(2) (Exercices) Supermarket Sales Analysis

July 23, 2022

```
[2]: print('Analysis of supermarket sales')
```

```
Analysis of supermarket sales
```

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

```
[4]: sales_df = pd.read_csv('Sales Dataset.csv')
```

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

```
[5]:   Order ID Customer Name          Category      Sub Category         City  \
     0      OD1        Harish      Oil & Masala           Masalas      Vellore
     1      OD2         Sudha         Beverages     Health Drinks  Krishnagiri
     2      OD3       Hussain       Food Grains      Atta & Flour   Perambalur
     3      OD4       Jackson  Fruits & Veggies  Fresh Vegetables   Dharmapuri
     4      OD5       Ridhesh       Food Grains   Organic Staples         Ooty

        Order Date Region  Sales  Discount  Profit       State
     0  11-08-2017  North   1254      0.12  401.28  Tamil Nadu
     1  11-08-2017  South    749      0.18  149.80  Tamil Nadu
     2  06-12-2017   West   2360      0.21  165.20  Tamil Nadu
     3  10-11-2016  South    896      0.25   89.60  Tamil Nadu
     4  10-11-2016  South   2355      0.26  918.45  Tamil Nadu
```

```
[6]: print('Count of values :')
     sales_df.count()
```

```
Count of values :
```

```
[6]: Order ID         9994
     Customer Name    9994
     Category         9994
     Sub Category     9994
     City             9994
```

```
Order Date       9994
Region           9994
Sales            9994
Discount         9994
Profit           9994
State            9994
dtype: int64
```

```
[7]: print('Count of missing values : ')
     sales_df.isna().sum()
```

```
Count of missing values :
```

```
[7]: Order ID         0
     Customer Name    0
     Category         0
     Sub Category     0
     City             0
     Order Date       0
     Region           0
     Sales            0
     Discount         0
     Profit           0
     State            0
     dtype: int64
```

```
[8]: print('Count of null values : ')
     sales_df.isnull().sum()
```

```
Count of null values :
```

```
[8]: Order ID         0
     Customer Name    0
     Category         0
     Sub Category     0
     City             0
     Order Date       0
     Region           0
     Sales            0
     Discount         0
     Profit           0
     State            0
     dtype: int64
```

```
[9]: print('There are no missing or null values.')
```

```
There are no missing or null values.
```

[10]:
```
sales_df['Order Date'] = pd.to_datetime(sales_df['Order Date'])
```

[11]:
```
print('Start date of the data set : January th 3rd 2015')
np.min(sales_df['Order Date'])
```

```
Start date of the data set : January th 3rd 2015
```

[11]:
```
Timestamp('2015-01-03 00:00:00')
```

[12]:
```
print('End date of the data set : December the 30th 2018')
np.max(sales_df['Order Date'])
```

```
End date of the data set : December the 30th 2018
```

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

[13]:
```
print('Range time of the data set : 4 days less to 4 years')
print((np.max(sales_df['Order Date']) - np.min(sales_df['Order Date'])))
```

```
Range time of the data set : 4 days less to 4 years
1457 days 00:00:00
```

[14]:
```
cities_sales_values = sales_df.City.value_counts()
print('Sales values by city : ')
cities_sales_values
```

```
Sales values by city :
```

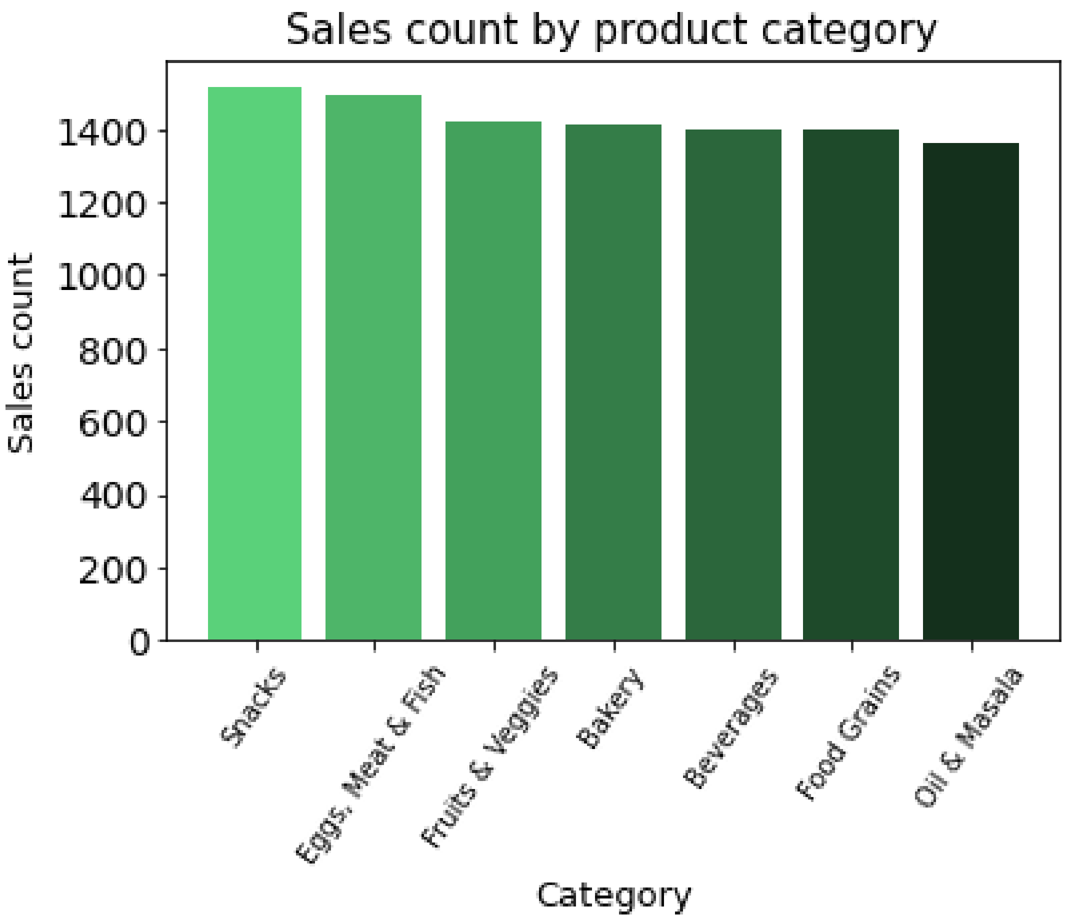
[14]:
```
Kanyakumari       459
Tirunelveli       446
Bodi              442
Krishnagiri       440
Vellore           435
Perambalur        434
Tenkasi           432
Chennai           432
Salem             431
Karur             430
Pudukottai        430
Coimbatore        428
Ramanadhapuram    421
Cumbum            417
Virudhunagar      416
Madurai           408
Ooty              404
Namakkal          403
Viluppuram        397
Dindigul          396
```

```
Theni                387
Dharmapuri           376
Nagercoil            373
Trichy               357
Name: City, dtype: int64
```

[15]:

```
sales_count_by_product_category = sales_df['Category'].value_counts()
print('Sales count by product category :')
sales_count_by_product_category_df = sales_count_by_product_category.
 ↪reset_index().rename(columns={'index':'Category','Category':'Sales count'})
x = sales_count_by_product_category_df['Category']
y = sales_count_by_product_category_df['Sales count']
plt.
 ↪bar(x,y,color=['#5ad17a','#4eb569','#43a15c','#347d48','#2b663b','#1e4a2a','#14301c'])
plt.title('Sales count by product category',fontsize=15)
plt.xlabel('Category',fontsize=13)
plt.xticks(rotation=55)
plt.ylabel('Sales count',fontsize=13)
plt.yticks(fontsize=14)
plt.show()
```

```
Sales count by product category :
```



[16]:
```
print('Sales values by product category for Kanyakumari')
sales_values_by_product_category_for_Kanyakumari = sales_df.
 ↪loc[sales_df['City']=='Kanyakumari']
sales_values_by_product_category_for_Kanyakumari['Category'].value_counts()
```

```
Sales values by product category for Kanyakumari
```

[16]:
```
Oil & Masala        79
Snacks              75
Eggs, Meat & Fish   73
Bakery              64
Fruits & Veggies    59
Food Grains         58
Beverages           51
Name: Category, dtype: int64
```

[17]:
```
print('While Oil & Masala are the least sold products overall they are the␣
 ↪highest sales in the highest sales store, \nmarketing should focus locally.')
```

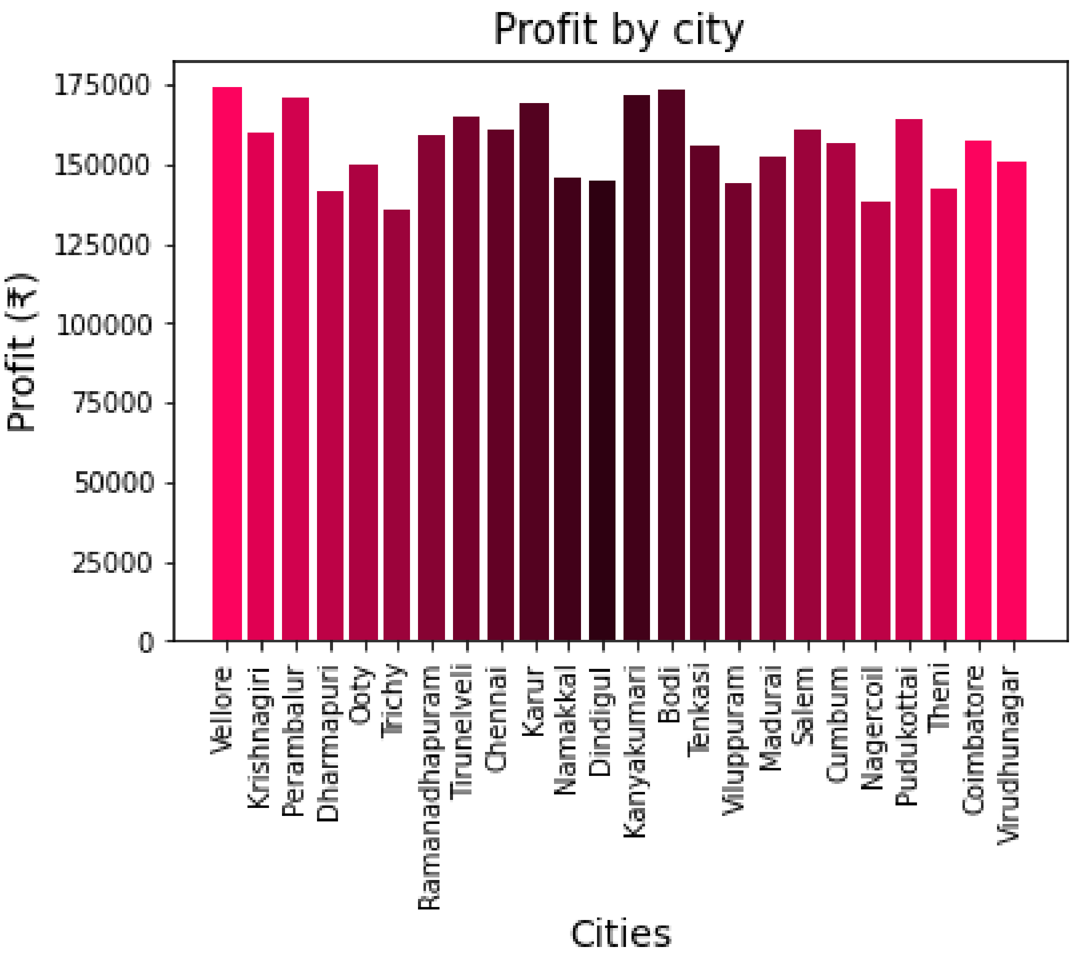
While Oil & Masala are the least sold products overall they are the highest sales in the highest sales store,
marketing should focus locally.

[18]:

```
cities = sales_df['City']
profit_by_city_dict = {}
for city in cities:
    profit_by_city_dict.update({str(city):str(int(sum(sales_df.
 ↪loc[sales_df['City']==city].Profit)))})
    set(profit_by_city_dict)
```

[19]:

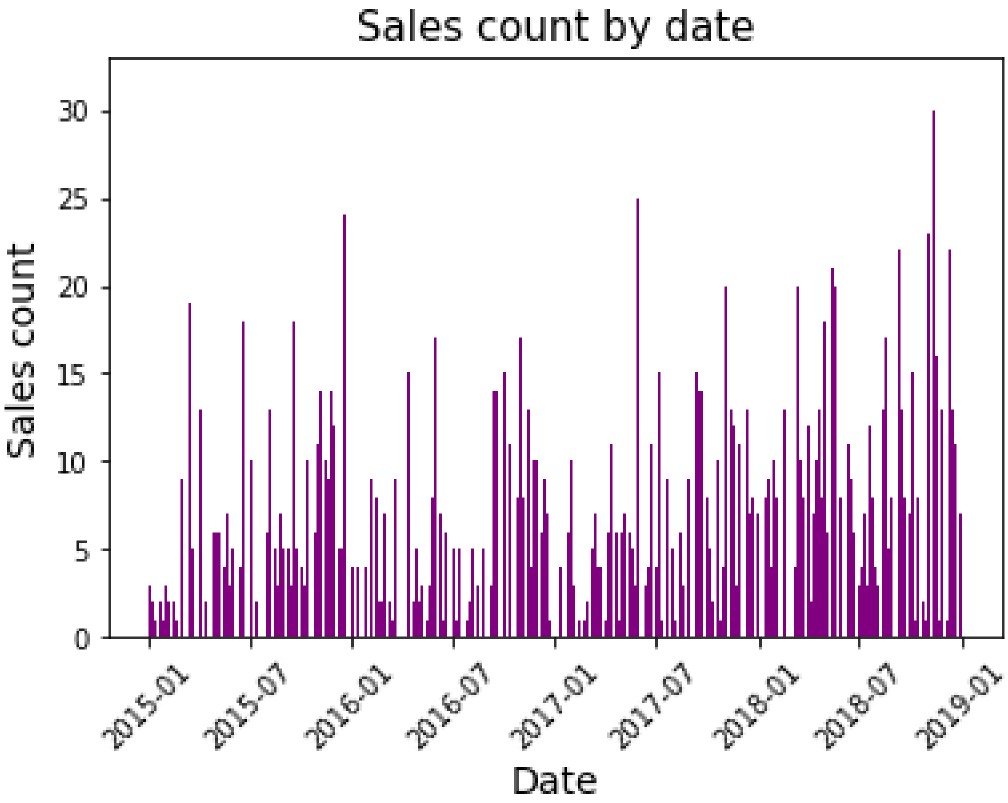
```
profit_by_city_df = pd.DataFrame(list(profit_by_city_dict.items()))
profit_by_city_df = profit_by_city_df.rename(columns={0:'City',1:'Profit'})
x_1 = profit_by_city_df['City']
y_1 = [int(x) for x in profit_by_city_df['Profit']]
plt.
 ↪bar(x_1,y_1,color=['#fc035e','#e00052','#d1024e','#bd0246','#ad0241','#9c033b','#870333','#
plt.xlabel('Cities',fontsize=14)
plt.xticks(rotation= 90,fontsize=10)
plt.ylabel('Profit ( )',fontsize=14)
plt.title('Profit by city',fontsize=15)
plt.show()
```



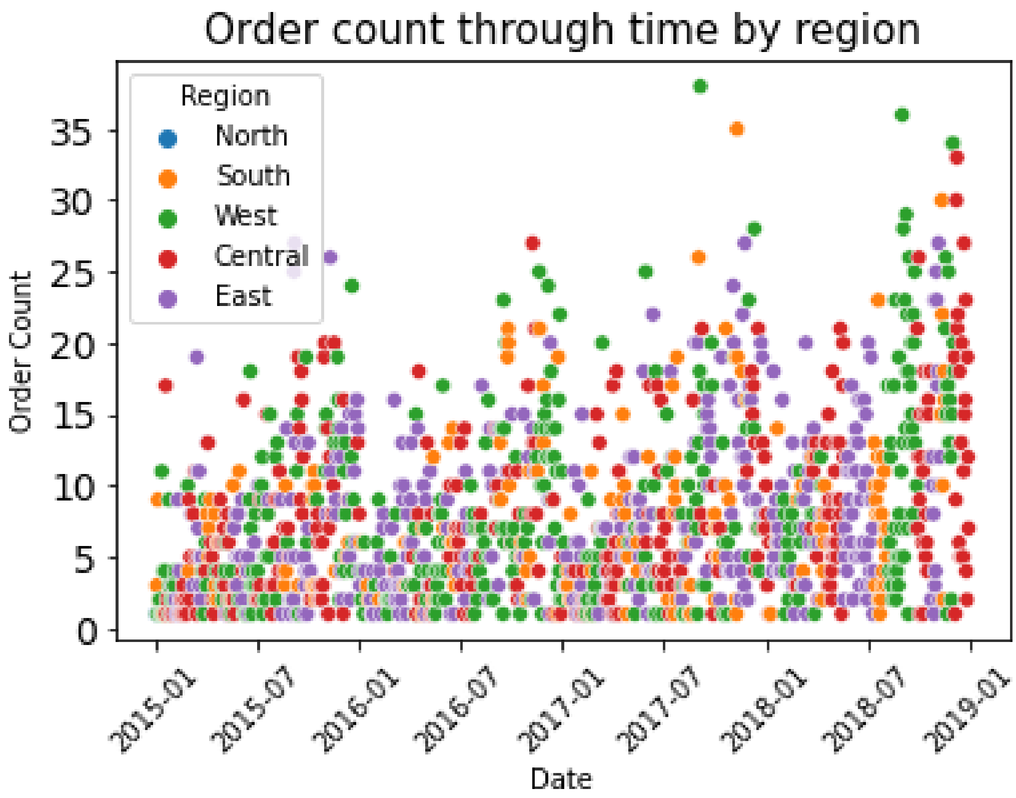
[20]:
```
sales_count_by_date = sales_df['Order Date'].sort_values().reindex().
 ↪value_counts().sort_index().reset_index(level=0).rename(columns={'index':
 ↪'TimeStamp','Order Date':'Sales Count'})
```

[21]:
```
x_2 = sales_count_by_date['TimeStamp']
y_2 = sales_count_by_date['Sales Count']
```

[22]:
```
plt.bar(x_2,y_2,color='purple')
plt.xticks(rotation= 45)
plt.xlabel('Date',fontsize=14)
plt.ylabel('Sales count',fontsize=14)
plt.ylim(0,33)
plt.title('Sales count by date',fontsize=15)
plt.show()
```



```
[59]: order_values_by_date = (sales_df['Order Date'].sort_values()).value_counts().
       ↪sort_index().reset_index().rename(columns={'index':'Date','Order Date':
       ↪'Order Count'})
      sns.scatterplot(x=order_values_by_date['Date'],y=order_values_by_date['Order␣
       ↪Count'],data=order_values_by_date,hue=sales_df['Region'])
      plt.yticks(fontsize=14)
      plt.xticks(rotation=45)
      plt.title('Order count through time by region',fontsize=15)
      plt.show()
```

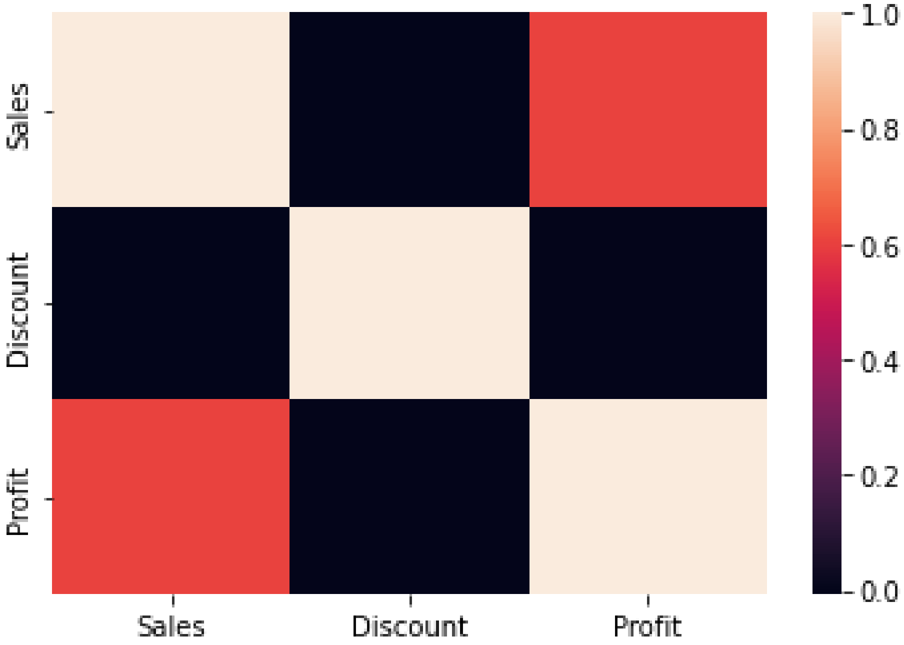


[64]:
```
correlation_matrix = sales_df.corr()
```

[65]:
```
correlation_matrix
```

[65]:
```
             Sales  Discount    Profit
Sales     1.000000 -0.005512  0.605349
Discount -0.005512  1.000000  0.000017
Profit    0.605349  0.000017  1.000000
```

[67]:
```
correlation_heatmap = sns.heatmap(correlation_matrix)
```



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

```
Made by : Nicolas Mrynck
```