Exercise: Exploring E-Commerce

This lesson tests the learners on EDA on an e-commerce dataset.

WE'LL COVER THE FOLLOWING

E-Commerce data

In this lesson, you are going to be tested on exploring E-commerce data. The dataset was made available on the UCI Machine Learning Repository. This is a transnational data set that contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail. We will be using a **sample** of it.

In the below exercise, you will be writing functions for every task. The functions will receive a dataframe `df` as an input argument. Your task will be to perform the required operations to answer a question and return the answer to that.

```
# E-commerce Data Attributes
# InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each tra
# StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to eac
# Description: Product (item) name. Nominal.
# Quantity: The quantities of each product (item) per transaction. Numeric.
# InvoiceDate: Invoice Date and time. Numeric, the day and time when each transaction was ger
# UnitPrice: Unit price. Numeric, Product price per unit in sterling.
# CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each c
# Country: Country name. Nominal, the name of the country where each customer resides.
# AmountSpent: Total amount of order i.e. UnitPrice * Quantity
# PurchaseYear: year of purchase
# PurchaseMonth: Month of purchase (1-12)
# PurcahseHour: Hour of purchase (0-23)
```

1. Top 5 customers with the highest number of orders

Find the `CustomerID` of the top 5 customers with the highest number of orders.

Input

A dataframe

Output

The number of orders against `CustomerID` of top 5 customers.

2. Top 5 customers with most amount of money spent

Find the `CustomerID` of the top 5 customers with the most amount of money spent.

Input

A dataframe

Output

Amount of Money Spent against `CustomerID` of top 5 customers.

3. Top 5 countries with the highest number of orders

Find the top 5 Countries from where most orders come from.

Input

A dataframe

Output

Number of Orders against the names of the top 5 countries.

4. Number of orders for every month in 2011.

Find the number of orders for every month in the year 2011.

Input

A dataframe

Output

Numbers of orders against each month.

5. Top 10 most ordered products

Find the top 10 most ordered products.

Input

A dataframe

Output

Numbers of orders against each product.

Some useful tips

- You can use `size` function after `groupby` to retrieve the number of times a category appeared.

- To sort a series, `sort_values(ascending=True)` can be used. For descending order, give `ascending=False`.
- To sort a dataframe, `sort_values(by=column_name, ascending=True)` can be used. Replace `column_name` with the name of the column by which you want the rows to be sorted. For descending order, give `ascending=False`.
- You can use `df.iloc[:5]` to retrieve the first 5 rows from the dataframe or series.

```
# Top 5 Customers with Most Number of Orders
def most_orders(df):
  pass

# Top 5 Customers with Most Amount of Money Spent
def most_money_spent(df):
  pass

# Top 5 Countries with the Most Number of Orders
def most_orders_countries(df):
  pass

# Number of Orders for every Month in 2011.
def num_orders_months(df):
  pass

# Top 10 Most Ordered Products.
def most_ordered_products(df):
  pass
```