(4) (Exercice) Sales Analysis

July 23, 2022

[2]:
```python
print('Sales Analysis')
```

```
Sales Analysis
```

[3]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
```

[4]:
```python
sales_df = pd.read_csv('train.csv')
```

[5]:
```python
sales_df.head()
```

[5]:
```
   Row ID        Order ID  Order Date   Ship Date       Ship Mode Customer ID  \
0       1  CA-2017-152156  08/11/2017  11/11/2017    Second Class    CG-12520
1       2  CA-2017-152156  08/11/2017  11/11/2017    Second Class    CG-12520
2       3  CA-2017-138688  12/06/2017  16/06/2017    Second Class    DV-13045
3       4  US-2016-108966  11/10/2016  18/10/2016  Standard Class    SO-20335
4       5  US-2016-108966  11/10/2016  18/10/2016  Standard Class    SO-20335

     Customer Name    Segment        Country             City       State  \
0      Claire Gute   Consumer  United States        Henderson    Kentucky
1      Claire Gute   Consumer  United States        Henderson    Kentucky
2  Darrin Van Huff  Corporate  United States      Los Angeles  California
3   Sean O'Donnell   Consumer  United States  Fort Lauderdale     Florida
4   Sean O'Donnell   Consumer  United States  Fort Lauderdale     Florida

   Postal Code Region       Product ID         Category Sub-Category  \
0      42420.0  South  FUR-BO-10001798        Furniture    Bookcases
1      42420.0  South  FUR-CH-10000454        Furniture       Chairs
2      90036.0   West  OFF-LA-10000240  Office Supplies       Labels
3      33311.0  South  FUR-TA-10000577        Furniture       Tables
4      33311.0  South  OFF-ST-10000760  Office Supplies      Storage

                                        Product Name     Sales
0                  Bush Somerset Collection Bookcase  261.9600
```

```
1  Hon Deluxe Fabric Upholstered Stacking Chairs,…  731.9400
2  Self-Adhesive Address Labels for Typewriters b…   14.6200
3      Bretford CR4500 Series Slim Rectangular Table  957.5775
4                     Eldon Fold 'N Roll Cart System   22.3680
```

[6]:
```
sales_df.count()
```

[6]:
```
Row ID           9800
Order ID         9800
Order Date       9800
Ship Date        9800
Ship Mode        9800
Customer ID      9800
Customer Name    9800
Segment          9800
Country          9800
City             9800
State            9800
Postal Code      9789
Region           9800
Product ID       9800
Category         9800
Sub-Category     9800
Product Name     9800
Sales            9800
dtype: int64
```

[7]:
```
print('Missing Data : ')
```

```
Missing Data :
```

[8]:
```
print('Equals to null data : ')
sales_df.isnull().sum()
```

```
Equals to null data :
```

[8]:
```
Row ID            0
Order ID          0
Order Date        0
Ship Date         0
Ship Mode         0
Customer ID       0
Customer Name     0
Segment           0
Country           0
City              0
State             0
```

```
Postal Code      11
Region            0
Product ID        0
Category          0
Sub-Category      0
Product Name      0
Sales             0
dtype: int64
```

[9]:
```
print('Equals to na data : ')
sales_df.isna().sum()
```

```
Equals to na data :
```

[9]:
```
Row ID            0
Order ID          0
Order Date        0
Ship Date         0
Ship Mode         0
Customer ID       0
Customer Name     0
Segment           0
Country           0
City              0
State             0
Postal Code      11
Region            0
Product ID        0
Category          0
Sub-Category      0
Product Name      0
Sales             0
dtype: int64
```

[10]:
```
print('There are 11 missing data in the postal code column.')
print('Let\'s get rid of the rows where the postal code is missing, since we␣
 ↪have 9800 values, 11 do not seem too much to throw away.')
```

```
There are 11 missing data in the postal code column.
Let's get rid of the rows where the postal code is missing, since we have 9800
values, 11 do not seem too much to throw away.
```

[11]:
```
cleaned_sales_df = sales_df.dropna(how='any')
```

[12]:
```
cleaned_sales_df.count()
```

```
[12]: Row ID           9789
      Order ID         9789
      Order Date       9789
      Ship Date        9789
      Ship Mode        9789
      Customer ID      9789
      Customer Name    9789
      Segment          9789
      Country          9789
      City             9789
      State            9789
      Postal Code      9789
      Region           9789
      Product ID       9789
      Category         9789
      Sub-Category     9789
      Product Name     9789
      Sales            9789
      dtype: int64
```

```
[13]: order_date = cleaned_sales_df['Order Date']
      order_date.replace('/','-')
      order_date = pd.to_datetime(order_date, dayfirst=True)
```

```
[14]: print('Range of the orders through time : ')
```

```
Range of the orders through time :
```

```
[15]: print('Start date of the orders : January the 2nd 2015')
      np.min(order_date)
```

```
Start date of the orders : January the 2nd 2015
```

```
[15]: Timestamp('2015-01-03 00:00:00')
```

```
[16]: print('End date of the orders : December the 30rd 2018')
      np.max(order_date)
```

```
End date of the orders : December the 30rd 2018
```

```
[16]: Timestamp('2018-12-30 00:00:00')
```

```
[17]: print('Time range of the orders : 1458 days, almost 4 years (4 years minus 3␣
       ↪days)')
      (np.max(order_date) - np.min(order_date))
```

```
Time range of the orders : 1458 days, almost 4 years (4 years minus 3 days)
```

```
[17]: Timedelta('1457 days 00:00:00')
```

```
[18]: print('Average time between orders and shipping date :')
```

```
Average time between orders and shipping date :
```

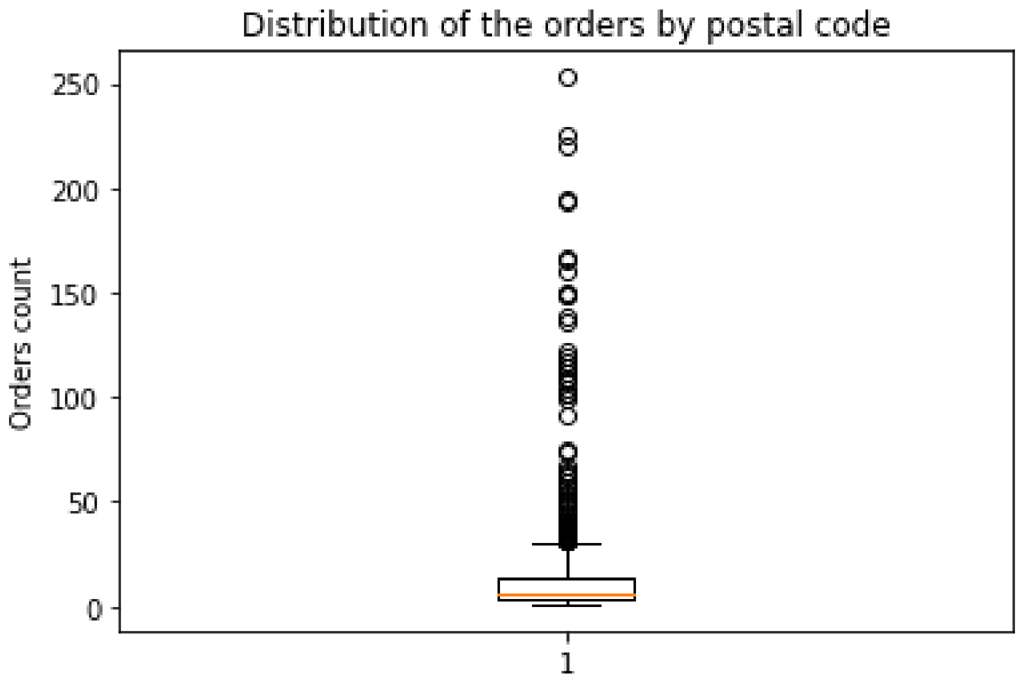
```
[19]: ship_date = cleaned_sales_df['Ship Date']
      ship_date.replace('/','-')
      ship_date = pd.to_datetime(ship_date, dayfirst=True)
      mean_orders_ship = np.mean(ship_date - order_date)
      print('The average time between the orders and the shipping date is 3 days and␣
       ↪23 hours.')
      mean_orders_ship
```

```
The average time between the orders and the shipping date is 3 days and 23
hours.
```

```
[19]: Timedelta('3 days 23:04:06.031259577')
```

```
[20]: most_orders_states = cleaned_sales_df['Postal Code'].value_counts()
```

```
[21]: plt.boxplot(most_orders_states)
      plt.title('Distribution of the orders by postal code')
      plt.ylabel('Orders count')
      plt.show()
      print('We can see that most of the orders are made in a few cities, while there␣
       ↪are lots of cities where few orders were placed.')
```



We can see that most of the orders are made in a few cities, while there are lots of cities where few orders were placed.

[62]:
```python
print('What is the most used shipping method ? (Standard Class shipping)')
```

```
What is the most used shipping method ? (Standard Class shipping)
```

[23]:
```python
shipping_method_count = cleaned_sales_df['Ship Mode'].value_counts()
shipping_method_count
```

[23]:
```
Standard Class    5849
Second Class      1901
First Class       1501
Same Day           538
Name: Ship Mode, dtype: int64
```

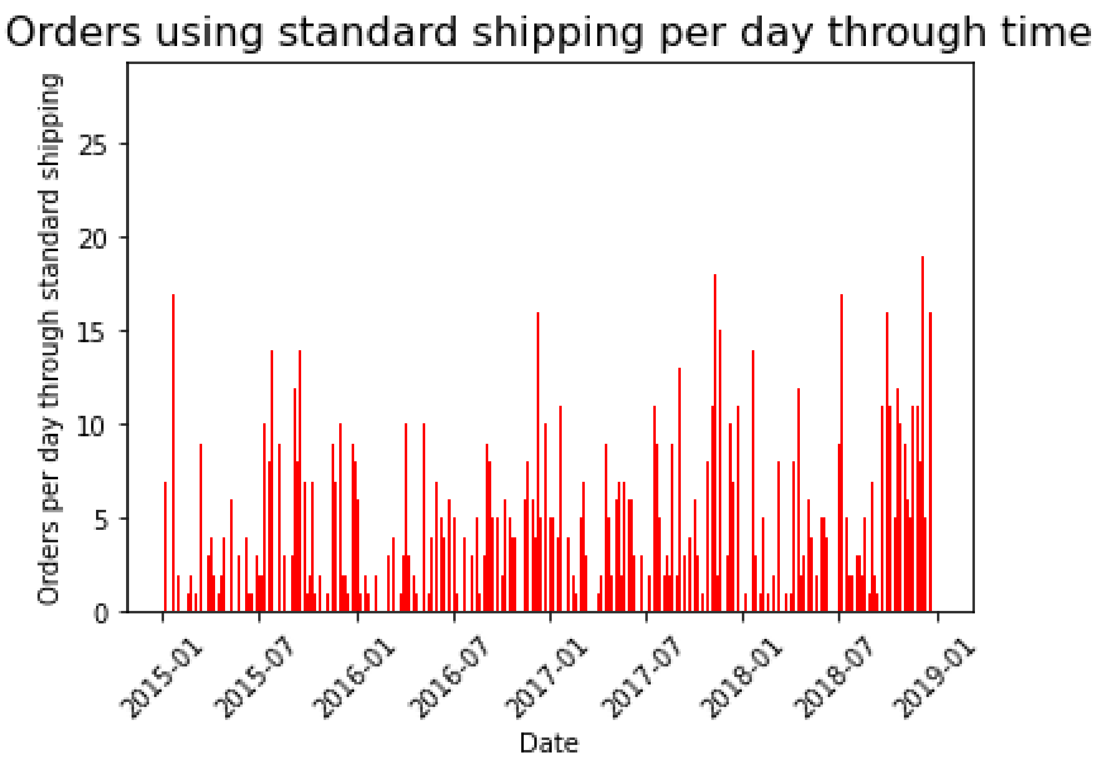
[24]:
```python
print('What is the use of standard shipping through time ?')
```

```
What is the use of standard shipping through time ?
```

[25]:
```python
standard_class_data = cleaned_sales_df.loc[cleaned_sales_df['Ship␣
 ↪Mode']=='Standard Class']
order_date_standard_class_data = standard_class_data['Order Date']
order_date_standard_class_data.replace('/','-')
order_date_standard_class_data = pd.to_datetime(order_date_standard_class_data,␣
 ↪dayfirst=True)
```

[26]:
```python
order_count_standard_class_data = order_date_standard_class_data.sort_values().
 ↪reindex().value_counts().sort_index().reset_index(level=0).
 ↪rename(columns={'index':'TimeStamp','Order Date':'Sales'})
```

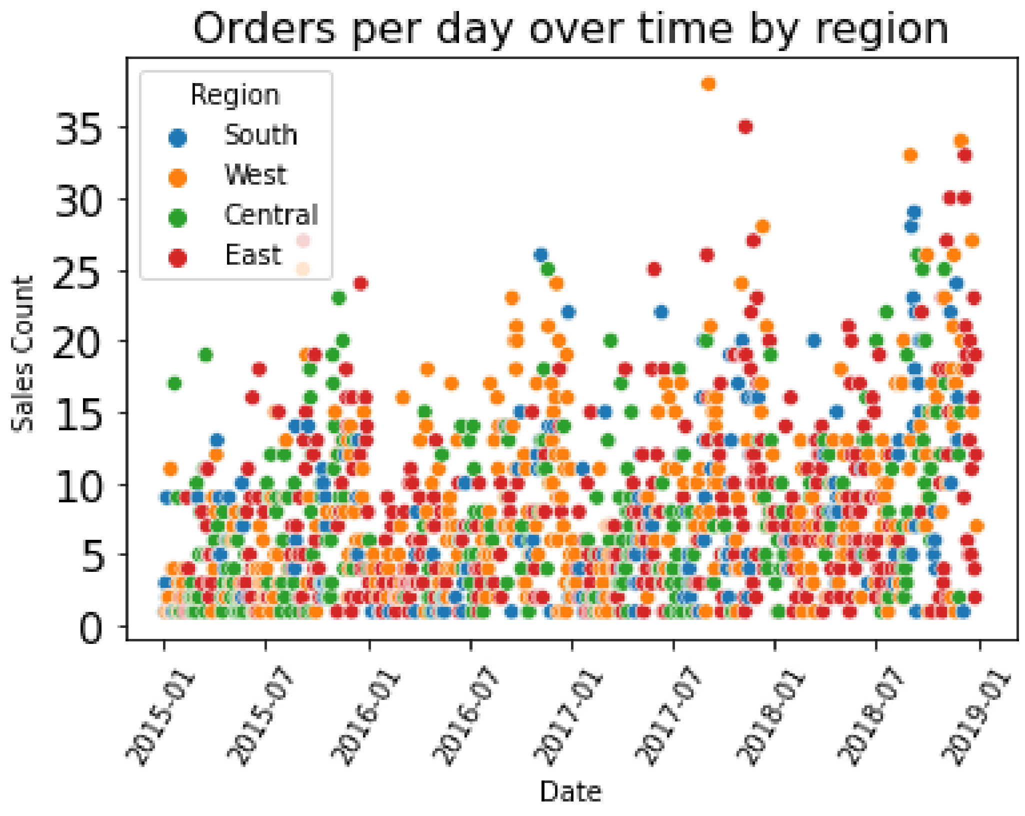
[59]:
```python
x = order_count_standard_class_data['TimeStamp']
y = order_count_standard_class_data['Sales']
plt.bar(x,y,color='red')
plt.title('Orders using standard shipping per day through time',fontsize=16)
plt.xlabel('Date')
plt.xticks(rotation=45)
plt.ylabel('Orders per day through standard shipping')
plt.show()
```



[42]:
```
sales_count_date_df = order_date.value_counts().sort_index().reset_index().
 ↪rename(columns={'index':'Date','Order Date':'Sales Count'})
```

[60]:
```
x_1 = sales_count_date_df['Date']
y_1 = sales_count_date_df['Sales Count']

sns.scatterplot(x=x_1,y=y_1,data=cleaned_sales_df,hue='Region')
plt.title('Orders per day over time by region',fontsize=16)
plt.xticks(rotation=60)
plt.yticks(fontsize=16)
plt.show()
```



[61]:
```
print('There are some clear peaks of orders at the end of the years, probably␣
 ↪due to holidays.')
```

```
There are some clear peaks of orders at the end of the years, probably due to
holidays.
```

[63]:
```
print('Made by : Nicolas Mrynck')
```

```
Made by : Nicolas Mrynck
```

[ ]:
```
```