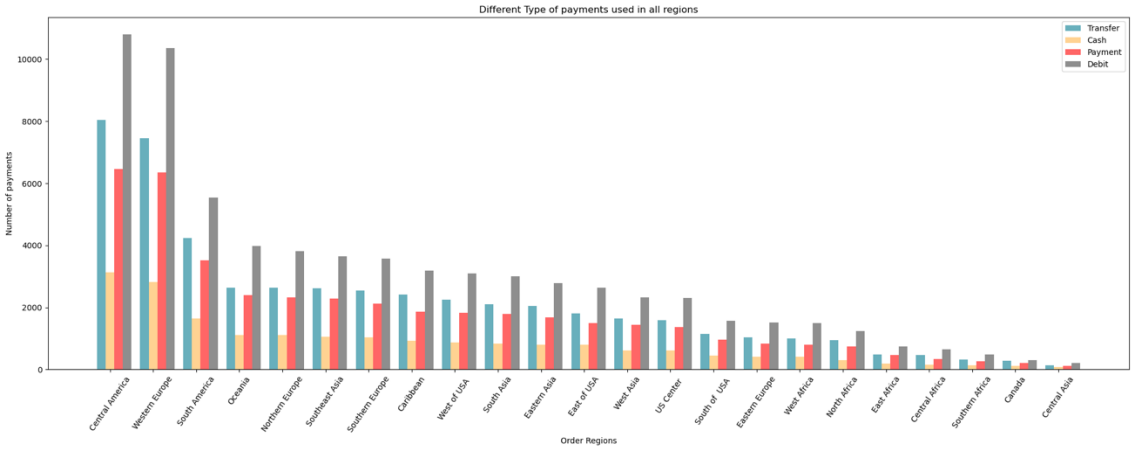
```
data_payment_preference = df.groupby(['Type'])['Order Id'].count().reset_index(name='Number of Orders').sort_values(
print(data_payment_preference)
```

```
       Type  Number of Orders
1     DEBIT             69295
3  TRANSFER             49883
2   PAYMENT             41725
0      CASH             19616
```

Observations: Debit card payment customers like most.

9. Which payment method is preferred the most by people in different regions?

```
count1 = df[(df['Type'] == 'TRANSFER')]['Order Region'].value_counts() # Total numbers of orders in TRANSFER payment
count2 = df[(df['Type'] == 'CASH')]['Order Region'].value_counts() # Total numbers of orders in CASH payment type in
count3 = df[(df['Type'] == 'PAYMENT')]['Order Region'].value_counts() # Total numbers of orders in PAYMENT payment t
count4 = df[(df['Type'] == 'DEBIT')]['Order Region'].value_counts() # Total numbers of orders in DEBIT payment type
names = df['Order Region'].value_counts().keys() # Regions name


n_groups = 23 # Show 23 result
fig,ax = plt.subplots(figsize=(20,8))
index = np.arange(n_groups)

bar_width=0.2
opacity=0.6

type1 = plt.bar(index, count1, bar_width, alpha=opacity, color='#037A90', label='Transfer') # x, y, opacity, color,l
type2 = plt.bar(index+bar_width, count2, bar_width, alpha=opacity, color='#FFB449', label='Cash')
type3 = plt.bar(index+bar_width+bar_width, count3, bar_width, alpha=opacity, color='#F00', label='Payment')
type4 = plt.bar(index+bar_width+bar_width+bar_width, count4, bar_width, alpha=opacity, color='#434343', label='Debit

plt.xlabel('Order Regions')
plt.ylabel('Number of payments')
plt.title('Different Type of payments used in all regions')
plt.legend()
plt.xticks(index+bar_width,names,rotation=55)
plt.tight_layout()
plt.show()
```

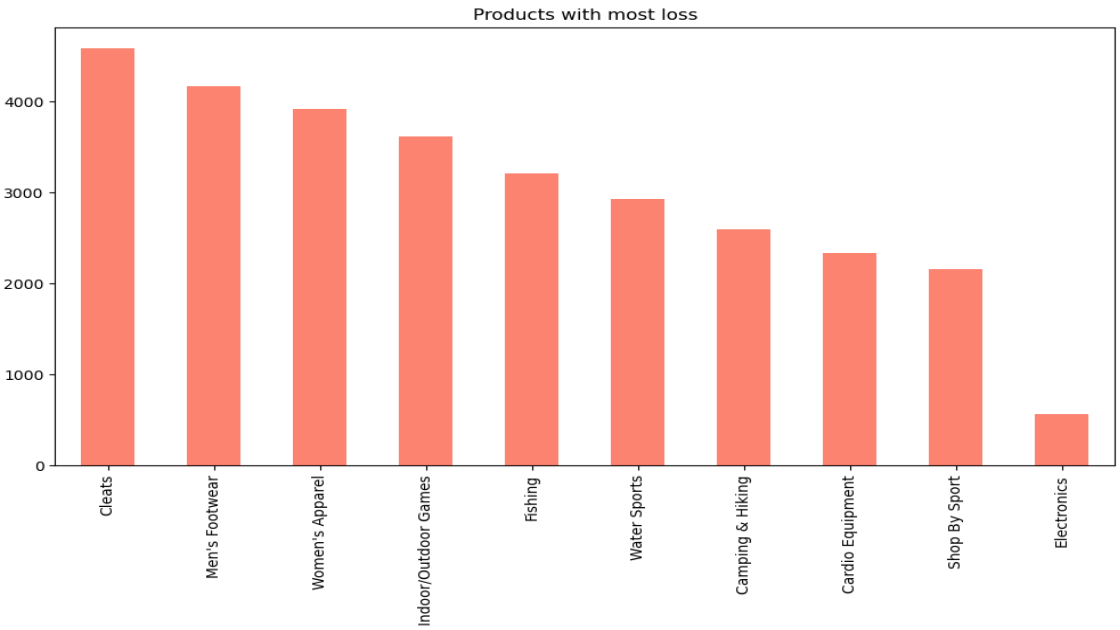


Observations:

- **Debit** type is most preferred by customer in all regions.
- **Cash** payment being the least preferred method by customers.

10. Which category products made most loss?

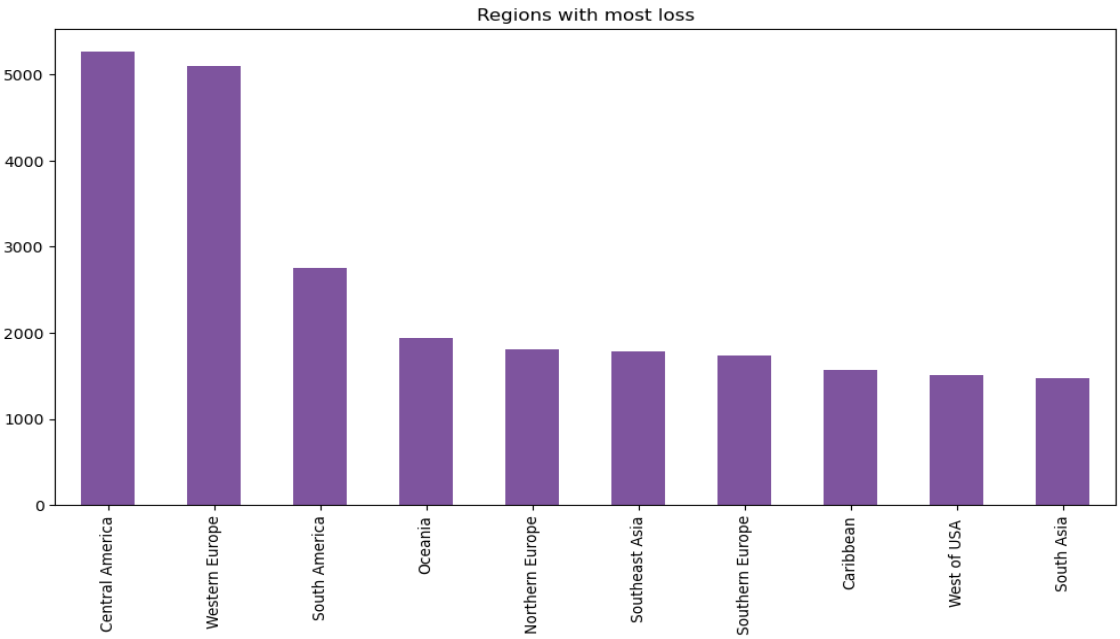
```
# Plotting top 10 category with most loss
loss['Category Name'].value_counts().nlargest(10).plot.bar(figsize=(12,6), title="Products with most loss", color="#
```



Observations: From the graph we can say **Cleats** category made heighest loss.

11.Which region made most loss?

```
# Plotting top 10 regions
loss['Order Region'].value_counts().nlargest(10).plot.bar(figsize=(12,6), title="Regions with most loss", color="#7e
```



Observations: From the above graph it clear that order from **Central America** resgion made most loss

It's possible that either suspected fraud or late deliveries caused these lost sales. It can be useful to prevent fraud in the future to identify the fraudulent payment method.

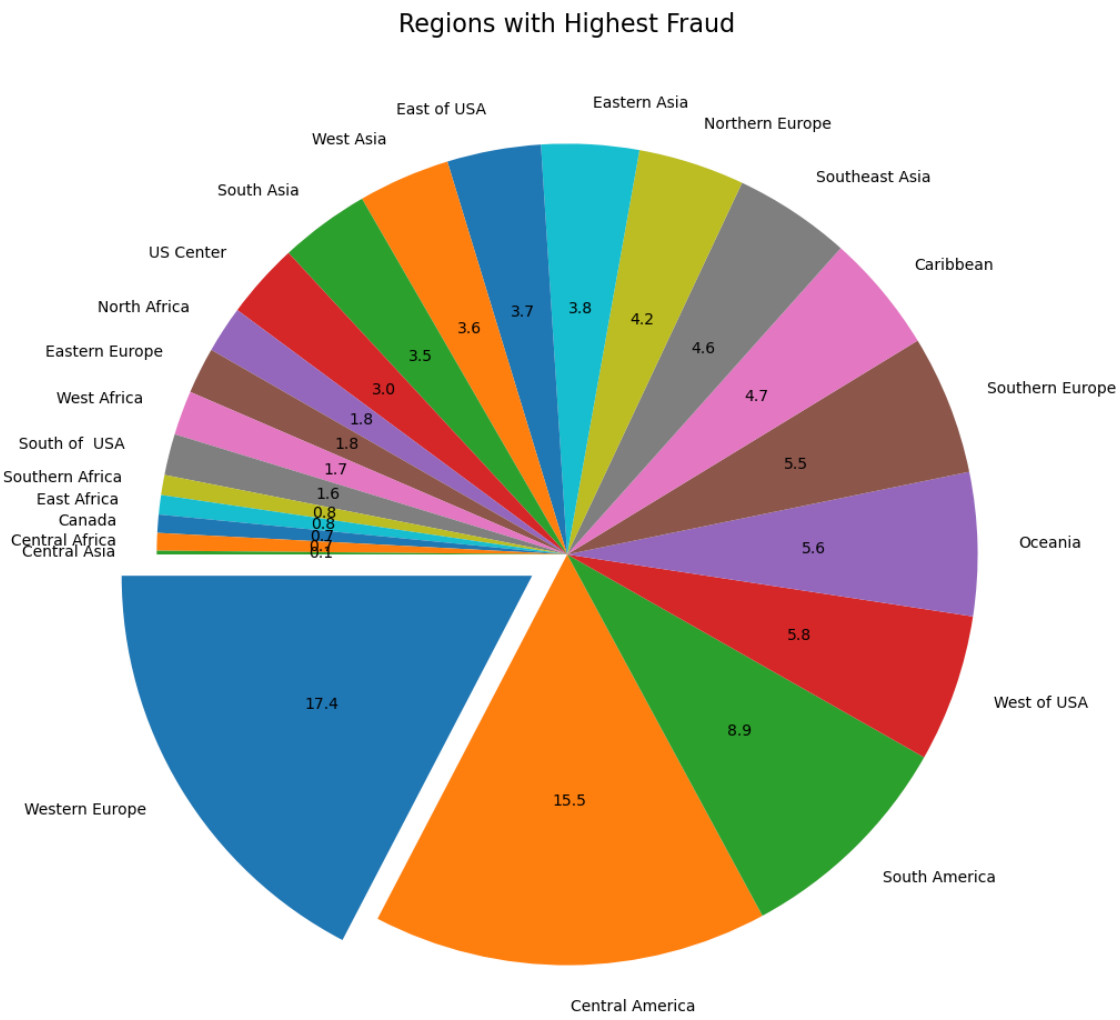
```
data_fraud_num = df[df['Order Status'] == 'SUSPECTED_FRAUD'].groupby(['Type'])['Order Status'].count().reset_index(n
print(data_fraud_num)
```

```
       Type  Number of Fraud
0  TRANSFER             4062
```

Observations: We can clearly see that there are no frauds conducted with DEBIT,CASH,PAYMENT methods. All the suspected fraud orders are made using **Transfer** payment method.

12. Which region being suspected to the fraud the most?

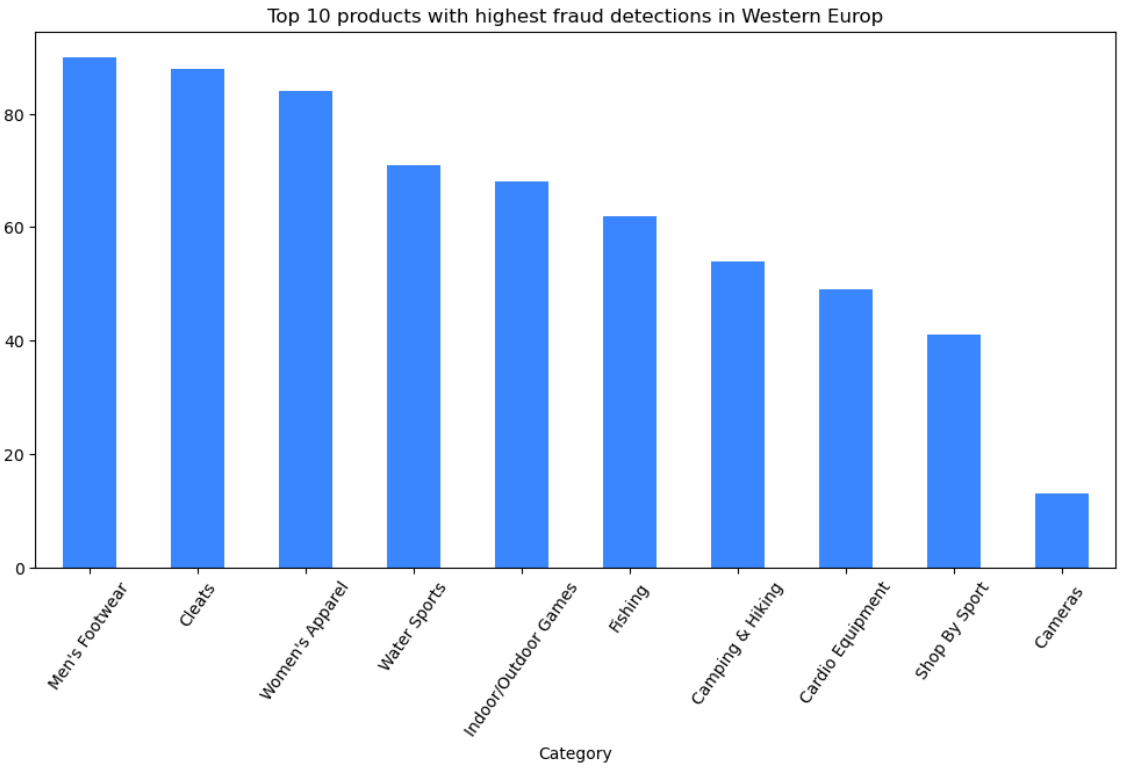
```
fraud_orders = df[(df['Order Status'] == 'SUSPECTED_FRAUD') & (df['Type'] == 'TRANSFER')] # separating orders with s

# Plotting pie chart with respect to order region
fraud = fraud_orders['Order Region'].value_counts().plot.pie(figsize=(24,12), startangle=180, explode=(0.1,0,0,0,0,0
plt.title("Regions with Highest Fraud",size=16) # Plotting title
plt.ylabel("")
plt.show()
```



Observations: It is evident that **Western Europe** accounts for approximately 17.4 percent of all orders suspected of fraud, followed by **Central America** with 15.5%

13. Which product is most frequently thought to be fraudulent in Western Europe?

```
# Plotting bar chart for top 10 most suspected fraud department in Western Europe
df[(df['Order Status'] == 'SUSPECTED_FRAUD') &(df['Order Region'] == 'Western Europe')]['Category Name'].value_count
plt.title("Top 10 products with highest fraud detections in Western Europ")
plt.xlabel("Category")
plt.xticks(rotation=55)
plt.show()
```

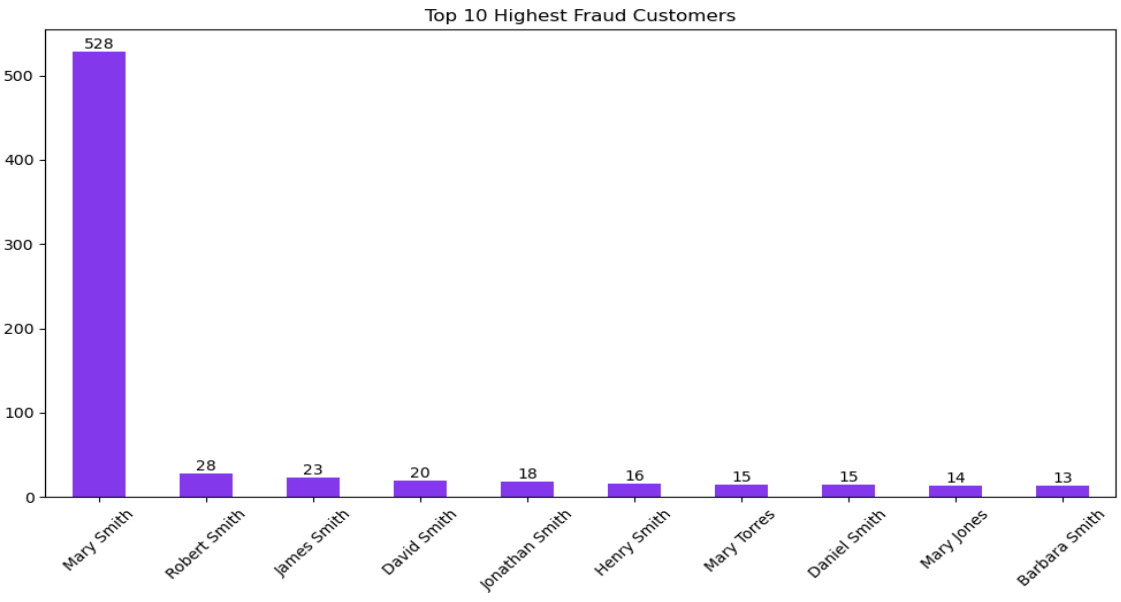


Observations: We can see that **cleats** department is being suspected to fraud the most followed by **Men's footwear** in Western Europe.

Let's detect which customers are conducting all these fraud.

14. Which customers are conducting all these fraud?

```
frud_customer = df[(df['Order Status'] == 'SUSPECTED_FRAUD')]
# Top 10 customers with most fraud
ax = frud_customer['Customer Name'].value_counts().nlargest(10).plot.bar(figsize=(12,6), title="Top 10 Highest Fraud

for container in ax.containers:
    ax.bar_label(container) # showing values on bar

plt.xticks(rotation=45)
plt.show()
```

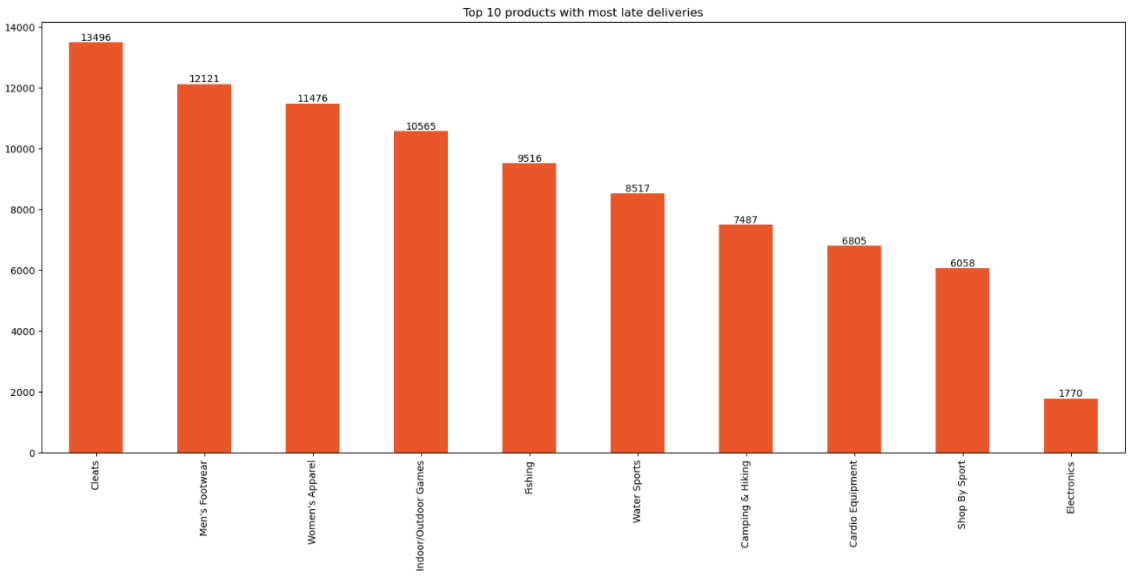


Observations: We can see that **cleats** department is being suspected to fraud the most followed by **Men's footwear** in Western Europe.

Let's detect which customers are conducting all these fraud.

15. Which category of products are being delivered the most late?

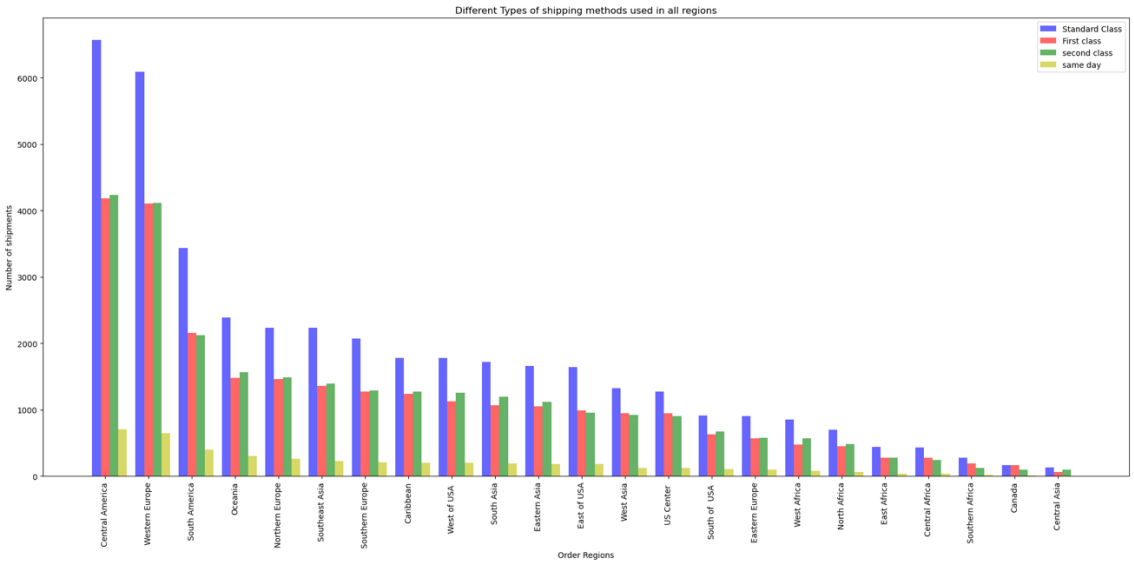
```
late_delivery = df[(df['Delivery Status'] == 'Late delivery')]

# Top 10 products with most late deliveries
ax = late_delivery['Category Name'].value_counts().nlargest(10).plot.bar(figsize=(20,8), title="Top 10 products with

for container in ax.containers:
    ax.bar_label(container) # showing values on bar
```



Observations: It can be seen that orders with **Cleats** department is getting delayed the most followed by **Men's Footwear**.

16. Late deliveried orders for different types of shipment method in all regions

```
# Counting total values
count1 = df[(df['Delivery Status'] == 'Late delivery') & (df['Shipping Mode'] == 'Standard Class')]['Order Region'].
count2 = df[(df['Delivery Status'] == 'Late delivery') & (df['Shipping Mode'] == 'First Class')]['Order Region'].val
count3 = df[(df['Delivery Status'] == 'Late delivery') & (df['Shipping Mode'] == 'Second Class')]['Order Region'].va
count4 = df[(df['Delivery Status'] == 'Late delivery') & (df['Shipping Mode'] == 'Same Day')]['Order Region'].value_

# Index names
names = df['Order Region'].value_counts().keys()
n_groups=23
fig,ax = plt.subplots(figsize=(20,10))
index = np.arange(n_groups)
bar_width = 0.2
opacity = 0.6
type1 = plt.bar(index,count1,bar_width,alpha=opacity,color='b',label='Standard Class')
type2 = plt.bar(index+bar_width,count2,bar_width,alpha=opacity,color='r',label='First class')
type3 = plt.bar(index+bar_width+bar_width,count3,bar_width,alpha=opacity,color='g',label='second class')
type4 = plt.bar(index+bar_width+bar_width+bar_width,count4,bar_width,alpha=opacity,color='y',label='same day')

plt.xlabel('Order Regions')
plt.ylabel('Number of shipments')
plt.title('Different Types of shipping methods used in all regions')
plt.legend()
plt.xticks(index+bar_width,names,rotation=90)
plt.tight_layout()
plt.show()
```



Observations:

- The most number of late deliveries for all regions occured with **standard** class shipping.
- In **same day** shipping being the one with least number of late deliveries.
- Both the **first class** and **second class** shipping have almost equal number of late deliveries.

```
data_cus_order = df.groupby(['Customer Segment'])['Order Id'].count().reset_index(name='Numbers of Order').sort_valu
data_cus_sales = df.groupby(['Customer Segment'])['Sales'].sum().reset_index(name='Sum of Sales').sort_values(by='Su

df2 = data_cus_order.merge(data_cus_sales, on="Customer Segment")
df2["Avg Sale"] = (df2["Sum of Sales"]/df2["Numbers of Order"])
print(df2)
```

```
  Customer Segment  Numbers of Order  Sum of Sales    Avg Sale
0         Consumer             93504  1.909579e+07  204.224313
1        Corporate             54789  1.116841e+07  203.843962
2      Home Office             32226  6.520538e+06  202.337802
```

Observations: Individual customers avg purchase rate is higher

Bivariate Analysis

1. Is there any relation between Sales & Product Price?

```
df.plot(x='Order Item Product Price', y='Sales per customer', linestyle='dotted', markerfacecolor='blue', markersize
plt.title('Product Price vs Sales per customer')
plt.xlabel('Order Item Product Price')
plt.ylabel('Sales per customer')
plt.show()
```

Observations: Above the graph it observed that prices has linear relation with sales

2. Every year what average sales in each month is?

```
df["Year"] = df["order date (DateOrders)"].dt.year
df["Month"] = df["order date (DateOrders)"].dt.month
df.head()

pivot_data = df.pivot_table(values="Sales", index="Month", columns="Year")
pivot_data
```

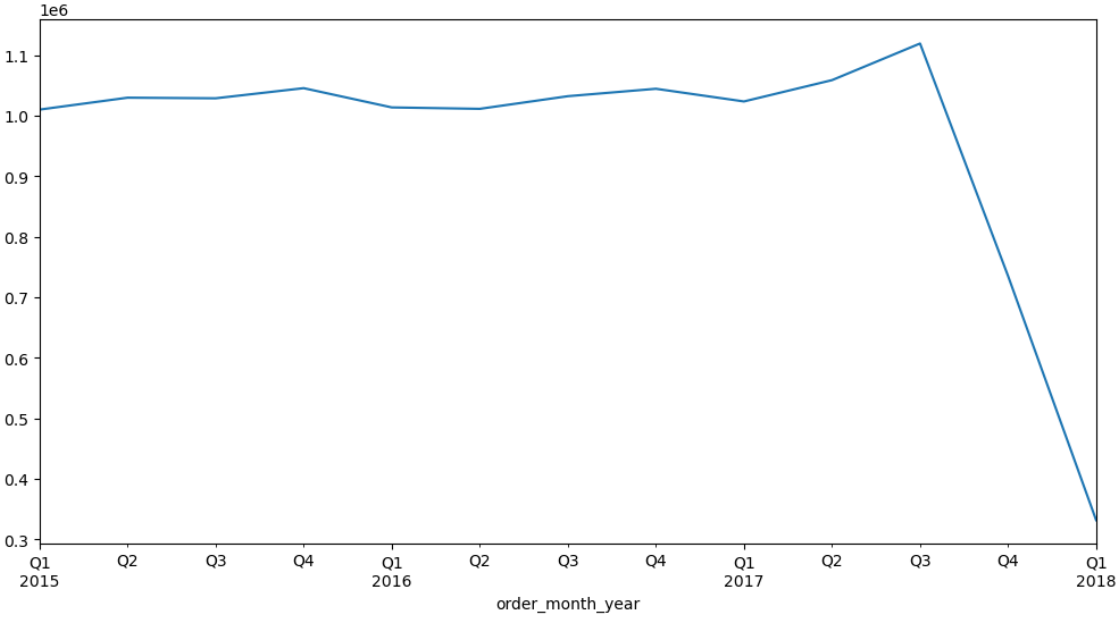
Year	2015	2016	2017	2018
Month				
1	197.593025	196.785452	197.373590	156.217671
2	196.026623	197.904138	202.310416	NaN
3	196.056265	196.900791	195.998650	NaN
4	197.905439	196.431544	199.217502	NaN
5	196.094538	194.153188	207.915238	NaN
6	199.455818	198.468451	208.460208	NaN
7	195.901338	197.118873	207.667048	NaN
8	195.238894	196.512983	209.111625	NaN
9	198.120350	194.262992	220.423031	NaN
10	197.878965	194.170165	476.272359	NaN
11	196.584573	201.072964	305.067827	NaN
12	196.954176	196.889006	237.246148	NaN

3. Which quarter recorded highest sales?

To better observe the trend, it can be found by dividing time into years, months, weeks, and hour

```
df['order_year']= df["order date (DateOrders)"].dt.year
df['order_month'] = df["order date (DateOrders)"].dt.month
df['order_week_day'] = df["order date (DateOrders)"].dt.weekday # Start from Monday
df['order_hour'] = df["order date (DateOrders)"].dt.hour
df['order_month_year'] = df["order date (DateOrders)"].dt.to_period('M')
```

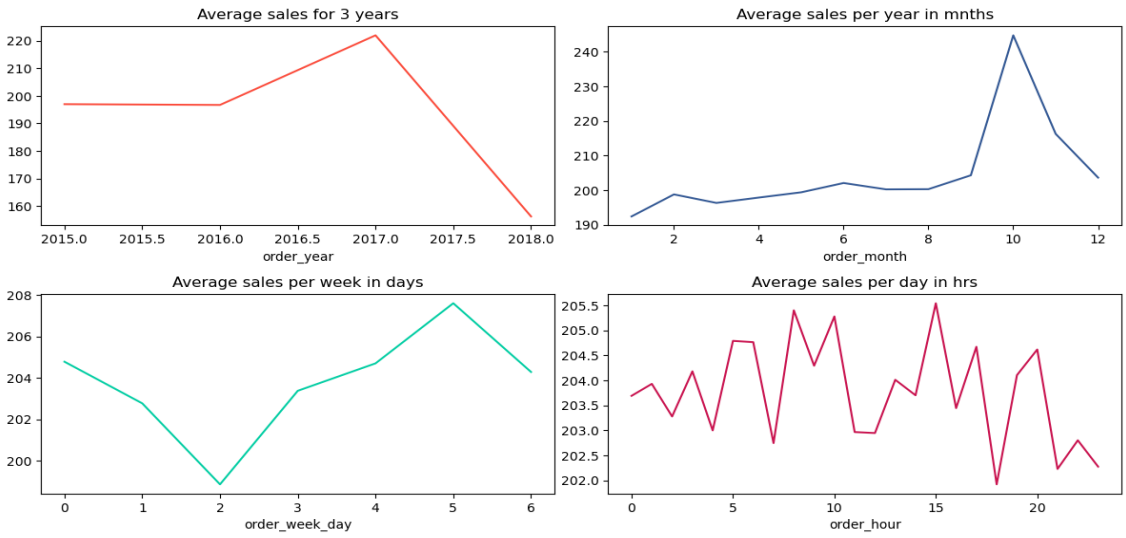
```
quater = df.groupby('order_month_year')
quartersales = quater['Sales'].sum().resample('Q').mean().plot(figsize=(12,6))
```



Obseervvations: From the above graph it seen that sales are consistent from Q1(1st quarter) of 2015 until Q3(3rd quarter) of 2017 and suddenly dipped by Q1(1st quarter) of 2018

4. What is the average sells in years, month, weeks and hours?

```
plt.figure(figsize=(10,12))
plt.subplot(4, 2, 1) #row, column, index
quater = df.groupby('order_year')
quater['Sales'].mean().plot(figsize=(12,12),title='Average sales for 3 years',color="#fb4d3d")

plt.subplot(4, 2, 2) #row, column, index
mnth = df.groupby("order_month")
mnth['Sales'].mean().plot(figsize=(12,12),title='Average sales per year in mnths',color="#345995")

plt.subplot(4, 2, 3) #row, column, index
days = df.groupby("order_week_day")
days['Sales'].mean().plot(figsize=(12,12),title='Average sales per week in days',color="#03cea4")

plt.subplot(4, 2, 4) #row, column, index
hrs = df.groupby("order_hour")
hrs['Sales'].mean().plot(figsize=(12,12),title='Average sales per day in hrs',color="#ca1551")
plt.tight_layout()
plt.show()
```



Observations:

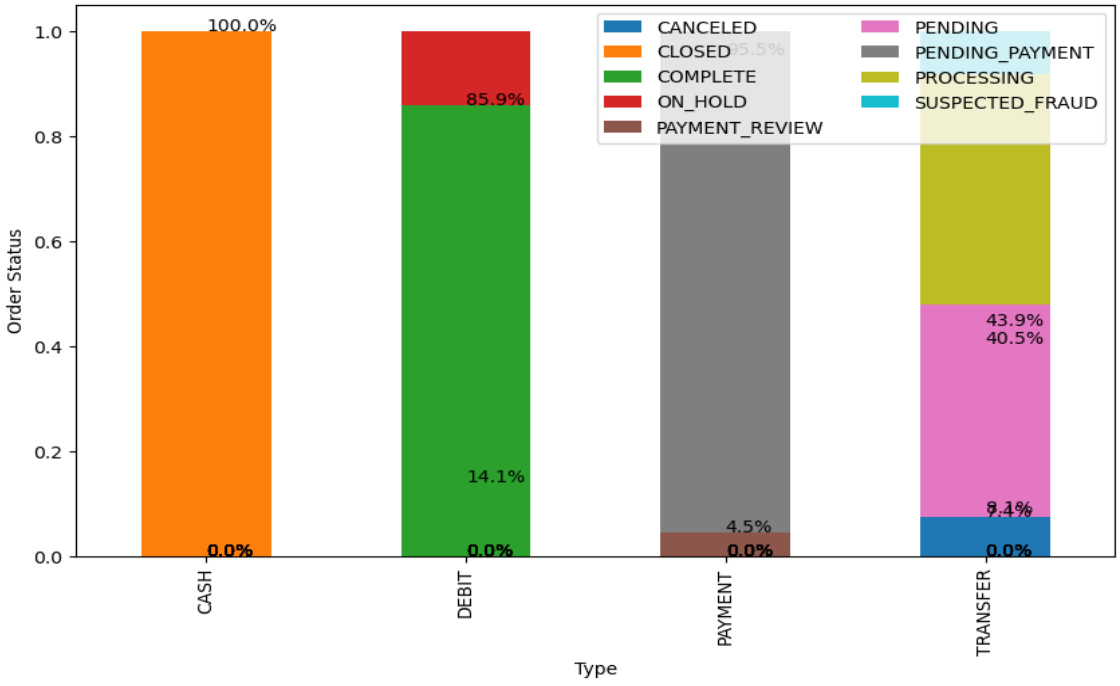
- In 2017 there was heightest numbers of orders are placed by customers.
- In **Saturday** recorded highest number of average sales and **wednesday** with the least number of sales.
- In **October** the most number of orders came followed by November.
- Daily average sales remain constant regardless of time.

5. Check the frequency respect to Payment Type & Order Status

```
# Crosstabs - Categorical to Categorical variable
cross_tab_prop = pd.crosstab(index=df['Type'], columns = df['Order Status'], normalize = "index")
cross_tab_prop
```

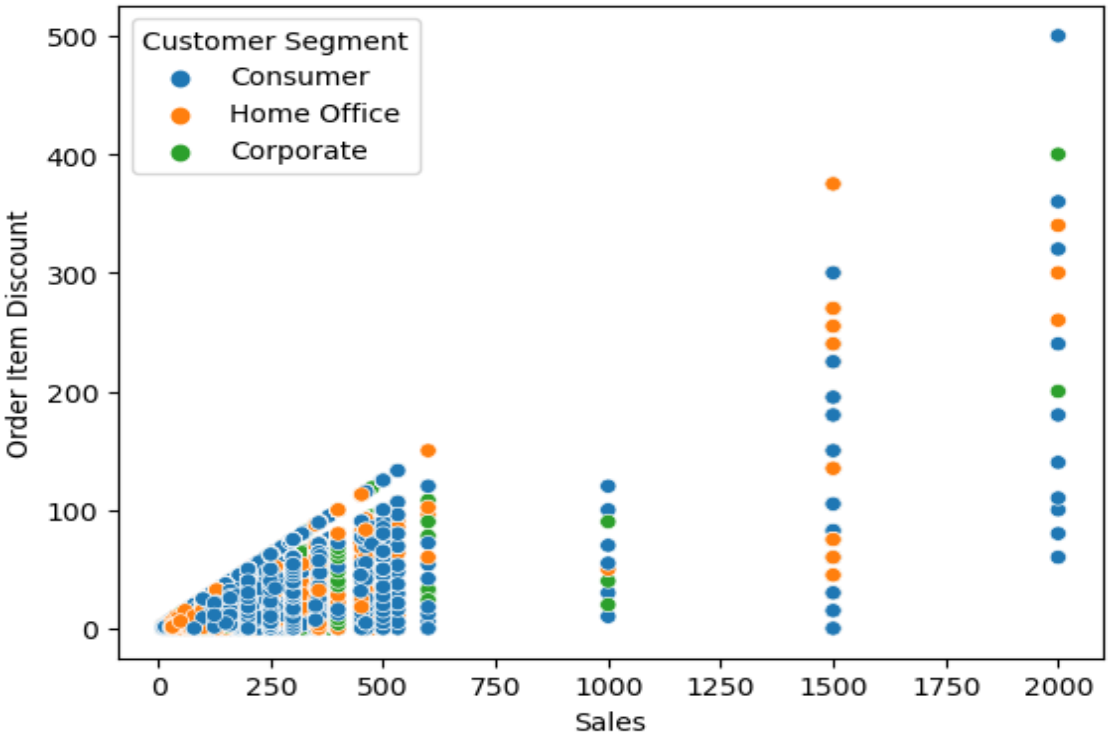
Order Status	CANCELED	CLOSED	COMPLETE	ON_HOLD	PAYMENT_REVIEW	PENDING	PENDING_PAYMENT	PROCESSING	SUSPECTED_FRAUD
Type									
CASH	0.000000	1.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DEBIT	0.000000	0.0	0.858518	0.141482	0.000000	0.000000	0.000000	0.000000	0.000000
PAYMENT	0.000000	0.0	0.000000	0.000000	0.045368	0.000000	0.954632	0.000000	0.000000
TRANSFER	0.074013	0.0	0.000000	0.000000	0.000000	0.405489	0.000000	0.439067	0.081431

```
cross_tab_prop.plot(kind='bar', stacked=True, colormap='tab10', figsize=(10, 6))

plt.legend(loc="upper right", ncol=2)
plt.xlabel("Type")
plt.ylabel("Order Status")
for n, x in enumerate([*cross_tab_prop.index.values]):
    for l in cross_tab_prop.loc[x]:
        plt.text(x=n, y=l, s=f'{np.round(l * 100, 1)}%', color="black")
plt.show()
```



6. Check whether there is any relation between Discount & Sales

```
# Scatterplot - Numerical to Numerical
sns.scatterplot(data=df, x="Sales", y="Order Item Discount", hue="Customer Segment")
```



Observations: It is positive correlation, So, we can conclude that as Discount increased the sales also increased

Some products have negative benefit per order, which indicating that the orders are lose in the business.

```
loss = df[(df['Benefit per order']<0)] # taking values which are less than 0
print('Total revenue lost with orders', loss['Benefit per order'].sum())
```

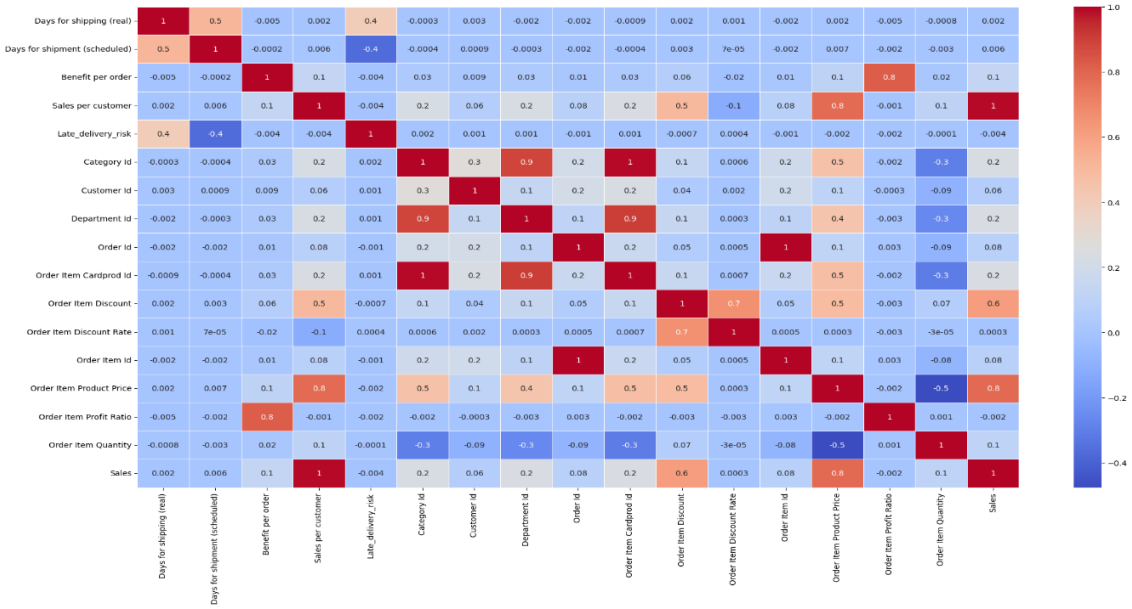
```
Total revenue lost with orders -3883547.345768667
```

Observations: Huge amount of loss are happened which are approximately 3.9 Millions.

Multivariate Analysis

1. Show relationships between two variables

```
# HeatMap - How one variable is moving with respect ot another variable
fig, ax = plt.subplots(figsize=(24,12))
sns.heatmap(df.corr(), annot=True, linewidths=.5, fmt=".1g", cmap="coolwarm")
plt.show()
```

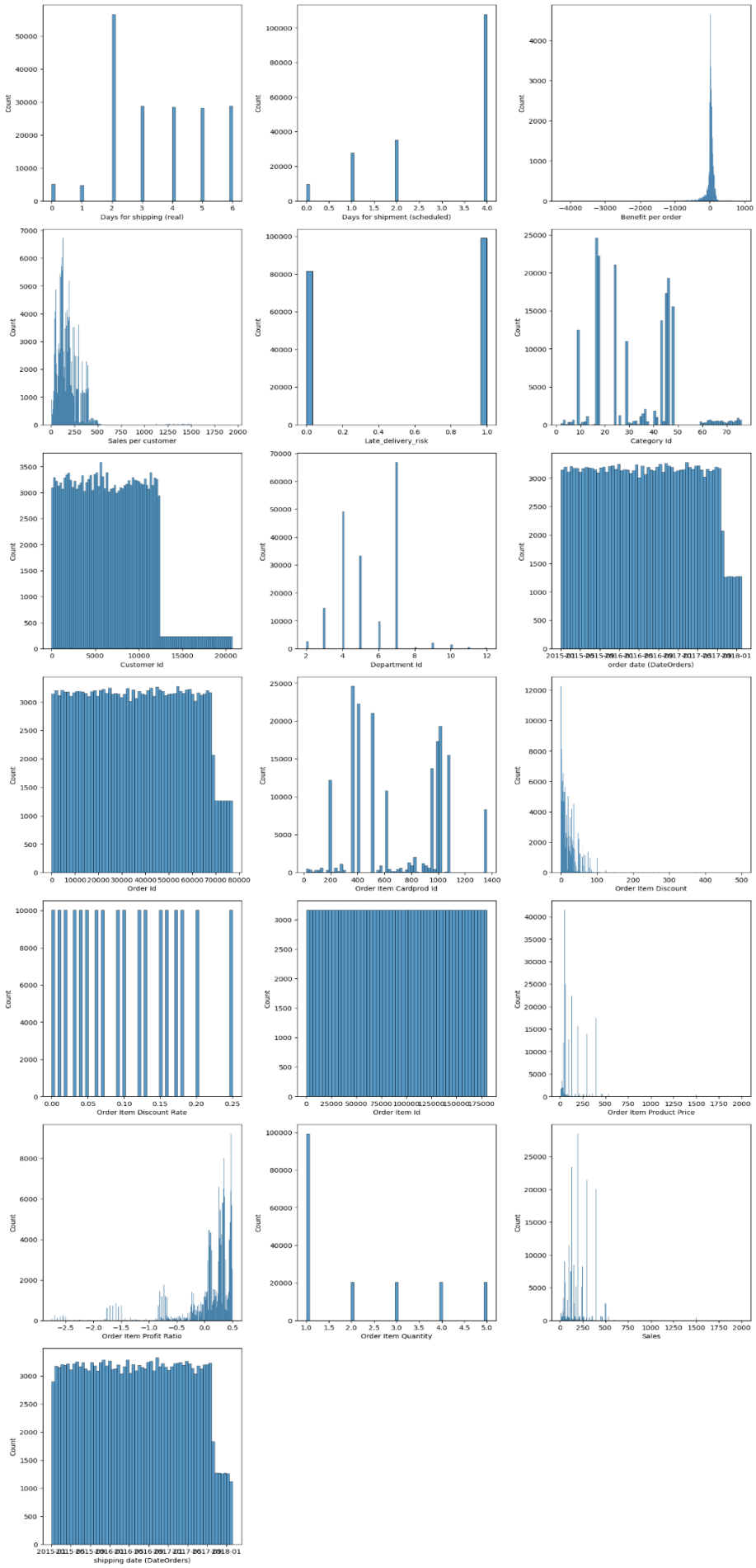


Observations: ***Highly correlation***

- Sales per customer & Sales
- Category id & Department id
- Category id & Order item cardprod id
- Department id & Category id
- Department id & Order item cartprod id

2. Histplot for all columns exclude object columns type

```
cols = 3
rows = 15
num_cols = df.select_dtypes(exclude="object").columns

fig = plt.figure(figsize=(cols*5, rows*5))
for i, col in enumerate(num_cols):
    try:
        ax=fig.add_subplot(rows,cols,i+1)
        sns.histplot(x = df[col], ax = ax)
    except ValueError:
        continue
fig.tight_layout()
plt.show()
```


Count
Days for shipping (real)
Days for shipment (scheduled)
Benefit per order
Sales per customer
Late_delivery_risk
Category Id
Customer Id
Department Id
order date (DateOrders)
Order Id
Order Item Cardprod Id
Order Item Discount
Order Item Discount Rate
Order Item Id
Order Item Product Price
Order Item Profit Ratio
Order Item Quantity
Sales
shipping date (DateOrders)

Observations:

- The schedules delivery date 4 is higher
- The most of the order delivered within 2 days
- Most of the order have 0 benifits
- Most of the customer purchase amount is approx 125
- Most of the customer oder only 1 product

Predictive Analysis

1. Understanding Customer Needs

Understanding customer needs and targeting specific clusters of customers based on their needs, is one way for a supply chain company to increase the number of customers and generate more profits. Since the purchase history of customers is already available in the dataset, it can be used RFM(Recency, Frequency, and Monetary) analysis for customer segmentation. Because it uses numerical values to show customer novelty, frequency and financial values, and the output results are easy to interpret.

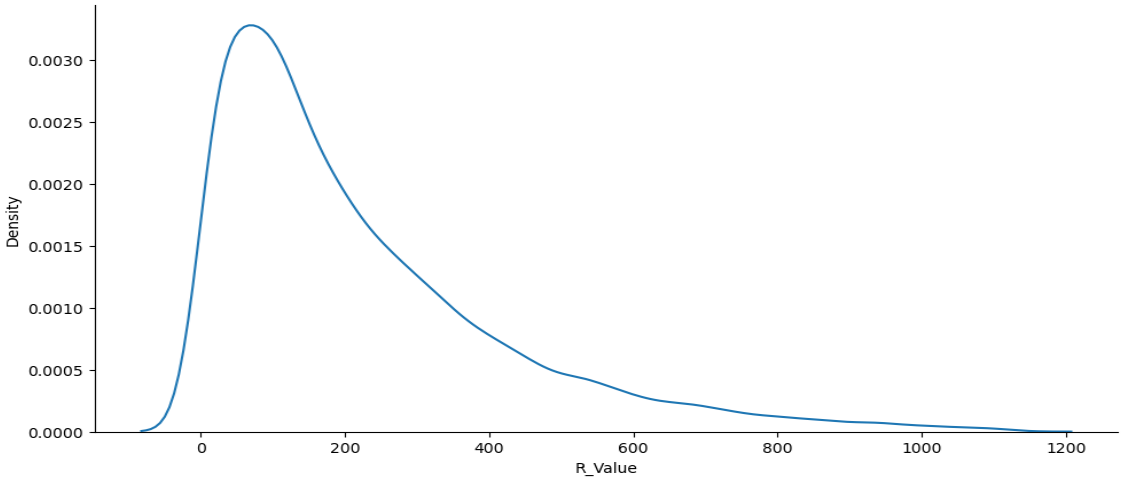
```
import datetime as dt

# Calculating total price for which each order
df['TotalPrice'] = df['Order Item Quantity'] * df['Sales'] # Multiplying item price * Order quantity

df['order date (DateOrders)'].max() # Calculating when the last order come to check recency

#Present date was set to next day of the last order. i.e,2018-02-01
present = dt.datetime(2018,2,1)
df['order date (DateOrders)'] = pd.to_datetime(df['order date (DateOrders)'])

Customer_seg = df.groupby('Customer Id').agg({'order date (DateOrders)': lambda x: (present - x.max()).days, 'Order
# Changing order dates to int format
Customer_seg['order date (DateOrders)'] = Customer_seg['order date (DateOrders)'].astype(int)
# Renaming columns as R_Value, F_Value, M_Value
Customer_seg.rename(columns={'order date (DateOrders)': 'R_Value', 'Order Id': 'F_Value', 'TotalPrice': 'M_Value'},
Customer_seg.head()
```

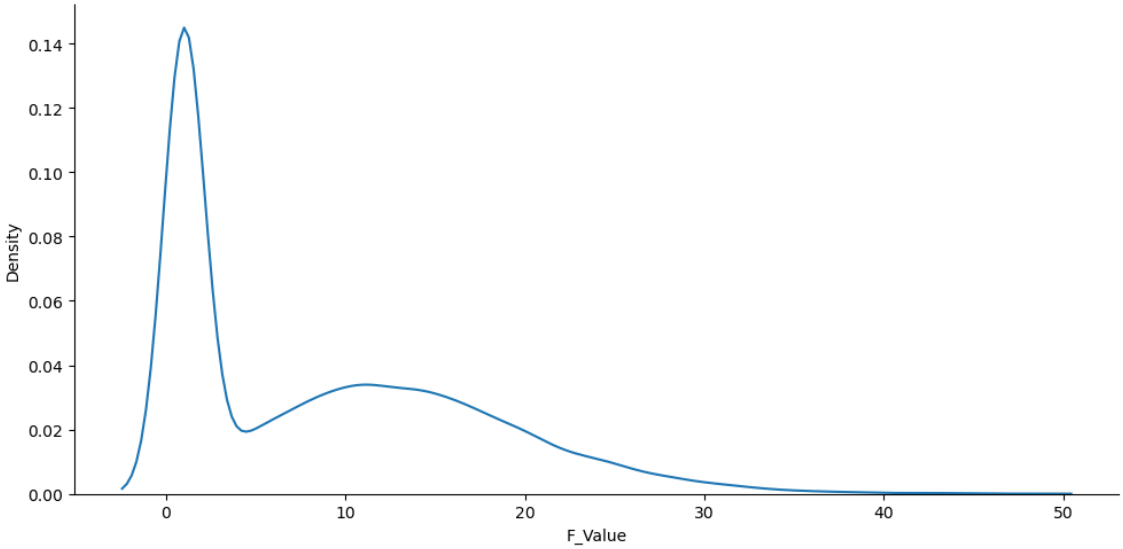
Customer Id	R_Value	F_Value	M_Value
1	792	1	2499.750061
2	136	10	3239.310028
3	229	18	6777.200182
4	380	14	5078.870088
5	457	7	3394.150024

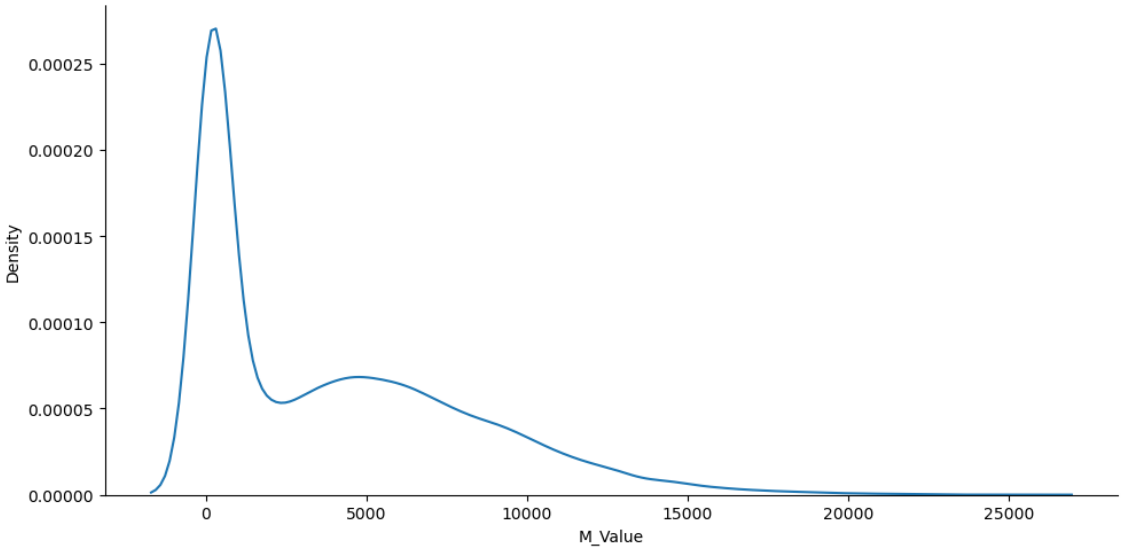
- R_Value(Recency) indicates how much time elapsed since a customer last order.
- F_Value(Frequency) indicates how many times a customer ordered.
- M_Value(Monetary value) tells us how much a customer has spent purchasing items

Let's plot on graph

```
sns.displot(data=Customer_seg, x="R_Value", kind="kde", aspect=2) #Plot distribution of R_Value
sns.displot(data=Customer_seg, x='F_Value', kind="kde", aspect=2) #Plot distribution of F_Value
sns.displot(data=Customer_seg, x='M_Value', kind="kde", aspect=2) #Plot distribution of M_Value
```




0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Density
0
10
20
30
40
50
F_Value


0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
Density
0
5000
10000
15000
20000
25000
M_Value

Hypothesis Testing

Null Hypothesis H0:

- The region to which the shipment is made does not influence late delivery
- The type of shipment does not influence late delivery
- The category of the product does not influence the late delivery

Alternative Hypothesis Ha:

- The region to which the shipment is made influences the late delivery
- The type of shipment influences late delivery
- Product category influences late delivery

```
def calculate_chi2(dependiente,independientes):
    for var in independientes:
        primary_location_cross = pd.crosstab(df[dependiente], df[var])
        g, p, dof, expctd = chi2_contingency(primary_location_cross)
        print("p-value of Chi-square test for " + dependiente + " vs " + var + " = " , p)

columns = ['Order Region','Shipping Mode','Category Name','Type','Customer City']
calculate_chi2('Delivery Status', columns)
```

```
p-value of Chi-square test for Delivery Status vs Order Region =  3.912852929033907e-23
p-value of Chi-square test for Delivery Status vs Shipping Mode =  0.0
p-value of Chi-square test for Delivery Status vs Category Name =  0.6712499177801518
p-value of Chi-square test for Delivery Status vs Type =  0.0
p-value of Chi-square test for Delivery Status vs Customer City =  0.0
```

Based on the previous hypothesis test in which the Chi2 distribution is used and the levels of significance between the different categorical variables were calculated, the following can be concluded:

- Reject null hypothesis 1
- Reject null hypothesis 2
- Accept the null hypothesis 3

Conclusion

After DataCo analyzed the company's dataset it discovered that Western Europe and Central America are both the regions with the highest number of sales, but the company lost the most revenue from these regions only. And both these regions have the highest number of fraudulent transactions and more late delivery orders are suspected. More than half of shoppers here shop in person. Then corporate and home office. The highest profit is from the fishing category. The company's total sales were consistent till 2017 quarter 3 and total sales increased by 10% quarter over quarter and then suddenly declined by almost 65% in 2018 quarter 1. October and November are the months with the highest sales in the total year. Saturdays are the most sold. Most people prefer to pay through debit card and all fraudulent transactions are happening with wire transfer so the company should be careful when customers are using wire transfer as the company has been scammed more than 528 times by a single customer. Products in the Cleats, Men's Shoes, and Women's Apparel categories are causing most orders to be delivered late, and these products are most likely to be suspected of fraud. In addition, hypothesis testing suggests that shipping class has a relationship with late delivery.

References

- https://www.geeksforgeeks.org/how-to-find-drop-duplicate-columns-in-a-pandas-dataframe/
- https://www.kaggle.com/code/sanketchavan5595/data-analysis-smart-supply-chain/data
- https://medium.com/datadriveninvestor/building-neural-network-using-keras-for-classification-3a3656c726c1
- https://medium.com/epfl-extension-school/advanced-exploratory-data-analysis-eda-with-python-536fa83c578a
- https://towardsdatascience.com/exploratory-data-analysis-eda-visualization-using-pandas-ca5a04271607
- https://matplotlib.org/stable/gallery/color/named_colors.html
- https://www.geeksforgeeks.org/how-to-find-drop-duplicate-columns-in-a-pandas-dataframe/