Question 1

In [1]:
```
import pandas as pd
import numpy as np
df = pd.read_csv("shopify_data.csv")
```

In [2]:
```
# 100 unique shops
df['shop_id'].nunique()
```

Out[2]:
```
100
```

In [3]:
```
# sanity check --> data is over a 30 day window
print(min(df['created_at']))
print(max(df['created_at']))
```

```
2017-03-01 0:08:09
2017-03-30 9:55:00
```

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

In [4]:
```
# calculate Average Order Value (AOV)
AOV = df['order_amount'].mean()
print(AOV)
```

```
3145.128
```

In [5]:
```
# outliers
from scipy import stats
order_amounts = df['order_amount'].to_frame()
print(order_amounts[(np.abs(stats.zscore(order_amounts)) > 3).all(axis=1)])
#print(order_amounts[(np.abs(stats.zscore(order_amounts)) < -3).all(axis=1)])
```

```
      order_amount
15          704000
60          704000
520         704000
691         154350
1104        704000
1362        704000
1436        704000
1562        704000
1602        704000
2153        704000
2297        704000
2835        704000
2969        704000
3332        704000
4056        704000
4646        704000
4868        704000
4882        704000
```

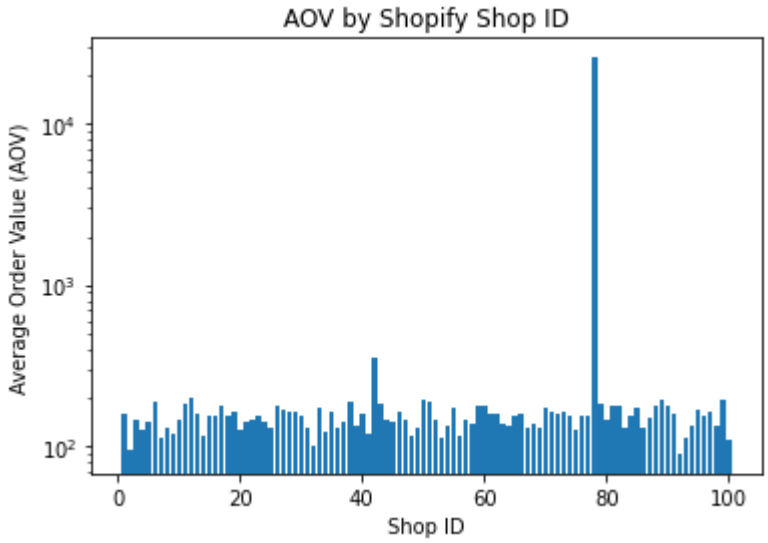
The problem with the average order value calculation is that the total items (i.e. quantity) of Sneaker sold at each location is different. If, for example, there are multiple large orders our data will be skewed right and we will get an absurdly large average order amount (i.e. AOV). This is likely due to the fact that the number of orders and the order amount is not the same across each of the 100 stores in our data. We create a histogram below, with the y axis being log scaled so we can visualize the outliers that exist in our data.

In [6]:
```
import matplotlib.pyplot as plt
# make y-axis log is scaled to better visualize data of varying orders of magnitude
plt.hist(df['order_amount'], log = True);
plt.xlabel('order_amount');
plt.ylabel('frequency');
plt.title('Order Amounts of Shopify Customers');
```

There appear to be a number of larger order amounts that are contributing to our abnormally high AOV. Let us see how AOV fluctuates across shops

In [7]:
```
# Let us investigate how AOV fluctuates on a shop-by-shop basis
aov_series = df.groupby(['shop_id'])['order_amount'].agg('sum') / df.groupby(['shop_id'
aov_by_shop = aov_series.to_frame(name = 'AOV')
```

In [8]:
```
import matplotlib.pyplot as plt
# make y-axis log is scaled to better visualize data of varying orders of magnitude
plt.bar(aov_by_shop.index, aov_by_shop['AOV'], log = True);
plt.xlabel('Shop ID');
plt.ylabel('Average Order Value (AOV)');
plt.title('AOV by Shopify Shop ID');
```



In [9]:
```
# outliers
from scipy import stats
print(aov_by_shop[(np.abs(stats.zscore(aov_by_shop)) > 3).all(axis=1)])
#print(aov_by_shop[(np.abs(stats.zscore(aov_by_shop)) < -3).all(axis=1)])
```

```
             AOV
shop_id
78       25725.0
```

It appears that shop ID 78 is an outlier, and has the most anomalous (i.e. highest) AOV.

b. What metric would you report for this dataset?

In order to better evaluate the data I would look at the **Median Average Order Value Amount across shops**. In order to achieve this I would aggregate the AOVs on a shop-by-shop basis since we know order amounts can fluctuate wildly across different shops.

c. What is its value?

In [10]:
```
# When we calculate the median AOV across ALL the shops in our dataset we arrive at a m
aov_by_shop['AOV'].median()
```

Out[10]:
```
153.0
```

Question 2

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

```
SELECT count(*)
FROM Shippers s
JOIN Orders o on s.ShipperID = o.ShipperID
WHERE s.ShipperName = 'Speedy Express';
```

In total, 54 Orders were shipped by Speedy Express

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

```
SELECT e.LastName
FROM Orders o
JOIN Employees e
ON o.EmployeeID = e.EmployeeID
GROUP BY e.LastName
ORDER BY COUNT(*)
DESC LIMIT 1
```

The last name of the employee with the most orders is Peacock

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

```
SELECT p.productName
FROM products p
WHERE p.productID =
(SELECT productID FROM
(SELECT * FROM Orders o JOIN customers c ON o.CustomerID = c.CustomerID WHERE
c.Country = 'Germany') q
 JOIN orderDetails od on od.OrderID = q.OrderID)
GROUP BY p.productID
```

Boston Crab Meat was ordered most by customers in Germany