(3) (Exercices) Supermarket Sales Analysis

July 23, 2022

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

```
Groceries Sales Analysis
```

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

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

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

```
[5]:    Member_number        Date   itemDescription
     0           1808  21-07-2015    tropical fruit
     1           2552  05-01-2015        whole milk
     2           2300  19-09-2015         pip fruit
     3           1187  12-12-2015  other vegetables
     4           3037  01-02-2015        whole milk
```

```
[6]: print('Date Range Of The Data : ')
```

```
Date Range Of The Data :
```

```
[7]: print('Start date of the data : January the 1st 2014')
     sales_df.Date = pd.to_datetime(sales_df.Date)
     np.min(sales_df.Date)
```

```
Start date of the data : January the 1st 2014
```

```
[7]: Timestamp('2014-01-01 00:00:00')
```

```
[8]: print('End date of the data : December the 30rd 2015')
     np.max(sales_df.Date)
```

```
End date of the data : December the 30rd 2015
```

```
[8]: Timestamp('2015-12-30 00:00:00')
```

```
[9]: print('Time range of the data : 2 years')
     np.max(sales_df.Date) - np.min(sales_df.Date)
```

```
     Time range of the data : 2 years
```

```
[9]: Timedelta('728 days 00:00:00')
```

```
[10]: print('Missing values : ')
```

```
      Missing values :
```

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

```
[11]: Member_number      38765
      Date               38765
      itemDescription    38765
      dtype: int64
```

```
[12]: sales_df.isnull().sum()
```

```
[12]: Member_number      0
      Date               0
      itemDescription    0
      dtype: int64
```

```
[13]: sales_df.isna().sum()
```

```
[13]: Member_number      0
      Date               0
      itemDescription    0
      dtype: int64
```

```
[14]: print('There are no record of missing data.')
```

```
      There are no record of missing data.
```

```
[15]: print('Products analysis : ')
```

```
      Products analysis :
```

```
[31]: product_sales = sales_df.itemDescription.value_counts()
      product_sales
```

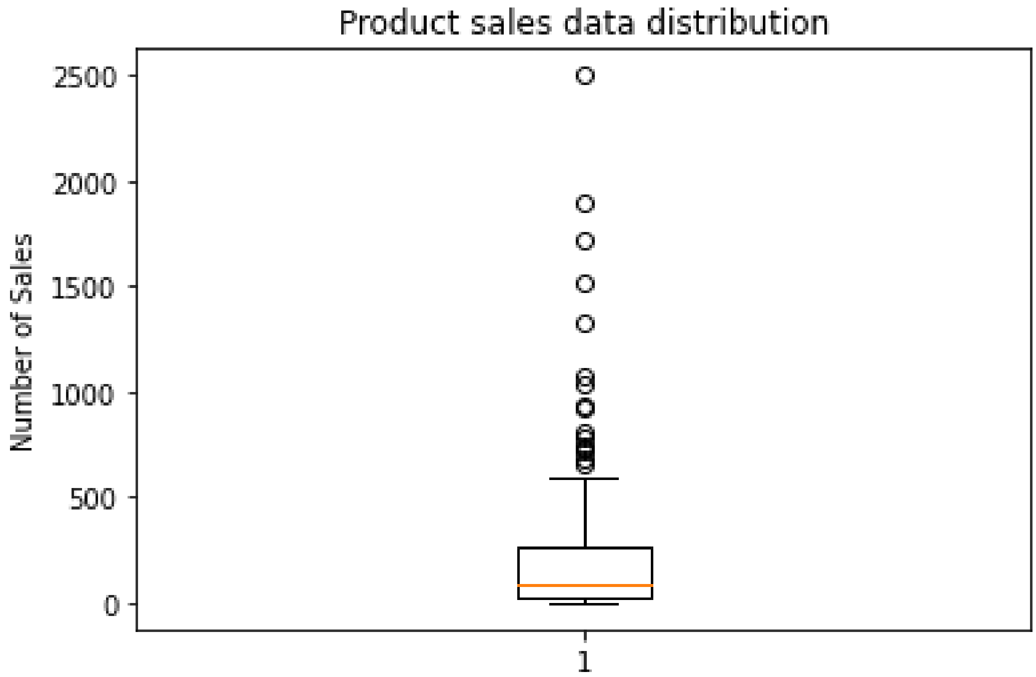
```
[31]: whole milk                 2502
      other vegetables           1898
      rolls/buns                 1716
```

```
soda                        1514
yogurt                      1334
                            …
rubbing alcohol                5
bags                           4
baby cosmetics                 3
kitchen utensil                1
preservation products          1
Name: itemDescription, Length: 167, dtype: int64
```

[17]:
```python
print('Visualisation of the product sales data distribution : ')
```

```
Visualisation of the product sales data distribution :
```

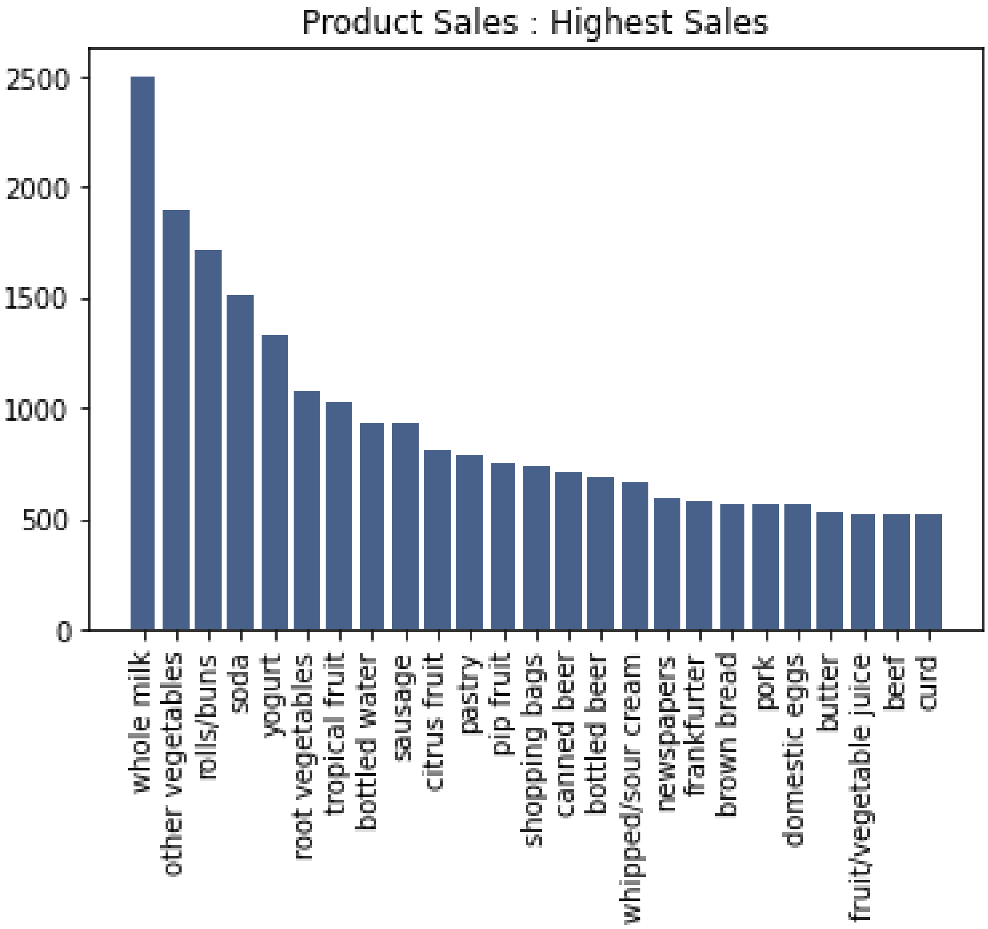
[18]:
```python
plt.boxplot(sales_df.itemDescription.value_counts())
plt.title('Product sales data distribution')
plt.ylabel('Number of Sales')
plt.show()
```



[30]:
```python
print('Most products have been sold below 500 times in 2 years, while most sold␣
 ↪products are outliers, maybe the store\'s inventory is too large.')
```

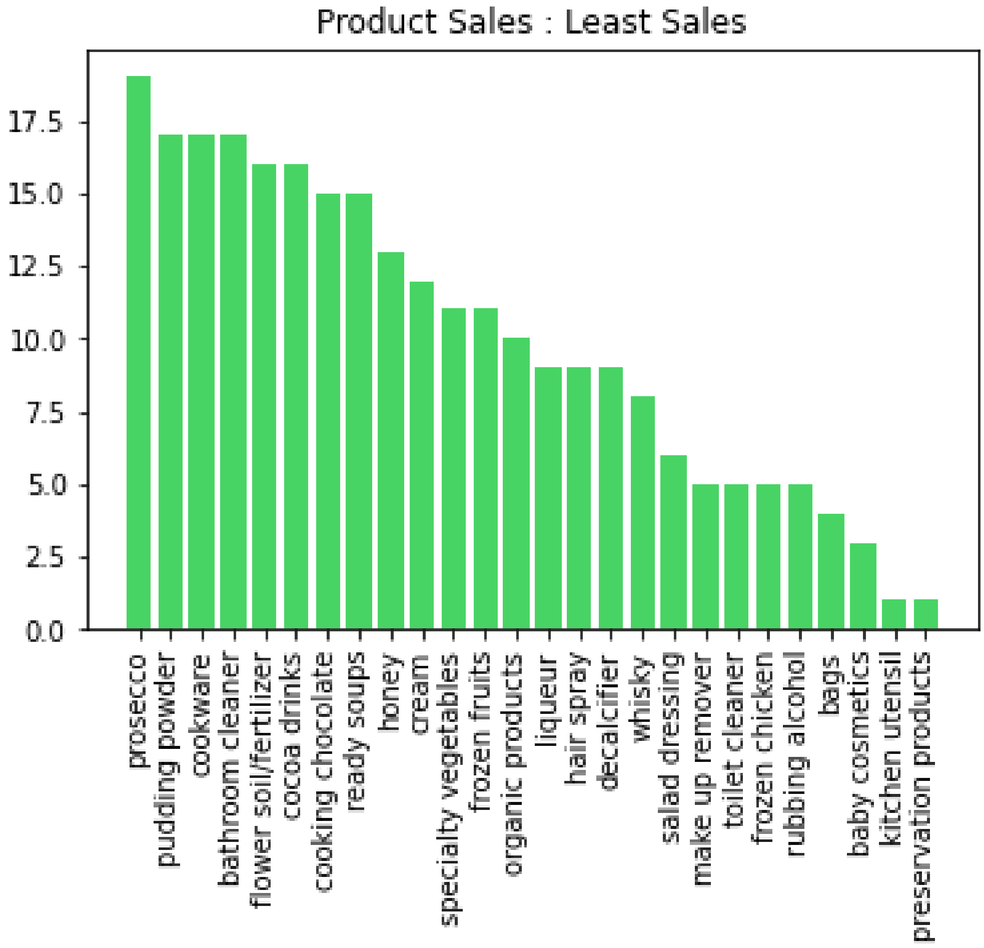
```
Most products have been sold below 500 times in 2 years, while most sold
products are outliers, maybe the store's inventory is too large.
```

```
[51]: product_sales_df = product_sales.reset_index().rename(columns={'index':
       ↪'Products','itemDescription':'Sales count'})
      product_sales_df_high_sales = product_sales_df.loc[product_sales_df['Sales␣
       ↪count']>500]
      x_3 = product_sales_df_high_sales['Products']
      y_3 = product_sales_df_high_sales['Sales count']

      plt.bar(x_3,y_3,color='#48618a')
      plt.title('Product Sales : Highest Sales')
      plt.xticks(rotation=90)
      plt.show()
```



```
[79]: product_sales_df_least_sales = product_sales_df.loc[product_sales_df['Sales␣
       ↪count']<20]
      x_3 = product_sales_df_least_sales['Products']
      y_3 = product_sales_df_least_sales['Sales count']
```

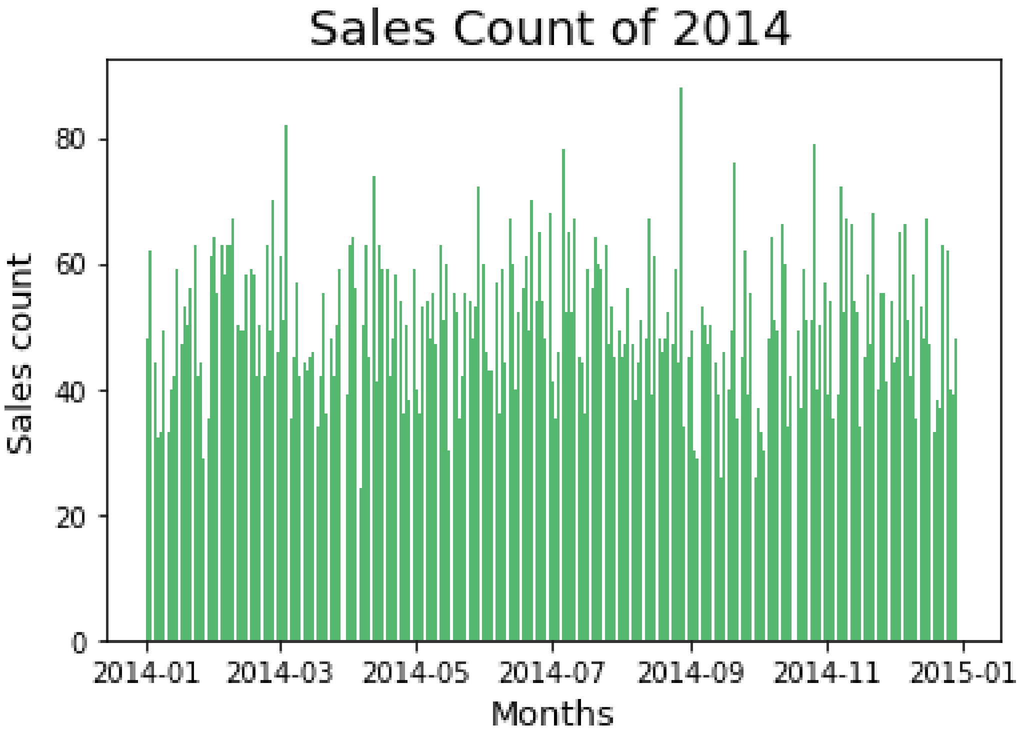
```
plt.bar(x_3,y_3,color='#48d464')
plt.title('Product Sales : Least Sales')
plt.xticks(rotation=90)
plt.show()
```



[55]:
```
sales_count_by_date = sales_df['Date'].sort_values().reindex().value_counts().
 ↪sort_index().reset_index(level=0).rename(columns={'index':'Date','Date':
 ↪'Sales Count'})
```

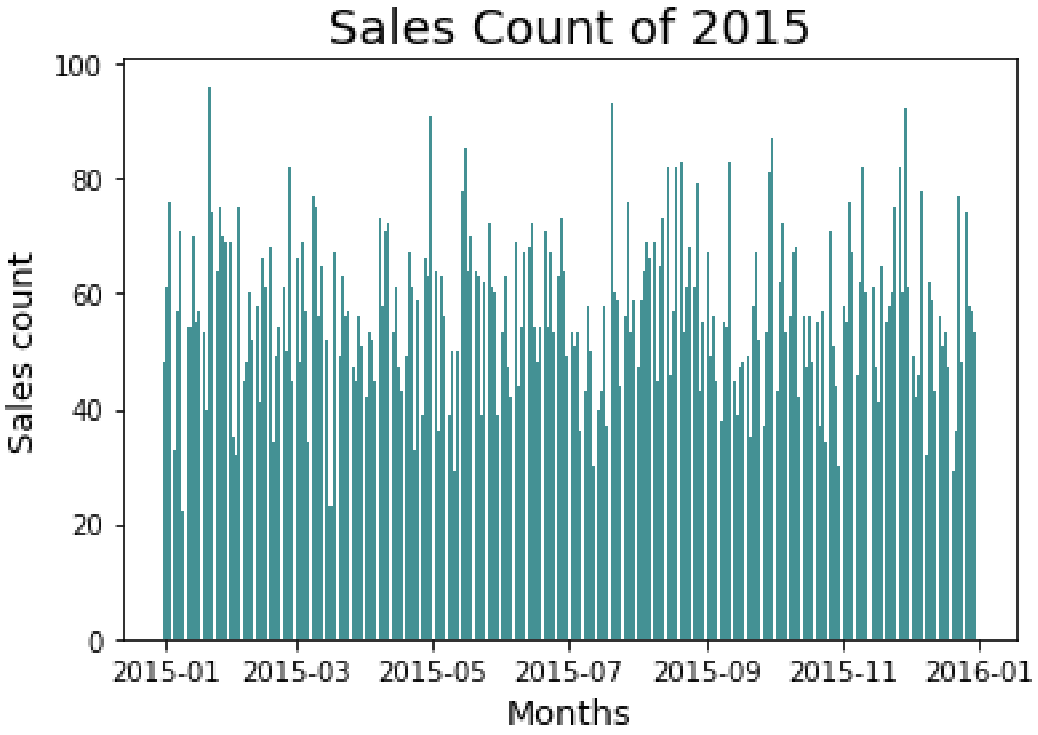
[57]:
```
sales_2014 = sales_count_by_date.loc[sales_count_by_date['Date']<pd.
 ↪to_datetime('2015-01-01')]
```

[58]:
```
x = sales_2014['Date']
y = sales_2014['Sales Count']

plt.bar(x,y,color='#54b86f')
plt.title('Sales Count of 2014',fontsize=18)
plt.ylabel('Sales count', fontsize=13)
plt.xlabel('Months', fontsize=13)
plt.show()
```



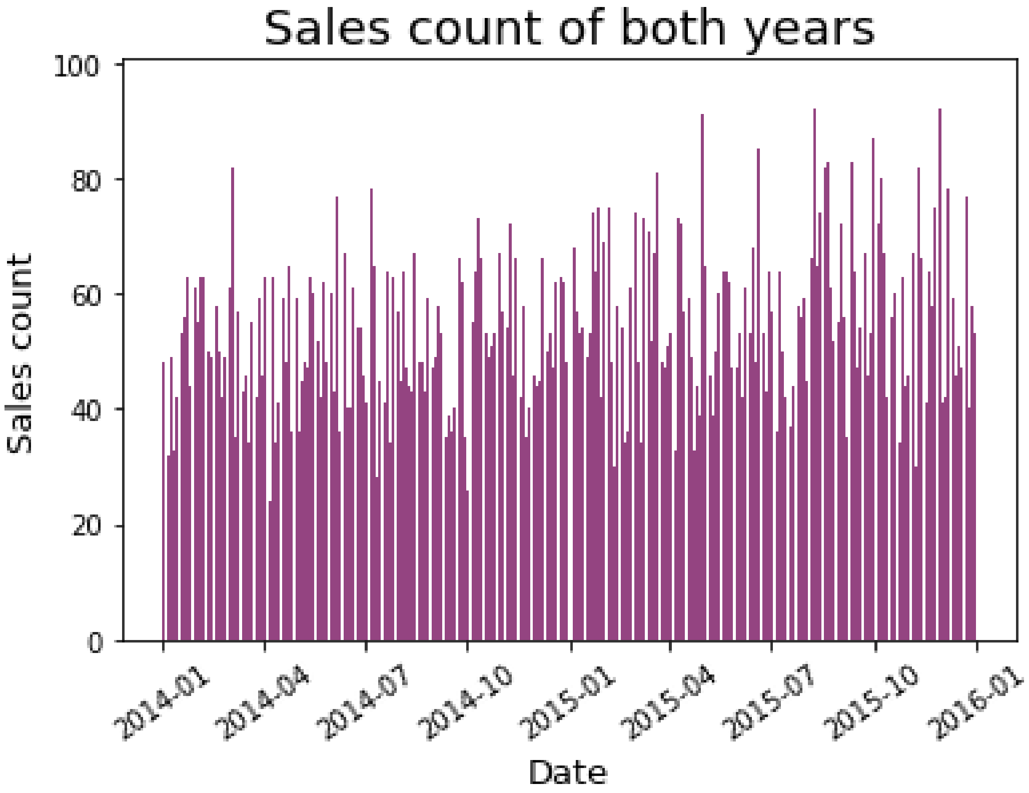
[59]:
```
sales_2015 = sales_count_by_date.loc[sales_count_by_date['Date']>=pd.
 ↪to_datetime('2015-01-01')]
```

[60]:
```
x_1 = sales_2015['Date']
y_1 = sales_2015['Sales Count']

plt.bar(x_1,y_1,color='#449194')
plt.title('Sales Count of 2015',fontsize=18)
plt.ylabel('Sales count', fontsize=13)
plt.xlabel('Months', fontsize=13)
plt.show()
```



[64]:

```
x_2 = sales_count_by_date['Date']
y_2 = sales_count_by_date['Sales Count']

plt.bar(x_2,y_2,color='#944481')
plt.title('Sales count of both years',fontsize=18)
plt.ylabel('Sales count', fontsize=13)
plt.xlabel('Date', fontsize=13)
plt.xticks(rotation=35)
plt.show()
```


Sales count of both years
Sales count
0
20
40
60
80
100
2014-01
2014-04
2014-07
2014-10
2015-01
2015-04
2015-07
2015-10
2016-01
Date