Shopify data challenge

Q1

a. Think about what could be going wrong with our calculation. Think about a better way to evaluate this data.

I believe it's highly likely that there are outliers, which could be due to frauds or log errors.

In [15]:
```
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:
```
table=pd.read_csv("A:/work/fulltime/SHOP/2019 Winter Data Science Intern Challenge Data Set - Sheet1.csv")
```

In [5]:
```
table.head()
```

Out[5]:

	order_id	shop_id	user_id	order_amount	total_items	payment_method	created_at
0	1	53	746	224	2	cash	2017-03-13 12:36:56
1	2	92	925	90	1	cash	2017-03-03 17:38:52
2	3	44	861	144	1	cash	2017-03-14 4:23:56
3	4	18	935	156	1	credit_card	2017-03-26 12:43:37
4	5	18	883	156	1	credit_card	2017-03-01 4:35:11

In [39]:
```
True in (table.isnull())
```

Out[39]:
```
False
```

In [6]:
```
table['order_amount'].mean()
```

Out[6]:
```
3145.128
```

In [18]:
```
table['order_amount'].describe()
```

Out[18]:
```
count     5000.000000
mean      3145.128000
std      41282.539349
min         90.000000
25%        163.000000
50%        284.000000
```

```
75%         390.000000
max      704000.000000
Name: order_amount, dtype: float64
```

In [27]:
```
table_user=table.groupby("user_id").agg({"order_amount":"mean"}).rename(
    columns={"order_amount":"mean_order_amount"}).sort_values(by=["mean_order_amount"])
table_user
```

Out[27]:

user_id	mean_order_amount
864	209.157895
939	219.600000
827	226.500000
899	226.666667
892	227.307692
...	...
915	5785.142857
834	6019.000000
766	8007.600000
878	14266.909091
607	704000.000000

301 rows × 1 columns

In [28]:
```
table_shop=table.groupby("shop_id").agg({"order_amount":"mean"}).rename(
    columns={"order_amount":"mean_order_amount"}).sort_values(by=["mean_order_amount"])
table_shop
```

Out[28]:

shop_id	mean_order_amount
92	162.857143
2	174.327273
32	189.976190
100	213.675000
53	214.117647
...	...
38	390.857143

shop_id	mean_order_amount
90	403.224490
50	403.545455
78	49213.043478
42	235101.490196

100 rows × 1 columns

We can see that there is no missing value but there are outliers when we calculate the mean_order_amount by user and shop respectively. In particular, user 607 and shop 42 and 78.

In [29]:
```
table[table["user_id"]==607]
```

Out[29]:

	order_id	shop_id	user_id	order_amount	total_items	payment_method	created_at
15	16	42	607	704000	2000	credit_card	2017-03-07 4:00:00
60	61	42	607	704000	2000	credit_card	2017-03-04 4:00:00
520	521	42	607	704000	2000	credit_card	2017-03-02 4:00:00
1104	1105	42	607	704000	2000	credit_card	2017-03-24 4:00:00
1362	1363	42	607	704000	2000	credit_card	2017-03-15 4:00:00
1436	1437	42	607	704000	2000	credit_card	2017-03-11 4:00:00
1562	1563	42	607	704000	2000	credit_card	2017-03-19 4:00:00
1602	1603	42	607	704000	2000	credit_card	2017-03-17 4:00:00
2153	2154	42	607	704000	2000	credit_card	2017-03-12 4:00:00
2297	2298	42	607	704000	2000	credit_card	2017-03-07 4:00:00
2835	2836	42	607	704000	2000	credit_card	2017-03-28 4:00:00
2969	2970	42	607	704000	2000	credit_card	2017-03-28 4:00:00
3332	3333	42	607	704000	2000	credit_card	2017-03-24 4:00:00
4056	4057	42	607	704000	2000	credit_card	2017-03-28 4:00:00
4646	4647	42	607	704000	2000	credit_card	2017-03-02 4:00:00
4868	4869	42	607	704000	2000	credit_card	2017-03-22 4:00:00
4882	4883	42	607	704000	2000	credit_card	2017-03-25 4:00:00

In [31]:
```
table[table["shop_id"]==78]
```

Out[31]:

	order_id	shop_id	user_id	order_amount	total_items	payment_method	created_at
160	161	78	990	25725	1	credit_card	2017-03-12 5:56:57
490	491	78	936	51450	2	debit	2017-03-26 17:08:19
493	494	78	983	51450	2	cash	2017-03-16 21:39:35
511	512	78	967	51450	2	cash	2017-03-09 7:23:14
617	618	78	760	51450	2	cash	2017-03-18 11:18:42
691	692	78	878	154350	6	debit	2017-03-27 22:51:43
1056	1057	78	800	25725	1	debit	2017-03-15 10:16:45
1193	1194	78	944	25725	1	debit	2017-03-16 16:38:26
1204	1205	78	970	25725	1	credit_card	2017-03-17 22:32:21
1259	1260	78	775	77175	3	credit_card	2017-03-27 9:27:20
1384	1385	78	867	25725	1	cash	2017-03-17 16:38:06
1419	1420	78	912	25725	1	cash	2017-03-30 12:23:43
1452	1453	78	812	25725	1	credit_card	2017-03-17 18:09:54
1529	1530	78	810	51450	2	cash	2017-03-29 7:12:01
2270	2271	78	855	25725	1	credit_card	2017-03-14 23:58:22
2452	2453	78	709	51450	2	cash	2017-03-27 11:04:04
2492	2493	78	834	102900	4	debit	2017-03-04 4:37:34
2495	2496	78	707	51450	2	cash	2017-03-26 4:38:52
2512	2513	78	935	51450	2	debit	2017-03-18 18:57:13
2548	2549	78	861	25725	1	cash	2017-03-17 19:36:00
2564	2565	78	915	77175	3	debit	2017-03-25 1:19:35
2690	2691	78	962	77175	3	debit	2017-03-22 7:33:25
2773	2774	78	890	25725	1	cash	2017-03-26 10:36:43
2818	2819	78	869	51450	2	debit	2017-03-17 6:25:51
2821	2822	78	814	51450	2	cash	2017-03-02 17:13:25
2906	2907	78	817	77175	3	debit	2017-03-16 3:45:46
2922	2923	78	740	25725	1	debit	2017-03-12 20:10:58
3085	3086	78	910	25725	1	cash	2017-03-26 1:59:27
3101	3102	78	855	51450	2	credit_card	2017-03-21 5:10:34
3151	3152	78	745	25725	1	credit_card	2017-03-18 13:13:07

	order_id	shop_id	user_id	order_amount	total_items	payment_method	created_at
3167	3168	78	927	51450	2	cash	2017-03-12 12:23:08
3403	3404	78	928	77175	3	debit	2017-03-16 9:45:05
3440	3441	78	982	25725	1	debit	2017-03-19 19:02:54
3705	3706	78	828	51450	2	credit_card	2017-03-14 20:43:15
3724	3725	78	766	77175	3	credit_card	2017-03-16 14:13:26
3780	3781	78	889	25725	1	cash	2017-03-11 21:14:50
4040	4041	78	852	25725	1	cash	2017-03-02 14:31:12
4079	4080	78	946	51450	2	cash	2017-03-20 21:14:00
4192	4193	78	787	77175	3	credit_card	2017-03-18 9:25:32
4311	4312	78	960	51450	2	debit	2017-03-01 3:02:10
4412	4413	78	756	51450	2	debit	2017-03-02 4:13:39
4420	4421	78	969	77175	3	debit	2017-03-09 15:21:35
4505	4506	78	866	25725	1	debit	2017-03-22 22:06:01
4584	4585	78	997	25725	1	cash	2017-03-25 21:48:44
4715	4716	78	818	77175	3	debit	2017-03-05 5:10:44
4918	4919	78	823	25725	1	cash	2017-03-15 13:26:46

As we can see from the above tables, all the transacrions by user 607 was in shop 42. It is highly likely a fraud user because all payments were made through credit cards within a month and the order amount were extremely large. Even though shop 78 has some large bills, we cannot say for sure that they were due to frauds. Because the majority of the payments were through debit cards or cash by different users and there exists limited edition of shoes that cost a lot. In conclusion, these outliers should be removed

b. What metric would you report for this dataset?

I will report the mean, the min, the max, the std and 25%,50% and 75% percentiles of the order_amount after removing the outliers.

c. What is its value?

In [35]:
```
table[(table["shop_id"]!=42)&(table["shop_id"]!=78)]["order_amount"].describe()
```

Out[35]:
```
count    4903.000000
mean      300.155823
std       155.941112
min        90.000000
25%       163.000000
50%       284.000000
75%       386.500000
```

```
max      1086.000000
Name: order_amount, dtype: float64
```

Q2

a. How many orders were shipped by Speedy Express in total?

Answer: 54

In [47]:
```
"""
SELECT COUNT(OrderID) FROM
Orders o INNER JOIN Shippers s
ON o.ShipperID = s.ShipperID
Where ShipperName = 'Speedy Express';
"""
```

Out[47]:
```
"\nSELECT COUNT(OrderID) FROM\nOrders o INNER JOIN Shippers s\nON o.ShipperID = s.ShipperID\nWhere ShipperName = 'Speedy Express';\n"
```

b. What is the last name of the employee with the most orders?

Anwser: Peacock, 40 orders

In [43]:
```
"""
SELECT LastName, COUNT(DISTINCT OrderID) AS NetOrders FROM
(SELECT o.OrderID, e.EmployeeID, e.LastName, e.FirstName
FROM Orders o INNER JOIN Employees e
ON o.EmployeeID = e.EmployeeID)
GROUP BY EmployeeID
ORDER BY COUNT(DISTINCT OrderID) DESC LIMIT 1;
"""
```

Out[43]:
```
'\nSELECT LastName, MAX(NetOrders) FROM\n(Select *, COUNT(DISTINCT OrderID) as NetOrders FROM\n(SELECT o.OrderID, e.EmployeeID, e.LastName, e.Fir
stName\nFROM Orders o Inner Join Employees e\nON o.EmployeeID = e.EmployeeID)\nGROUP BY EmployeeID\nORDER BY COUNT(DISTINCT OrderID) DESC)\n\n'
```

c. What product was ordered the most by customers in Germany?

Anwser: Gorgonzola Telino, 5 orders

In [46]:
```
"""
SELECT p.ProductName, t1.NetOrders FROM
(SELECT ProductID, NetOrders FROM
(SELECT *, COUNT(DISTINCT OrderID) AS NetOrders FROM
(SELECT * From Orders o INNER JOIN OrderDetails od
ON o.OrderID = od.OrderID
WHERE CustomerID IN
(SELECT CustomerID FROM
Customers
WHERE Country = 'Germany'))
GROUP BY ProductID
ORDER BY NetOrders DESC LIMIT 1))
```

```
t1, Products p
WHERE t1.ProductID = p.ProductID
"""
```

Out[46]:
```
"\nSELECT p.ProductName, t1.NetOrders FROM\n(SELECT ProductID, NetOrders FROM\n(SELECT *, COUNT(DISTINCT OrderID) AS NetOrders FROM\n(SELECT * Fr
om Orders o INNER JOIN OrderDetails od\nON o.OrderID = od.OrderID\nWHERE CustomerID IN\n(SELECT CustomerID FROM\nCustomers \nWHERE Country = 'Ger
many'))\nGROUP BY ProductID\nORDER BY NetOrders DESC LIMIT 1)) \nt1, Products p\nWHERE t1.ProductID = p.ProductID\n"
```

In [ ]: