Motorcycle Parts

February 3, 2022

0.1 Sales Report

by Scott Schmidt

Three warehouses in a large metropolitan area that sells motorcycle parts. The following sales report will analyze past sales data such as the following: 1. What are the total sales for each payment method? 2. What is the average unit price for each product line? 3. Create plots to visualize findings for questions 1 and 2. 4. Average purchase value by client type and total purchase value by product line 5. Summarize your findings.

0.2 The data

The sales data has the following fields:

- "date" - The date, from June to August 2021.
- "warehouse" - The company operates three warehouses: North, Central, and West.
- "client_type" - There are two types of customers: Retail and Wholesale.
- "product_line" - Type of products purchased.
- "quantity" - How many items were purchased.
- "unit_price" - Price per item sold.
- "total" - Total sale = quantity * unit_price.
- "payment" - How the client paid: Cash, Credit card, Transfer.

[6]:
```
import pandas as pd

df = pd.read_csv('data/sales_data.csv', parse_dates=['date'])
df = pd.DataFrame(df) #turn series into a DF
df.head() #Display part of dataset
```

[6]:
```
        date warehouse client_type           product_line  quantity  \
0 2021-06-01   Central      Retail          Miscellaneous         8
1 2021-06-01     North      Retail        Breaking system         9
2 2021-06-01     North      Retail  Suspension & traction         8
3 2021-06-01     North   Wholesale           Frame & body        16
4 2021-06-01   Central      Retail                 Engine         2

   unit_price   total      payment
0       16.85  134.83  Credit card
1       19.29  173.61         Cash
2       32.93  263.45  Credit card
```

```
3        37.84   605.44     Transfer
4        60.48   120.96   Credit card
```

0.3 Data analysis warehouse totals:

Find the total sales for each warehouse. We can use `groupby` to group the information by the column "warehouse". Then we select the column "total" and use `.sum()` to add the "total" column for each warehouse:

[2]:
```
df.groupby('warehouse')[['total']].sum()
```

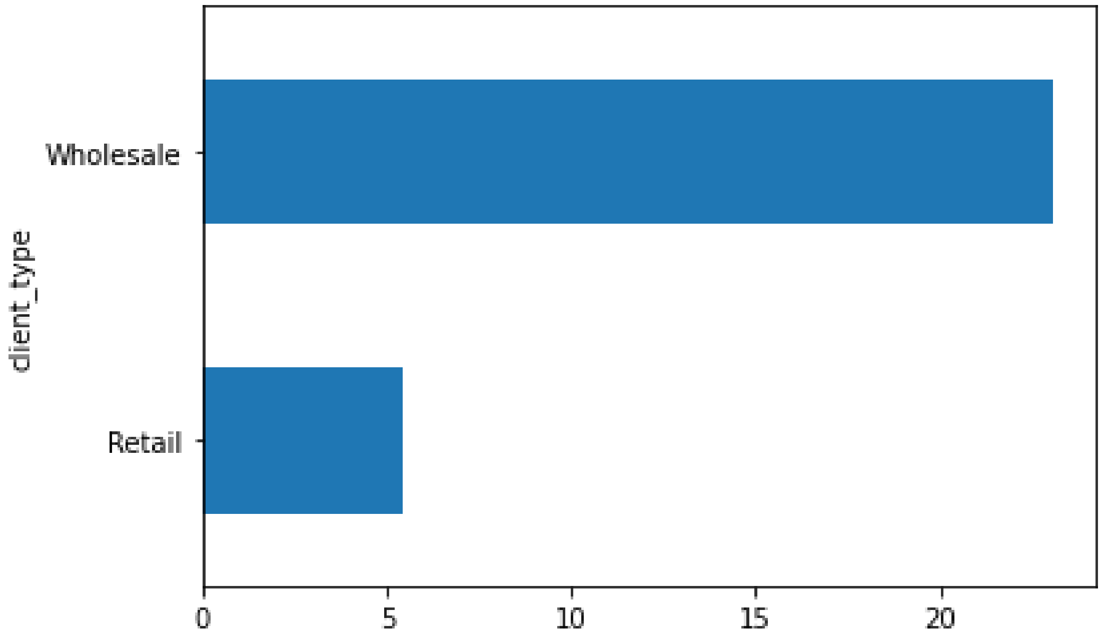
[2]:
```
               total
warehouse
Central    141982.88
North      100203.63
West        46926.49
```

0.4 Client Type Visual

A well-crafted chart often conveys information much better than a table. Here is the average number of items purchased by each client type. We are using the `matplotlib.pyplot` library for this example. We will run the `.plot()` method on the data we want to display and call `plt.show()` to draw the plot:

[3]:
```
import matplotlib.pyplot as plt

avg_units_client_type = df.groupby('client_type')['quantity'].mean()
avg_units_client_type.plot(kind='barh')
plt.show()
```



1 Total Sales Analysis

Group by Paytype

[17]:
```
payType=df.groupby('product_line')['total'].sum().reset_index().
↪sort_values(by='total', ascending=False)
print(payType)
```

```
             product_line     total
5  Suspension & traction  73014.21
3           Frame & body  69024.73
1      Electrical system  43612.71
0        Breaking system  38350.15
2                 Engine  37945.38
4          Miscellaneous  27165.82
```

Average unit price for each product line

[13]:
```
product_line=round(df.groupby('product_line')['total'].mean(),2)
print(product_line)
```

```
product_line
Breaking system          166.74
Electrical system        225.97
Engine                   622.06
Frame & body             415.81
Miscellaneous            222.67
Suspension & traction    320.24
Name: total, dtype: float64
```

Average purchase value by client type

[20]:
```
df_clients=round(df.groupby('client_type')['total'].mean(), 2)
print(df_clients)
```

```
client_type
Retail       167.06
Wholesale    709.52
Name: total, dtype: float64
```

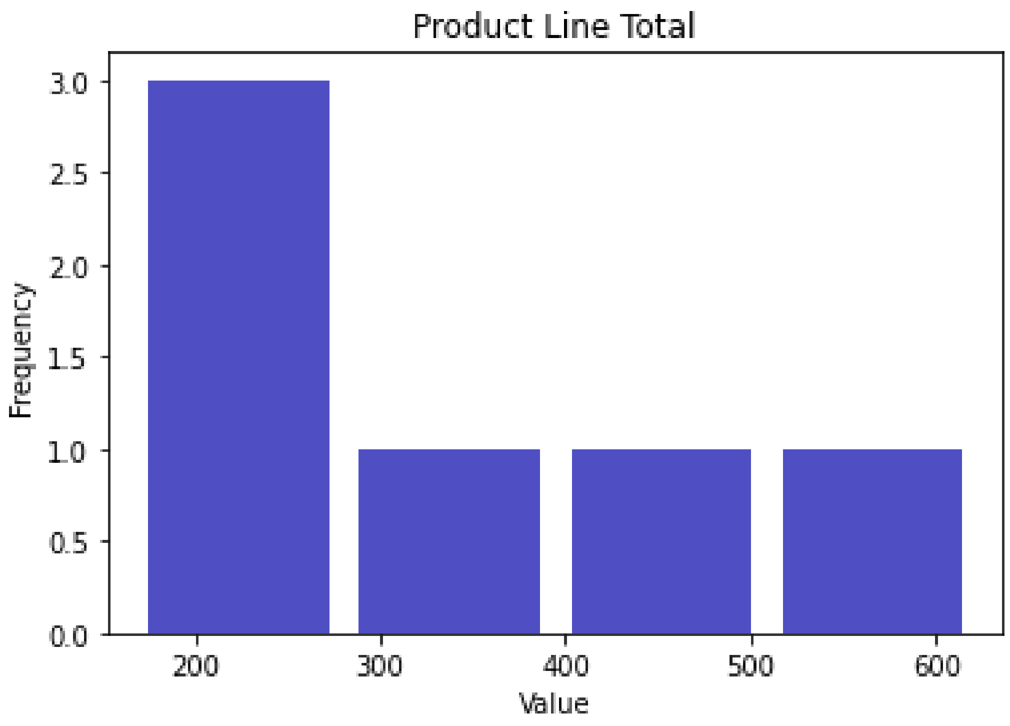
1.1 Visualizations

[19]:
```
import matplotlib.pyplot as plt
import numpy as np

#PLOT PRODUCT LINE TOTALS:
```

```
n, bins, patches = plt.hist(x=product_line_total, bins='auto', color='#0504aa',
                            alpha=0.7, rwidth=0.85)
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Product Line Total')
```

```
[19]: Text(0.5, 1.0, 'Product Line Total')
```



2 Summary

The above product_line is grouped and then sorted by “total” amount. This allows the user to see the data easily sorted from high to low.