Submitted by: Tanmay Birar

Q1

a. Getting the summary of the dataset, we find the mean order_amount to be 3145.

```
state = read.csv("/Users/tanmaybirar/Downloads/winter.csv")
library(tidyverse)
summary(state)
```

```
> summary(state)
    order_id       shop_id          user_id       order_amount
 Min.   :   1   Min.   :  1.00   Min.   :607.0   Min.   :    90
 1st Qu.:1251   1st Qu.: 24.00   1st Qu.:775.0   1st Qu.:   163
 Median :2500   Median : 50.00   Median :849.0   Median :   284
 Mean   :2500   Mean   : 50.08   Mean   :849.1   Mean   :  3145
 3rd Qu.:3750   3rd Qu.: 75.00   3rd Qu.:925.0   3rd Qu.:   390
 Max.   :5000   Max.   :100.00   Max.   :999.0   Max.   :704000
  total_items      payment_method      created_at
 Min.   :   1.000   Length:5000        Length:5000
 1st Qu.:   1.000   Class :character   Class :character
 Median :   2.000   Mode  :character   Mode  :character
 Mean   :   8.787
 3rd Qu.:   3.000
 Max.   :2000.000
```

The mean of 3145 happens to be unreasonable for a pair of sneakers.

b. To fix this, let us check the mean amounts.

```
state_2 = state[,c('shop_id', 'order_amount', 'total_items')]
state_2$mean_amount = state_2$order_amount / state_2$total_items
mean(state_2$mean_amount)
state_2 %>% arrange((mean_amount))
summary(state_2$mean_amount)
```

```
> summary(state_2$mean_amount)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   90.0   133.0   153.0   387.7   169.0 25725.0
>
```

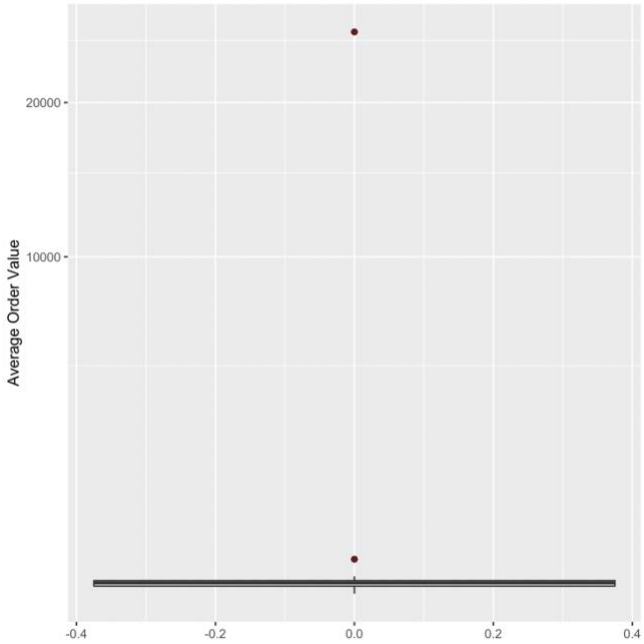
The highest mean_amount value happens to be 25725, which corresponds to shop_id 78. This is a likely outlier making influencing the mean significantly.

To validate the claim:

```
ggplot(state_2, aes(, mean_amount)) +
  ylab("Average Order Value") +
  geom_boxplot(outlier.colour = "#611f1f") +
```

Submitted by: Tanmay Birar

```
coord_trans(y="sqrt") +
scale_y_continuous(breaks = c(100, 200, 300, 500,5000, 2500, 10000, 15000, 20000))
```



The following boxplot hints towards the presence of two outliers.

One being an unreasonably high mean_amount corresponding to shop_id 78 and the other one corresponding to shop_id 42 likely.

Both these shops tend to deviate farther away from the mean amounts.

Solution: Exclude shop 78 and 42 from the analysis.

```
state_2 = subset(state, shop_id!=c(78,42))
summary(state_2)
```

```
    shop_id        order_amount     total_items       mean_amount
 Min.   :  1.00   Min.   :    90   Min.   :   1.000   Min.   :   90.0
 1st Qu.: 24.00   1st Qu.:   163   1st Qu.:   1.000   1st Qu.:  132.0
 Median : 50.00   Median :   284   Median :   2.000   Median :  153.0
 Mean   : 49.98   Mean   :  2230   Mean   :   6.841   Mean   :  259.9
 3rd Qu.: 75.00   3rd Qu.:   390   3rd Qu.:   3.000   3rd Qu.:  168.0
 Max.   :100.00   Max.   :704000   Max.   :2000.000   Max.   :25725.0
```

After having excluded shops 78 and 42, the analysis returns a more reasonable value of the mean which is 259. This however comes from the mean_amount column as the order_amount column by itself cannot be used for an accurate analysis.

c. As to the question about what metric to use, Median in this case should be the likely metric to be used while performing analyses in scenarios like this one.

Submitted by: Tanmay Birar

While mean is sensitive to outliers, median is relatively more robust. And from observation, the median order_amount remained 284 both before and after removal of the outliers. Hence, Median remains unaffected by extremities in the dataset.

Q2.

a. Ans: 54

```
SELECT ShipperName, COUNT(O.ShipperID)
FROM Orders O
INNER JOIN Shippers S ON O.ShipperID = S.ShipperID
WHERE S.ShipperName = "Speedy Express"
GROUP BY S.ShipperName;
```

Number of Records: 1

ShipperName	Expr1001
Speedy Express	54

b. Ans: 'Peacock'

```
SELECT TOP 1 E.LastName FROM Employees E
INNER JOIN Orders O on O.EmployeeID = E.EmployeeID
Group by E.LastName
Order by COUNT(O.EmployeeID) DESC;
```

Number of Records: 1

LastName
Peacock

c. Ans:

```
Select TOP 1 Products.ProductName, Customers.Country, COUNT(Orders.OrderID) AS Tot_Orders FROM
(((Customers
INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
INNER JOIN OrderDetails ON Orders.OrderID = Orderdetails.OrderID)
INNER JOIN Products ON Orderdetails.ProductID = Products.ProductID)
WHERE Country = 'Germany'
GROUP BY ProductName, Country
```

Submitted by: Tanmay Birar

```
ORDER BY COUNT(Orders.orderID) DESC;
```

Number of Records: 1

ProductName	Country	Tot_Orders
Gorgonzola Telino	Germany	5