Solution Review: Exploring E-Commerce

This lesson provides solutions to the exercise on exploring E-Commerce Dataset in previous lesson.

WE'LL COVER THE FOLLOWING

1. Top 5 customers with the highest number of orders

```
import pandas as pd
df = pd.read_csv('e_commerce.csv')

# solution
temp = df.groupby('CustomerID').size()
temp = temp.sort_values(ascending=False)
temp = temp.iloc[:5]

print(temp)
```

We do this task in three steps:

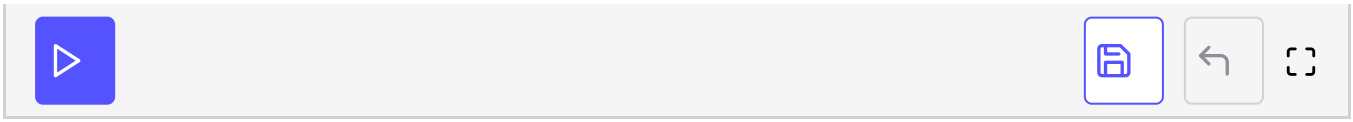
- First, we group our data with `CustomerID` and call `size` to retrieve the number of times each `CustomerID` appeared in the data in **line 5**.
- Second, we sort the values in descending order using `sort_values` in **line 6**.
- In the end, we just take the top 5 customers since they are already sorted.

2. Top 5 customers with the most amount of money spent

```
import pandas as pd
df = pd.read_csv('e_commerce.csv')

# solution
temp = df.groupby('CustomerID').sum()
temp = temp['AmountSpent']
temp = temp.sort_values(ascending=False)
temp = temp.iloc[:5]

print(temp)
```



We do this task in four steps:

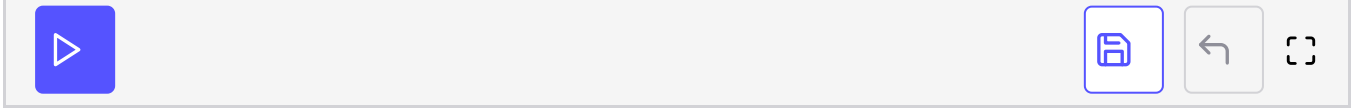
- First, we group our data with `CustomerID` and call `sum` since we want the `AmountSpent` of all orders added up for each customer in **line 5**.
- Then we select the `AmountSpent` column in **line 6**.
- Then we sort the values in descending order using `sort_values` in **line** 7.
- In the end, we just take the top 5 customers since they are already sorted.

3. Top 5 countries with the highest number of orders

```
import pandas as pd
df = pd.read_csv('e_commerce.csv')

# solution
temp = df.groupby('Country').size()
temp = temp.sort_values(ascending=False)
temp = temp.iloc[:5]

print(temp)
```



We do this task in three steps:

- First, we group our data with `Country` and call `size` to retrieve the number of times each `Country` appeared in the data in **line 5**.
- Second, we sort the values in descending order using `sort_values` in **line 6**.
- In the end, we just take the top 5 countries since these are already sorted.

4. Number of orders for every month in 2011

```
import pandas as pd
df = pd.read_csv('e_commerce.csv')

# solution
condition = df['PurchaseYear'] == 2011
temp = df[condition]
temp = temp.groupby('PurchaseMonth').size()

print(temp)
```

We do this task in two steps:

- First, we have to filter the data to keep entries for only 2011. For this, we specify our condition in **line 5**, then we filter using it in the next line.
- Second, we group our data by `PurchaseMonth` and call `size` to retrieve the number of times each month appeared in the data in **line** 7.

5. Top 10 most ordered products

```
import pandas as pd
df = pd.read_csv('e_commerce.csv')

# solution
temp = df.groupby('Description').sum()
temp = temp['Quantity']
temp = temp.sort_values(ascending=False)
temp = temp.iloc[:10]

print(temp)
```

We do this task in four steps:

- First, we group our data with `Description` and call `sum` since we want the `Quantity` of all orders added up for each product in **line 5**.
- Then we select the `Quantity` column in **line 6**.
- Then we sort the values in descending order using `sort_values` in **line** 7.
- In the end, we just take the top 10 products since they are already sorted.

In the next lesson, we will look at how we can perform RFM analysis in Python.