# Problem 1

***A wholesale distributor operating in different regions of Portugal has information on annual spending of several items in their stores across different regions and channels. The data consists of 440 large retailers' annual spending on 6 different varieties of products in 3 different regions (Lisbon, Oporto, Other) and across different sales channel (Hotel, Retail).***

## Question 1.1

From the above description we are expecting : 440 large retailers, Annual Spending Report based on 6 Varieties , 3 Different regions (Lisbon, Oporto, Other)and 2 Sales Channel(Hotel, Retail).

### *1.1.1 Use methods of descriptive statistics to summarize data*

**Exploratory data analysis:**

by putting df.head(),df.tail() command we can analyse first 5 rows, which can give us overall Idea about how the data is. It has 6 varieties (Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicatessen).2 Channels(Hotel, Retail) and by putting this command

"df['Region'].unique()" we will get to know the Three Regions are there:

```
array(['Other', 'Lisbon', 'Oporto'], dtype=object)
```

| Buyer/Spender | Channel | Region | Fresh | Milk | Grocery | Frozen | Detergents_Paper | Delicatessen |
|---|---|---|---|---|---|---|---|---|
| 1 | Retail | Other | 12669 | 9656 | 7561 | 214 | 2674 | 1338 |
| 2 | Retail | Other | 7057 | 9810 | 9568 | 1762 | 3293 | 1776 |
| 3 | Retail | Other | 6353 | 8808 | 7684 | 2405 | 3516 | 7844 |
| 4 | Hotel | Other | 13265 | 1196 | 4221 | 6404 | 507 | 1788 |
| 5 | Retail | Other | 22615 | 5410 | 7198 | 3915 | 1777 | 5185 |

Now by using command df.info(), we will get to know dataset has no null values, Column Fresh,Milk,Grocery Frozen Detergent_Paper and Delicatessen having integer values, int data type while Channel and Region are having string or Object non integers.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440 entries, 0 to 439
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Buyer/Spender     440 non-null    int64
 1   Channel           440 non-null    object
 2   Region            440 non-null    object
 3   Fresh             440 non-null    int64
 4   Milk              440 non-null    int64
 5   Grocery           440 non-null    int64
 6   Frozen            440 non-null    int64
 7   Detergents_Paper  440 non-null    int64
 8   Delicatessen      440 non-null    int64
dtypes: int64(7), object(2)
memory usage: 31.1+ KB
```

To re-confirm it is good habit to use command df.isnull.sum(), it will print total sum of null elements present in particular column, so it is confirmed that data has no null values.

```
df.isnull().sum()
```

```
Buyer/Spender       0
Channel             0
Region              0
Fresh               0
Milk                0
Grocery             0
Frozen              0
Detergents_Paper    0
Delicatessen        0
dtype: int64
```

Descriptive Analysis: It is used to get the idea about how data is distributed, mainly Mean, Median, Mode and standard deviation. 25%,50%, 75% data distribution and Interquirtile ranges.we can get this by giving command as df.describe()

```
df.describe()
```

| | Buyer/Spender | Fresh | Milk | Grocery | Frozen | Detergents_Paper | Delicatessen |
|---|---|---|---|---|---|---|---|
| **count** | 440.000000 | 440.000000 | 440.000000 | 440.000000 | 440.000000 | 440.000000 | 440.000000 |
| **mean** | 220.500000 | 12000.297727 | 5796.265909 | 7951.277273 | 3071.931818 | 2881.493182 | 1524.870455 |
| **std** | 127.161315 | 12647.328865 | 7380.377175 | 9503.162829 | 4854.673333 | 4767.854448 | 2820.105937 |
| **min** | 1.000000 | 3.000000 | 55.000000 | 3.000000 | 25.000000 | 3.000000 | 3.000000 |
| **25%** | 110.750000 | 3127.750000 | 1533.000000 | 2153.000000 | 742.250000 | 256.750000 | 408.250000 |
| **50%** | 220.500000 | 8504.000000 | 3627.000000 | 4755.500000 | 1526.000000 | 816.500000 | 965.500000 |
| **75%** | 330.250000 | 16933.750000 | 7190.250000 | 10655.750000 | 3554.250000 | 3922.000000 | 1820.250000 |
| **max** | 440.000000 | 112151.000000 | 73498.000000 | 92780.000000 | 60869.000000 | 40827.000000 | 47943.000000 |

By above table we get the knowledge of how this data is distributed. we can say that Buyer/Spender column has no as such significance we can remove it if we want. It is redundant. From above table we can see that Fresh has highest spending whereas Detergents_Paper has lowest Spending.

For this we can import seaborn library which is used as top of Matplotlib for better visualiation.

### *1.1.2. Which Region and which Channel seems to spend more? Which Region and which Channel seems to spend less?-->*

To get overall idea about regions : 3 regions

```
df['Region'].unique()
```

```
array(['Other', 'Lisbon', 'Oporto'], dtype=object)
```

To get Idea about Channels: 2 channels(Hotel/Retail)

To answer the question first we will create a dataframe which will tell us about region vise and Variety vise spending.

```
df_Region=df.groupby('Region').sum()
df_Region
```

| Region | Buyer/Spender | Fresh | Milk | Grocery | Frozen | Detergents_Paper | Delicatessen |
|---|---|---|---|---|---|---|---|
| Lisbon | 18095 | 854833 | 422454 | 570037 | 231026 | 204136 | 104327 |
| Oporto | 14899 | 464721 | 239144 | 433274 | 190132 | 173311 | 54506 |
| Other | 64026 | 3960577 | 1888759 | 2495251 | 930492 | 890410 | 512110 |

df_Region=df.groupby('Region').sum()
df_Region

In this code “df.groupby('Region')” will filter/group table on the basis of 3 region. Sum() will add all spendings within that region.

We can see Buyer/Spender column has no significance, it is random serial number, so we will try to omit it.

```
column_list = list(df_Region)
column_list.remove('Buyer/Spender')
column_list
```

```
['Fresh', 'Milk', 'Grocery', 'Frozen', 'Detergents_Paper', 'Delicatessen']
```

column_list = list(df_Region) → Pulling out column list and storing into variable column_list.

column_list.remove('Buyer/Spender')→ Using .remove() function to remove 'Buyer/Spender column.

column_list→ Again Printing column list which do not have column'Buyer/Spender'.

Now, Furthermore df[column_list] will print below result it has omitted 'Buyer/Spender column.

| Region | Fresh | Milk | Grocery | Frozen | Detergents_Paper | Delicatessen |
|---|---|---|---|---|---|---|
| **Lisbon** | 854833 | 422454 | 570037 | 231026 | 204136 | 104327 |
| **Oporto** | 464721 | 239144 | 433274 | 190132 | 173311 | 54506 |
| **Other** | 3960577 | 1888759 | 2495251 | 930492 | 890410 | 512110 |

df_Region[column_list].sum(axis=0)

it will print sum of all the elements along Y axis.

```
Fresh              5280131
Milk               2550357
Grocery            3498562
Frozen             1351650
Detergents_Paper   1267857
Delicatessen        670943
dtype: int64
```

df_Region[column_list].sum(axis=1)

it will print sum of all the elements along X axis.

```
Region
Lisbon      2386813
Oporto      1555088
Other      10677599
dtype: int64
```

df_Region["sum"]=df_Region[column_list].sum(axis=1)

df_Region

This two lines are adding separate Sum column and prints table with addition

| Region | Buyer/Spender | Fresh | Milk | Grocery | Frozen | Detergents_Paper | Delicatessen | sum |
|---|---|---|---|---|---|---|---|---|
| Lisbon | 18095 | 854833 | 422454 | 570037 | 231026 | 204136 | 104327 | 2386813 |
| Oporto | 14899 | 464721 | 239144 | 433274 | 190132 | 173311 | 54506 | 1555088 |
| Other | 64026 | 3960577 | 1888759 | 2495251 | 930492 | 890410 | 512110 | 10677599 |

To get Maximum value Regionwise, which region has Highest spending among all

```
Regionwise_max=pd.DataFrame(data=df_Region, columns=['sum'] )
Regionwise_max[Regionwise_max['sum'] == Regionwise_max['sum'].max()]
```

Now we are Particularly Intersting in Region and Sum column. So line 1 will store sum values and Region column. Next task is Print only that row which is having maximum value.

| Region | sum |
|---|---|
| Other | 10677599 |

In “Other” region highest Spending Occurred.

The below lines will print Regionwise_min table, by comparing to minimum value with region and sum.

```
Regionwise_min=Regionwise_max[Regionwise_max['sum'] == Regionwise_max['sum'].min()]
Regionwise_min
```

| Region | sum |
|---|---|
| Oporto | 1555088 |

“Oporto” region seems to spend less

Now we are focussing upon channel, Maximum and Minimum Spendings. The below code will print the table which contains minimum and maximum value with Separate column Sum.

```
# Print dataframe, where addition done for different channels with different Varieties.
df_channel=df.groupby('Channel').sum()
df_channel

# From the Dataframe generated above, we are removing Buyer/Spender column as it is not useful here
list2=list(df_channel)
list2.remove('Buyer/Spender')
list2

# Passing the Above created list to Dataframe df_Channel and addition will be done by using sum function on dataframe
df_channel2=df_channel[list2]
df_channel2['sum']=df_channel2.sum(axis=1)
df_channel2
```

| Channel | Fresh | Milk | Grocery | Frozen | Detergents_Paper | Delicatessen | sum |
|---|---|---|---|---|---|---|---|
| **Hotel** | 4015717 | 1028614 | 1180717 | 1116979 | 235587 | 421955 | 7999569 |
| **Retail** | 1264414 | 1521743 | 2317845 | 234671 | 1032270 | 248988 | 6619931 |

Now we are Interested in the channel which is Spending Maximum.

The below code will print Output of Channel which has highest spendings. First we have created df_channel3 which stores table which has Highest addition by using max function.

df_channel4 will take input as df_channel3 and it will show the Column Sum

```
df_channel3=df_channel2[df_channel2['sum']==df_channel2['sum'].max()]
df_channel4=pd.DataFrame(data= df_channel3,columns=['sum'])
df_channel4
```

| Channel | sum |
|---|---|
| **Hotel** | 7999569 |

Hotel is the channel which has Highest Spending.

Now we are Interested in the channel which has lowest Spending. We will just put min function to the dataframe, and it will print the output what we want.

```
# Channel which has Lowest Spending
df_channel5=df_channel2[df_channel2['sum']==df_channel2['sum'].min()]
df_channel6=pd.DataFrame(data= df_channel5,columns=['sum'])
df_channel6
```

| Channel | sum |
|---|---|
| Retail | 6619931 |

Retail is the channel which has Lowest spending amongst Channel.

## Question1.2 :

There are 6 different varieties of items are considered. Do all varieties show similar behaviour across Region and Channel? Provide justification for your answer.

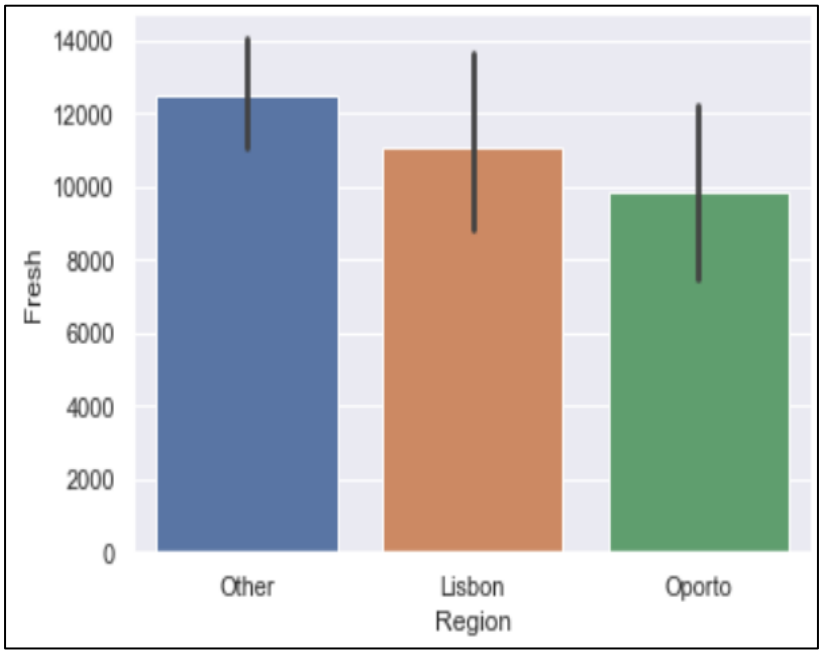
To Answer this question, we can plot a barplot(To get statistica difference between mean) and boxplot(To get the distribution idea,Outliers,mean), which will tell us the relations between each variety and Region, Each variety and Channel.

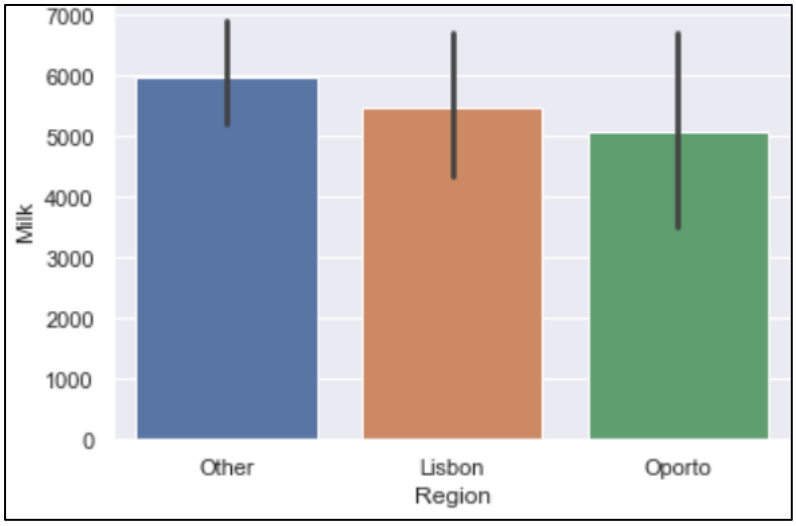
First we will try to create a desired dataframe, by dropping column(Buyer/Spender) and store this in new dfd datafrme.

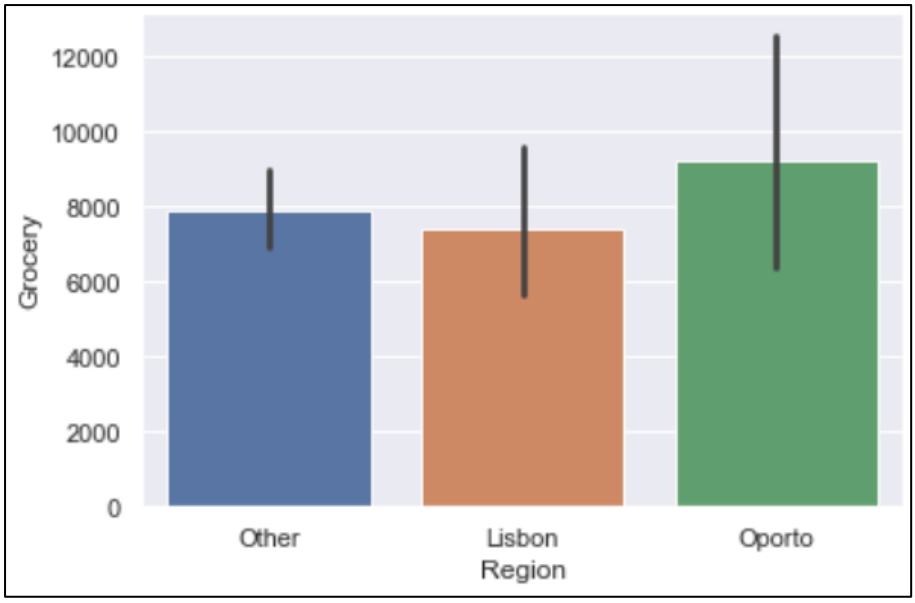
```
dfd=df.drop(['Buyer/Spender'], axis='columns')
```

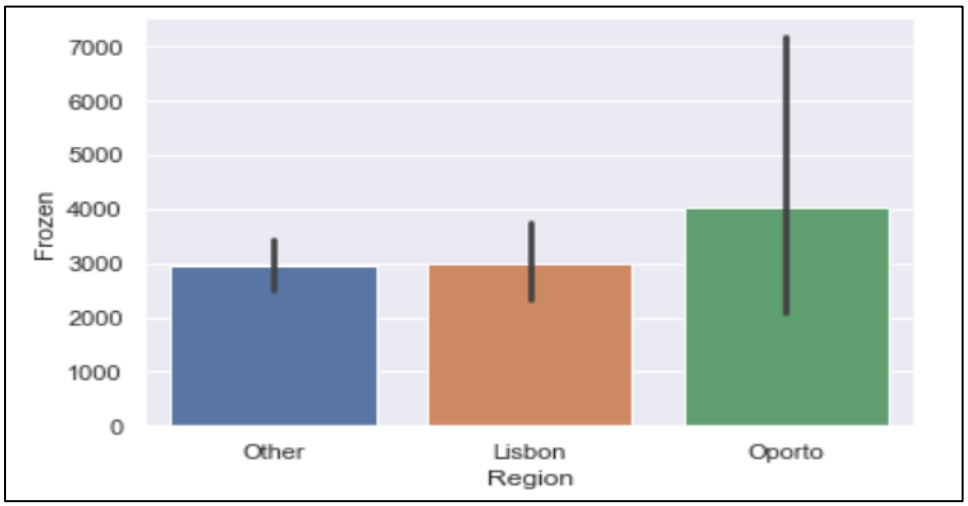
Now again we will introduce for loop to print all barplots Region vs Varieties.It will iterate through all columns.

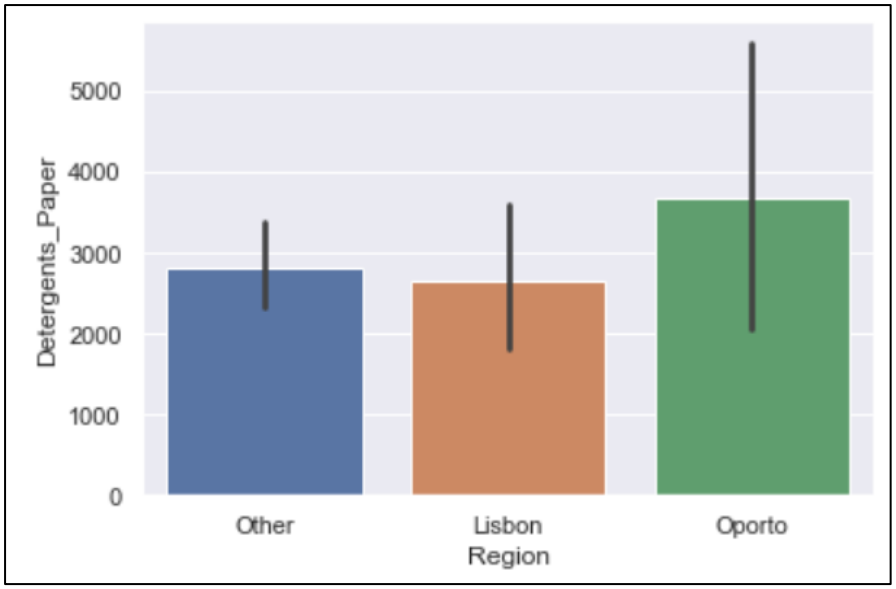
```
# ALl Varieties Across Region
for i in dfd.columns:
    if dfd[i].dtypes != "object":
        sns.barplot(df['Region'],dfd[i])
        plt.show()
```

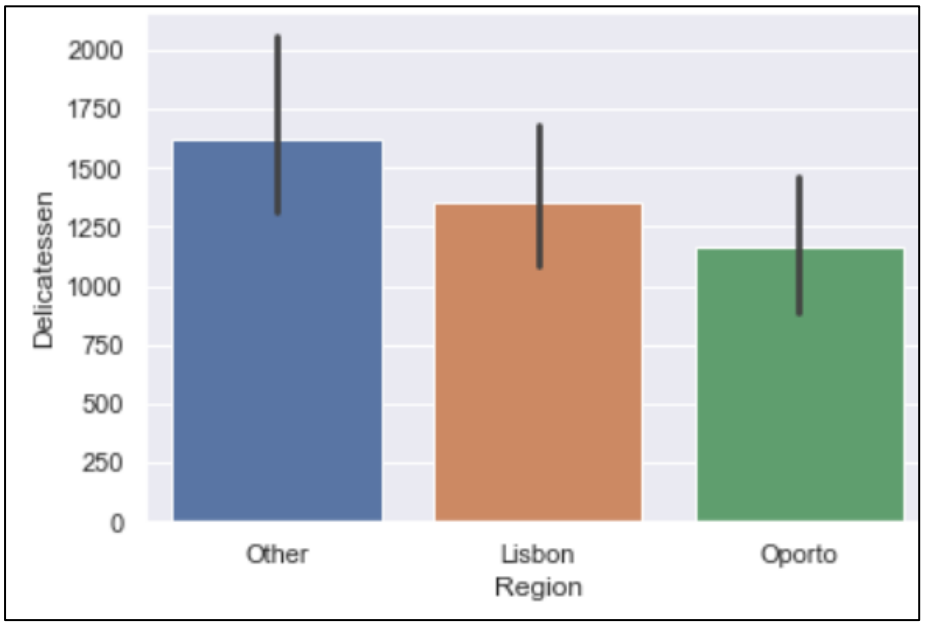



12000
10000
8000
6000
4000
2000
0
Grocery
Other
Lisbon
Oporto
Region


7000
6000
5000
4000
3000
2000
1000
0
Frozen
Other
Lisbon
Oporto
Region




If we see Graph1 for 'Fresh' variety - Other Region has highest mean value so it is highly bought variety as comapre to Lisbon and Oporto region.This variety is not showing similar nature in case of average selling.

If we see Graph2 for 'Milk' variety - Other Region has highest mean value so it is highly bought variety as comapre to Lisbon and Oporto region.This variety is not showing similar nature in case of average selling.

If we see Graph3 for 'Grocery' variety - Oporto Region has highest mean value so it is highly bought variety as comapre to Lisbon and Oporto region.This variety is not showing similar nature in case of average selling.

If we see Graph4 for 'Frozen' variety - Oporto Region has highest mean value so it is highly bought variety as comapre to Lisbon and Oporto region.This variety is not showing similar nature in case of average selling.

If we see Graph5 for 'Detergent_paper' variety - Oporto Region has highest mean value so it is highly bought variety as comapre to Lisbon and Oporto region.This variety is not showing similar nature in case of average selling.

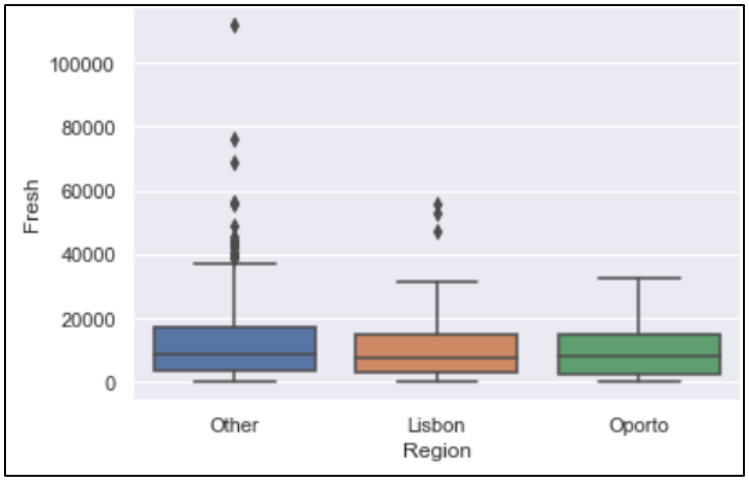
If we see Graph6 for 'Delicatessen' variety - Other Region has highest mean value so it is highly bought variety as comapre to Lisbon and Oporto region.This variety is not showing similar nature in case of average selling.

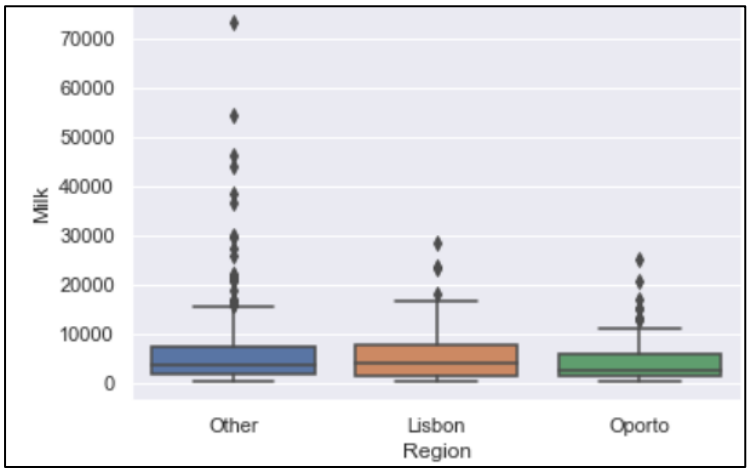
The similarity can be found is , In Nutshell, Fresh,Milk and Delicatessen are the varieties gives highest average income from Other Region and Grocery, Detergent_papers and Frozen varieties are giving highest revenue from Oporto region

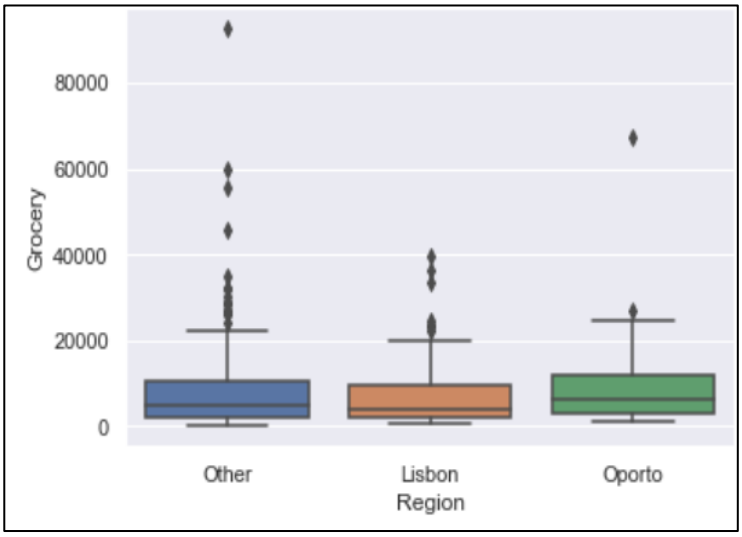
Lets plot Boxplot, so that we will deep dive into regionwise distribution. Again we will use for loop and plot as boxplot.

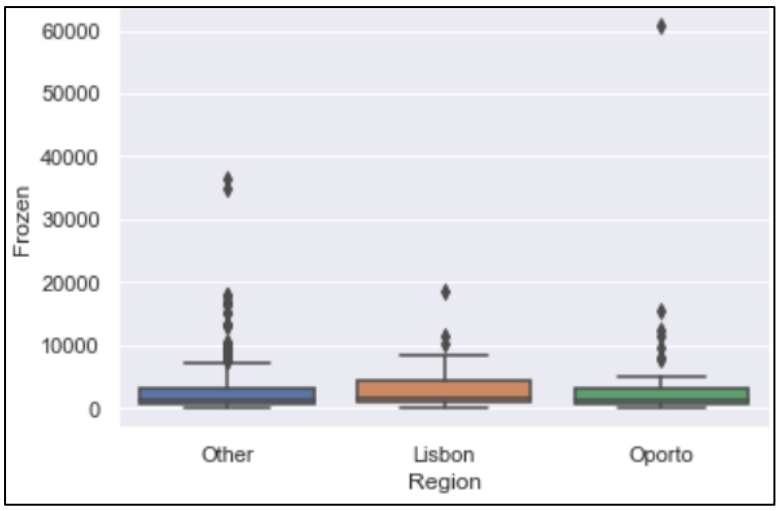
```
for i in dfd.columns:
    if df[i].dtypes != "object":
        sns.boxplot(df['Region'],df[i])
        plt.show()
```

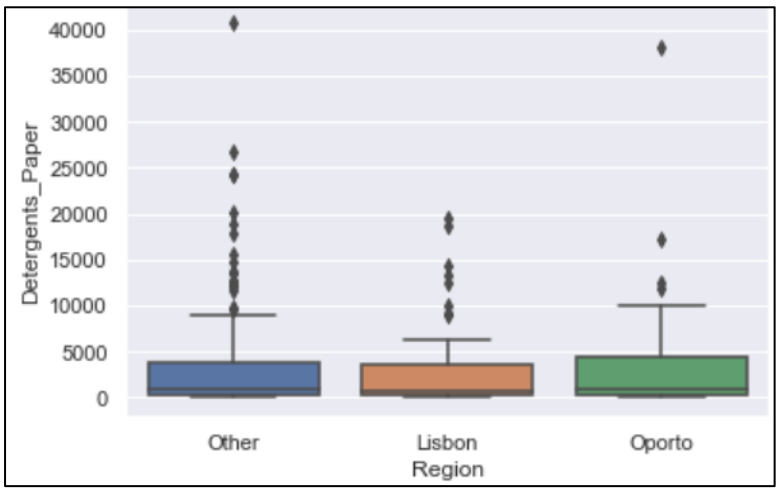
Plot can be seen for all 6 varieties and 3 regions:

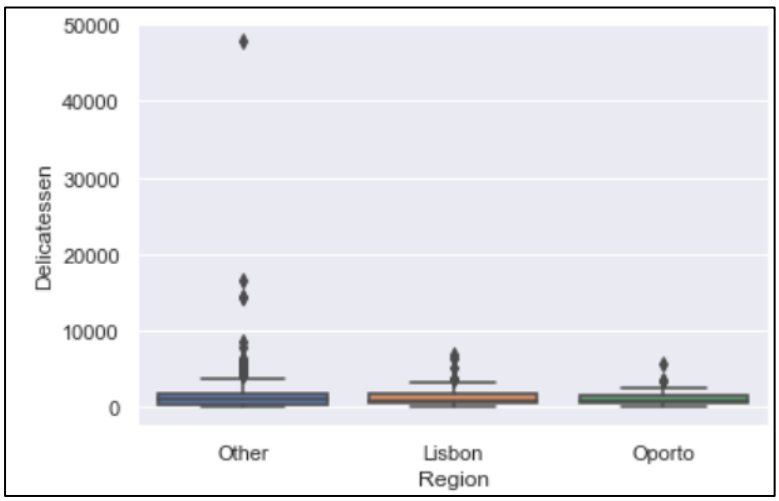

100000
80000
60000
40000
20000
0
Fresh
Other
Lisbon
Oporto
Region


70000
60000
50000
40000
30000
20000
10000
0
Milk
Other
Lisbon
Oporto
Region


80000
60000
40000
20000
0
Grocery
Other
Lisbon
Oporto
Region



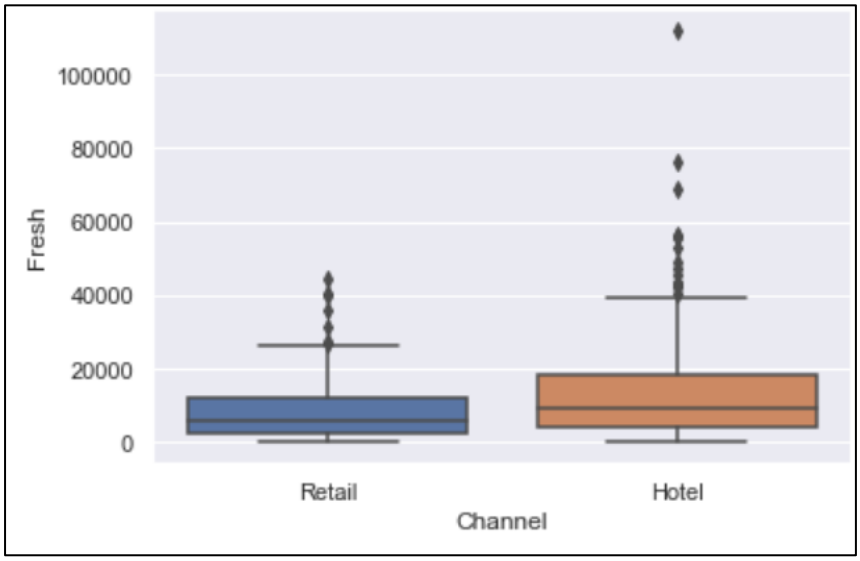



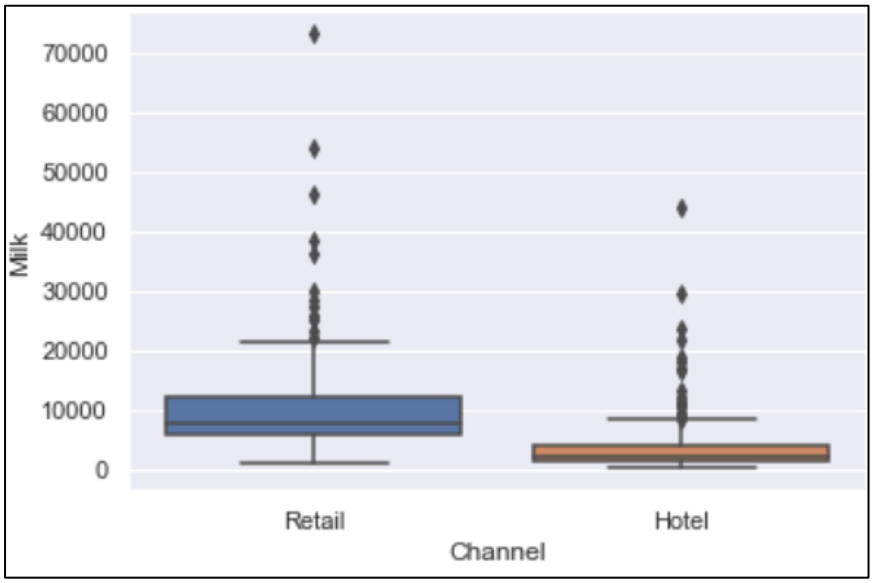

Only one Combination which is "Oporto and Fresh" has no outliers and data is consistent.

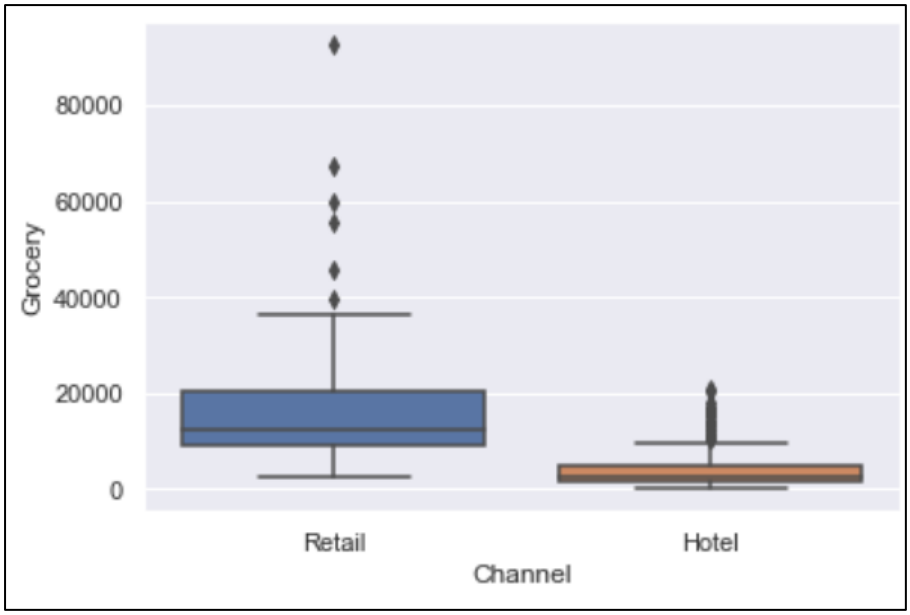
While other all Varieties across region has Outliers.

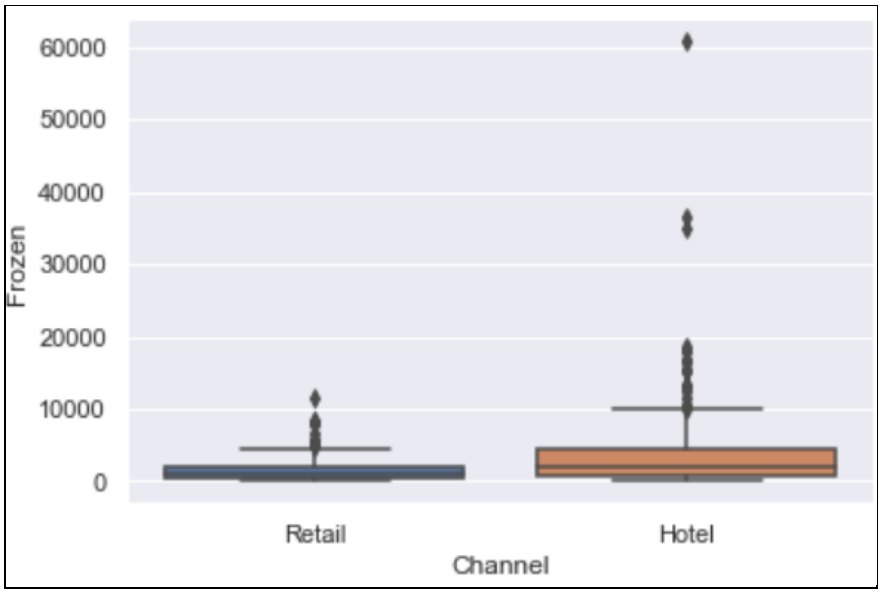
Now lets plot Channel vise boxplot :

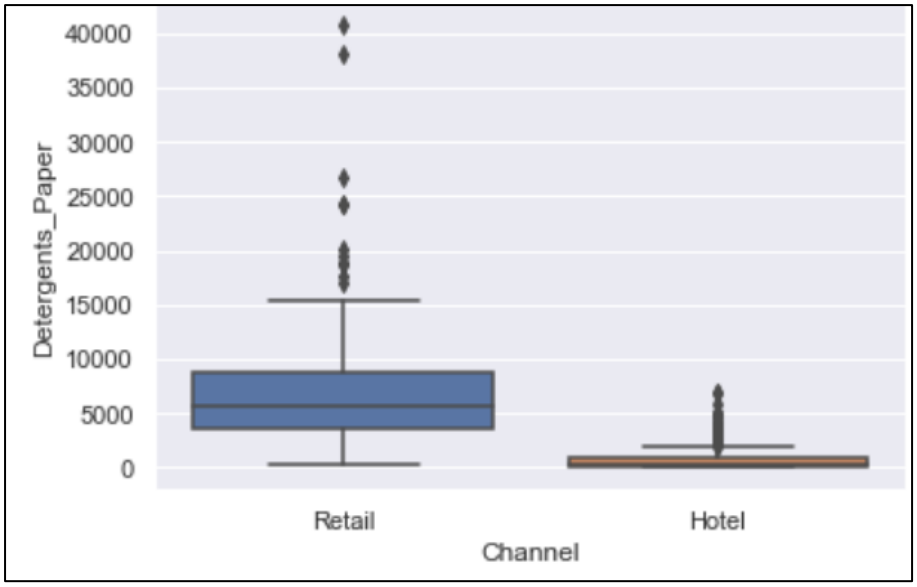
```
for i in dfd.columns:
    if df[i].dtypes != "object":
        sns.boxplot(df['Channel'],df[i])
        plt.show()
```

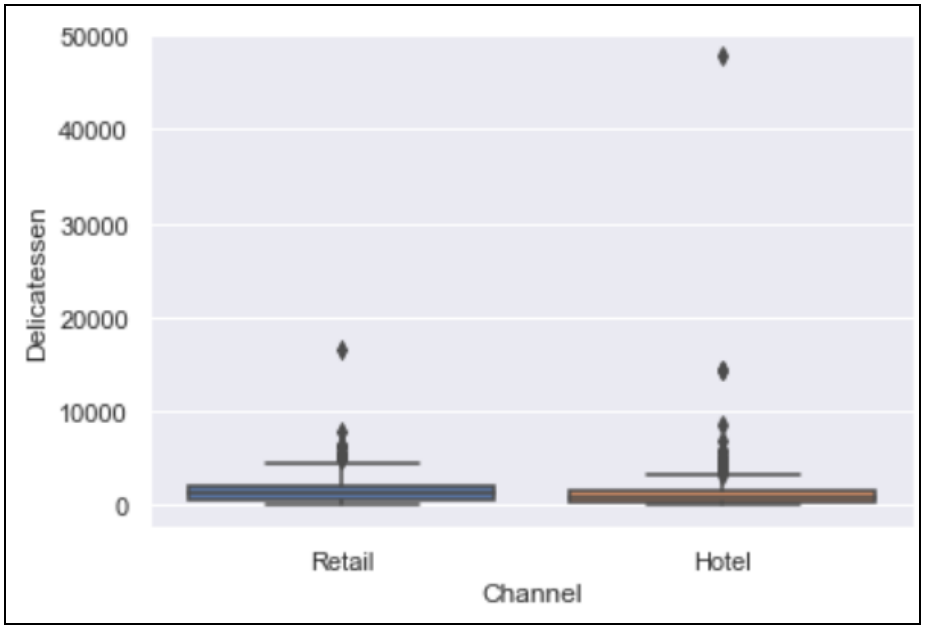



80000
60000
40000
20000
0
Grocery
Retail
Hotel
Channel


60000
50000
40000
30000
20000
10000
0
Frozen
Retail
Hotel
Channel



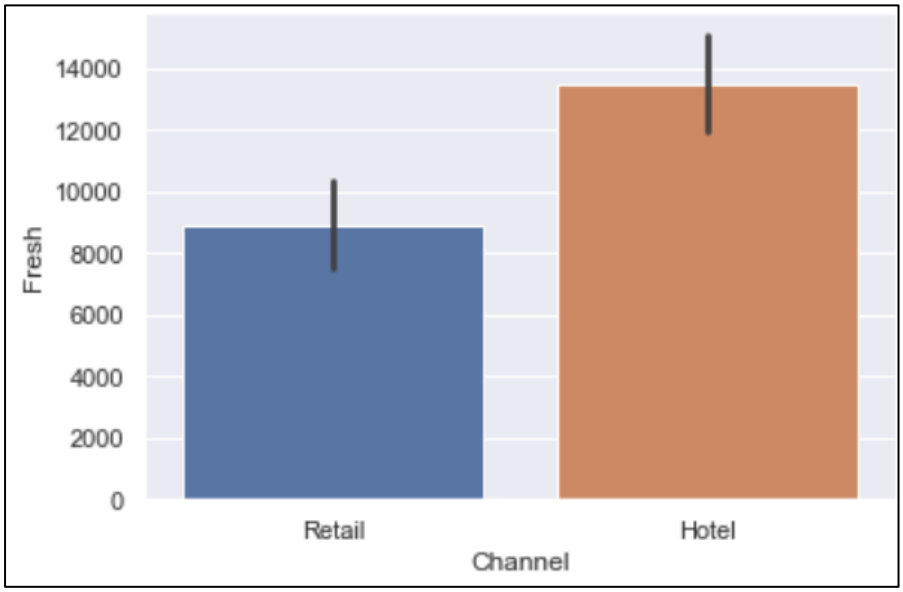

From plot 1, here it can be noticed that every variety has outliers across Channel and Fresh items are highly sold in Hotels than Retails, and diiference is significant.

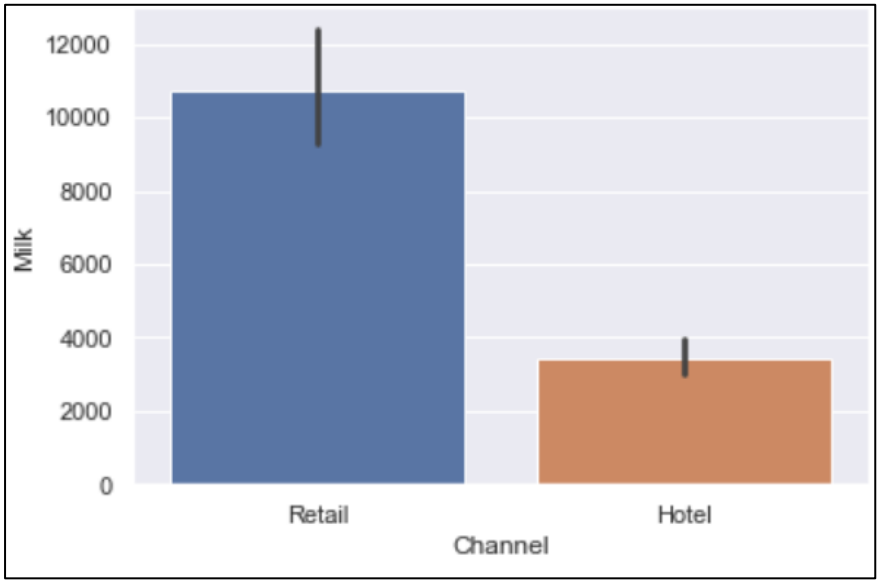
Milk, Grocery and detergent paper generating High revenues in retail sector as comapare to Hotel channel.

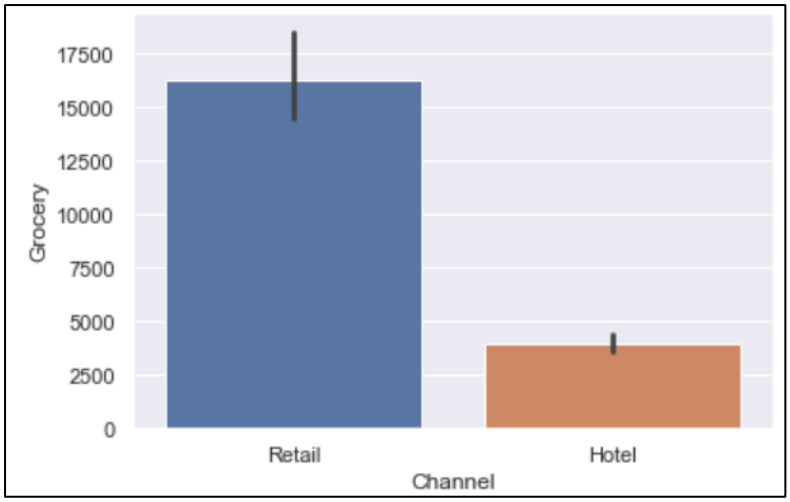
Delicatessen is the variety having almost similar distribution in Both the channels.

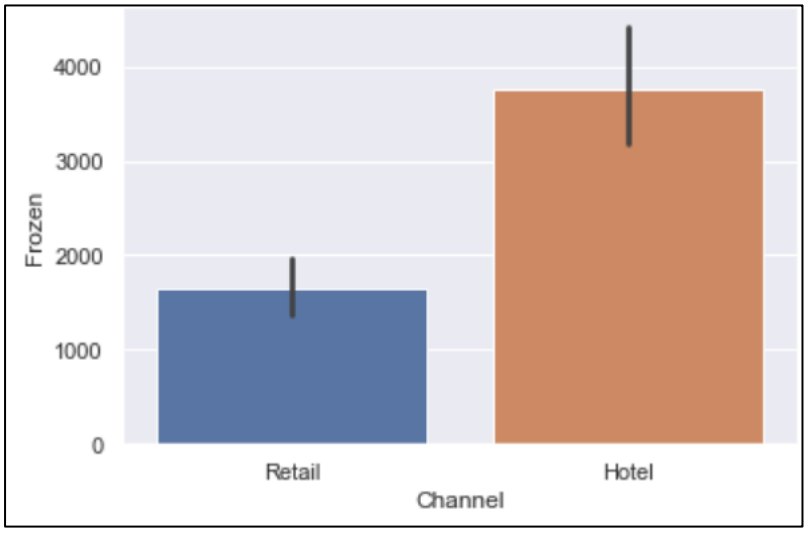
Now lets plot Channel vise barplot, to compare mean by visualization

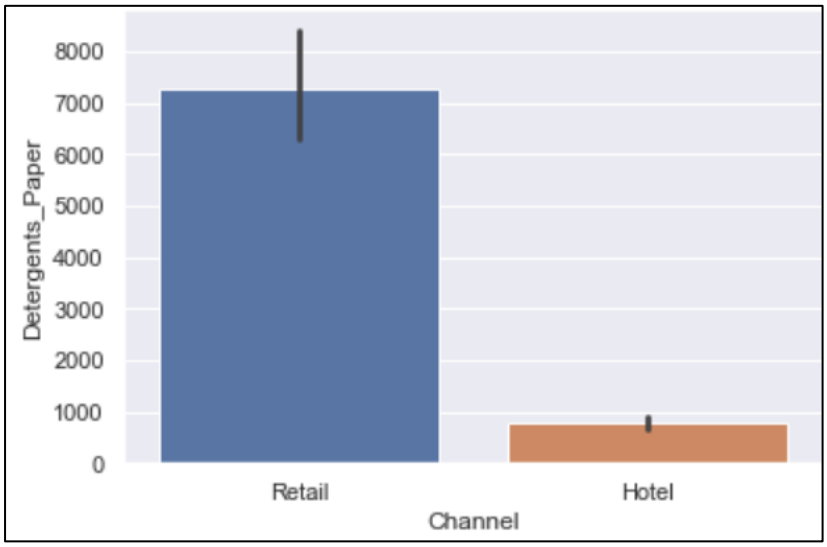
```
# All varieties across Channel.

for i in dfd.columns:
    if dfd[i].dtypes != "object":
        sns.barplot(df['Channel'],df[i])
        plt.show()
```

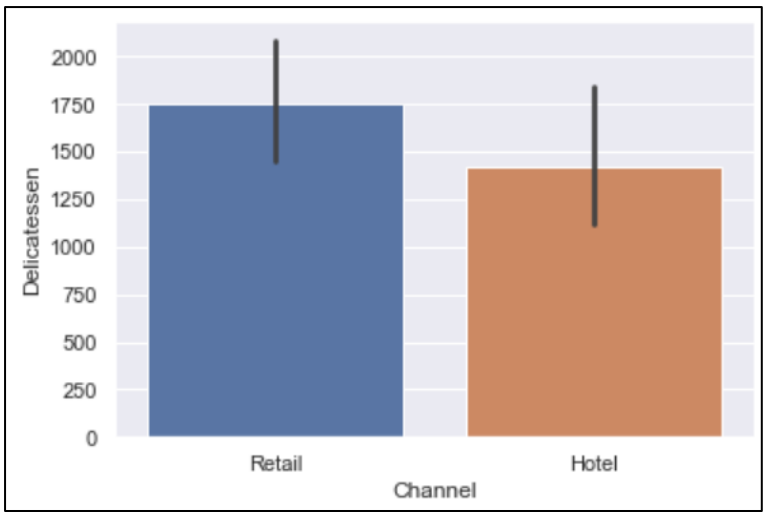



17500
15000
12500
10000
7500
5000
2500
0
Grocery
Retail
Hotel
Channel


4000
3000
2000
1000
0
Frozen
Retail
Hotel
Channel


8000
7000
6000
5000
4000
3000
2000
1000
0
Detergents_Paper
Retail
Hotel
Channel




If we see Fresh and Frozen in channel Hotel has the maximum average values over Channel Retail and Delicatessen,Detergent_paper,Milk,Grocery has highest average spendings in Retail Channel thats the inter Similarity we can get.

# Question 1.3

1.3 On the basis of a descriptive measure of variability, which item shows the most inconsistent behaviour? Which items show the least inconsistent behaviour?

```
df_del=df.drop(['Buyer/Spender','Channel','Region'], axis = 1)
df_del.std()
```

from the below Output Fresh has the highest Standard deviation , hence it shows Highest inconsistency whereas Delicatessen has the lowest Standard deviation , hence it shows least inconsistent behaviour.

```
Fresh              12647.328865
Milk                7380.377175
Grocery             9503.162829
Frozen              4854.673333
Detergents_Paper    4767.854448
Delicatessen        2820.105937
dtype: float64
```

```
df_del.var()
```

from the below Output Fresh has the highest variance , hence it shows Highest inconsistency whereas Delicatessen has the lowest variance , hence it shows least inconsistent behaviour.

```
Fresh               1.599549e+08
Milk                5.446997e+07
Grocery             9.031010e+07
Frozen              2.356785e+07
Detergents_Paper    2.273244e+07
Delicatessen        7.952997e+06
dtype: float64
```

# Question 1.4

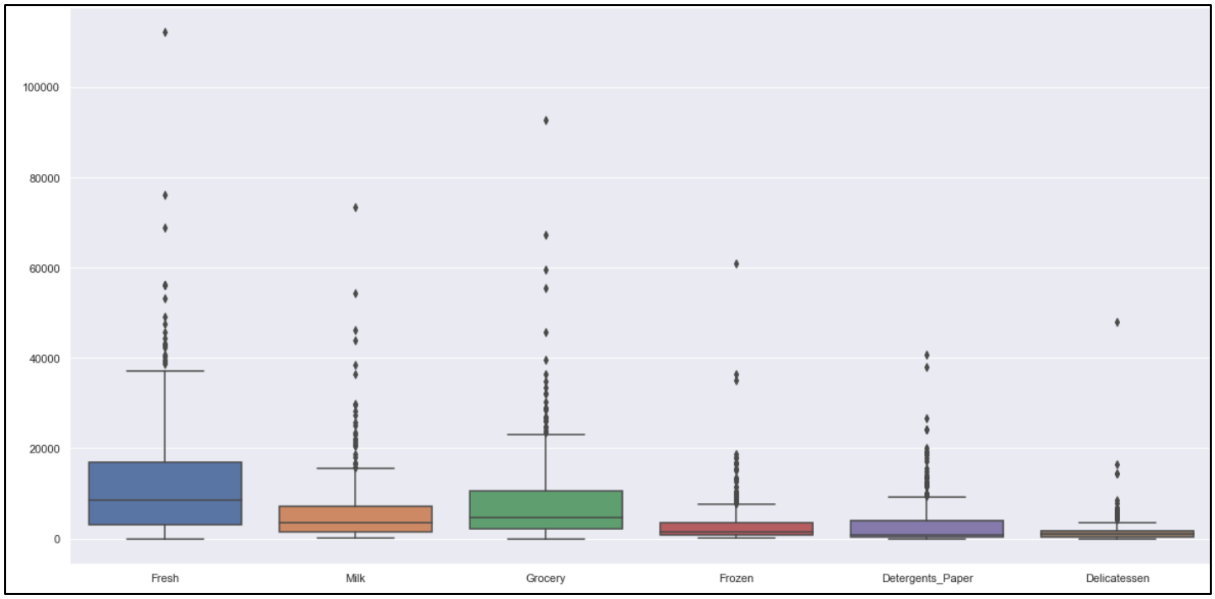
Are there any outliers in the data?

Here to get the dea of outliers we will plot a barplot for each variety.

From below boxplots, we can say that every variety has Outlier

```
plt.figure(figsize=(20,10))
sns.boxplot(data=dfd)
```

By observing below plot we can say that Yes All Varieties: Fresh, Milk, Frozen, Grocery, Detergents_paper, Delicatessen  have outliers.

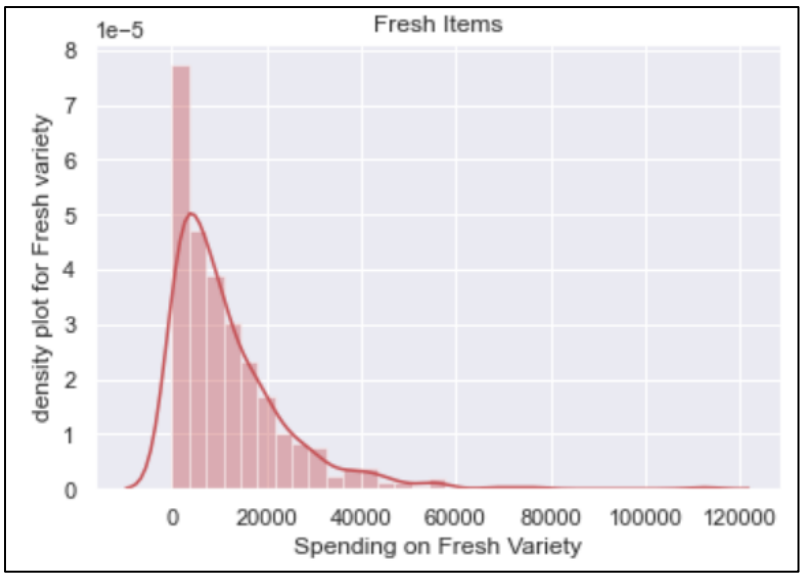



# Question 1.5

On the basis of your analysis, what are your recommendations for the business? How can your analysis help the business to solve its problem? Answer from the business perspective

Now for clear understanding of Distribution we can plot 6 distribuion plots for each variety having continuous distribution. For this we can import seaborn library which is used as top of Matplotlib for better visualiation.

```
# From this below line, we are defining color palete we have used further for different colors R for Red
sns.set(color_codes=True)

# Here we are plotting distplot and assigning it to variale dist, color Red.
dist=sns.distplot(df['Fresh'],color='R')

# Setting Title to the graph as Fresh Items
dist.set_title('Fresh Items')

# Setting x label to the graph as Spending on Fresh Variety
dist.set_xlabel('Spending on Fresh Variety')

# Setting y label to the graph as density plot for Fresh variety
dist.set_ylabel('density plot for Fresh variety')
```

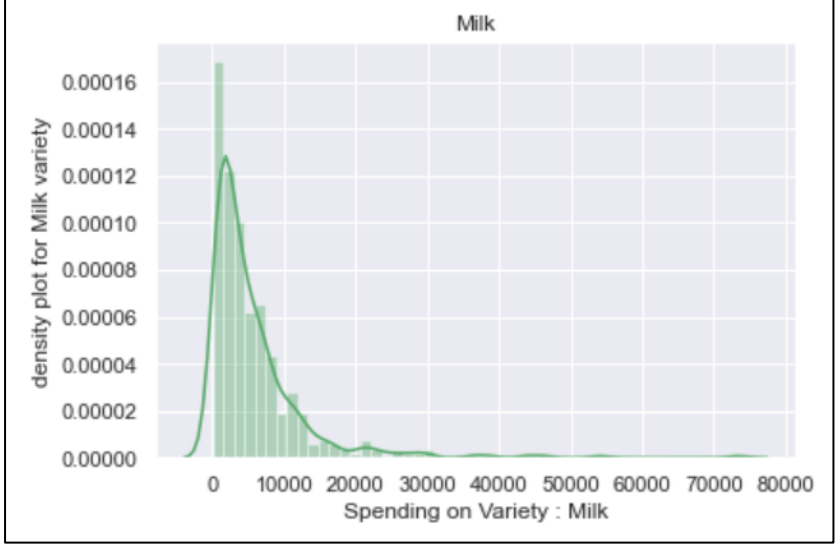



Maximum data points accumulated spendings in between 0 to 40000, very less buyer or Spender spent money on Fresh variety greater than 40000 and distribution is right skewed.

Similarly we can have distplot for each variety,

```
sns.set(color_codes=True)
dist=sns.distplot(df['Milk'],color='g')
dist.set_title('Milk')
dist.set_xlabel('Spending on Variety : Milk')
dist.set_ylabel('density plot for Milk variety')
```

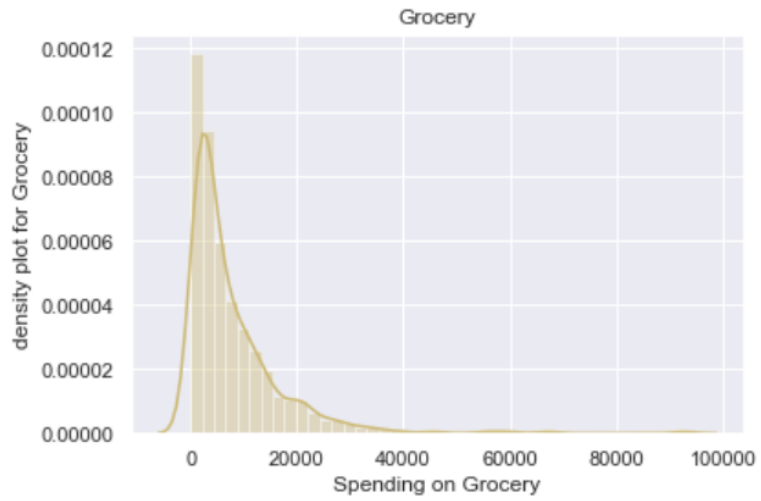
```
Text(0, 0.5, 'density plot for Milk variety')
```




Maximum data points accumulated spendings in between 0 to 30000, very less buyer or Spender spent money on Milk greater than 30000 and distribution is right skewed.

```
sns.set(color_codes=True)
dist=sns.distplot(df['Grocery'],color='y')
dist.set_title('Grocery')
dist.set_xlabel('Spending on Grocery')
dist.set_ylabel('density plot for Grocery')
```

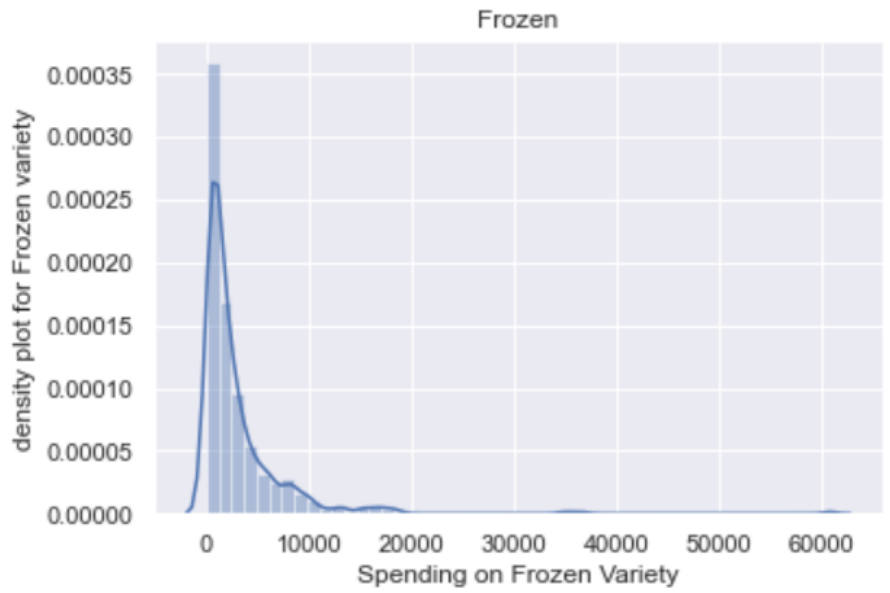
```
Text(0, 0.5, 'density plot for Grocery')
```




Maximum data points accumulated spendings in between 0 to 30000, very less buyer or Spender spent money on Grocery greater than 30000 and distribution is right skewed.

```
sns.set(color_codes=True)
dist=sns.distplot(df['Frozen'],color='b')
dist.set_title('Frozen')
dist.set_xlabel('Spending on Frozen Variety')
dist.set_ylabel('density plot for Frozen variety')
```

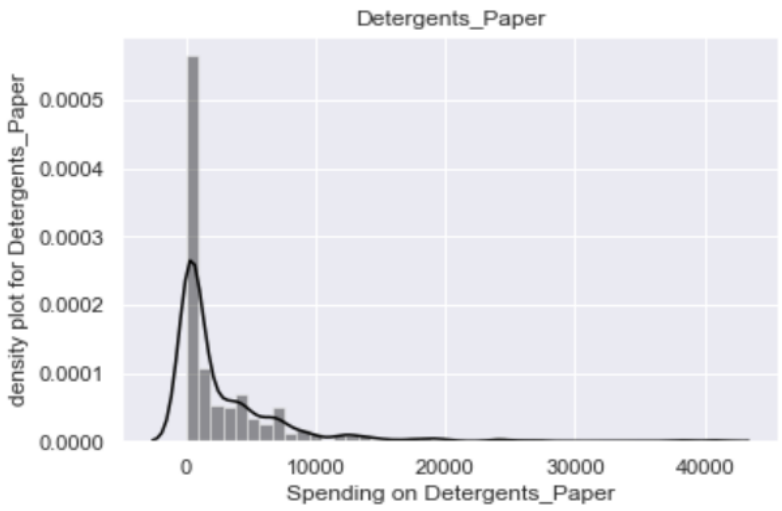
```
Text(0, 0.5, 'density plot for Frozen variety')
```




Maximum data points accumulated spendings in between 0 to 10000, very less buyer or Spender spent money on Frozen items greater than 10000 and distribution is right skewed.

```
sns.set(color_codes=True)
dist=sns.distplot(df['Detergents_Paper'],color='black')
dist.set_title('Detergents_Paper')
dist.set_xlabel('Spending on Detergents_Paper')
dist.set_ylabel('density plot for Detergents_Paper')
```

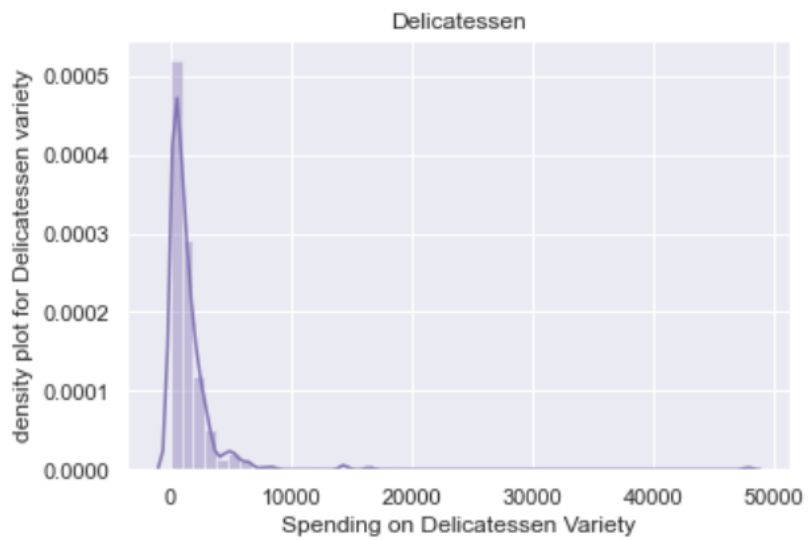
```
Text(0, 0.5, 'density plot for Detergents_Paper')
```




Maximum data points accumulated spendings in between 0 to 10000, very less buyer or Spender spent money on Detergent_Paper greater than 10000 and distribution is right skewed.

```
sns.set(color_codes=True)
dist=sns.distplot(df['Delicatessen'],color='M')
dist.set_title('Delicatessen')
dist.set_xlabel('Spending on Delicatessen Variety')
dist.set_ylabel('density plot for Delicatessen variety')
```

```
Text(0, 0.5, 'density plot for Delicatessen variety')
```




Maximum data points accumulated spendings in between 0 to 9000, very less buyer or Spender spent money on Delicatessen variety greater than 9000 and distribution is right skewed.

From Above graphs, Overall Revenue for varieties and its production, Production should be high in case of Milk,Grocery,Fresh Items and also it will increase chances of Revenues and for Frozen and Detergents_Paper production should be moderate to avoid initial investment issues. And Delicatessen has low revenues, so to be in the business we should produce this in limited quantity. More Focus should be given on Milk,Grocery,Fresh till some saturation.

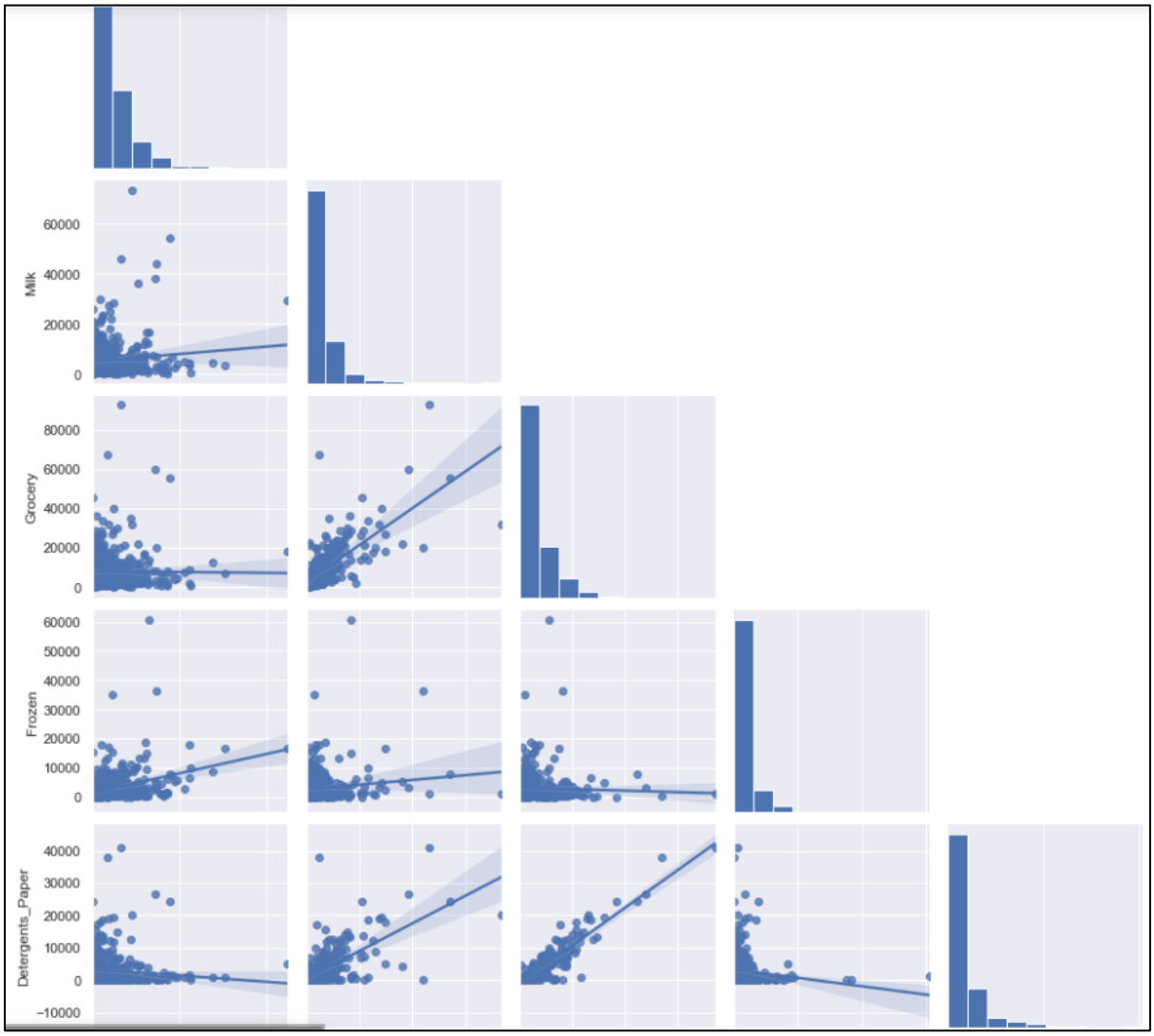
We can also consider some branding factor and usability of varieties such as Delicatessen, Frozen and Detergents_Paper. by collecting data, we can decide for more branding, Marketing to improve sales.

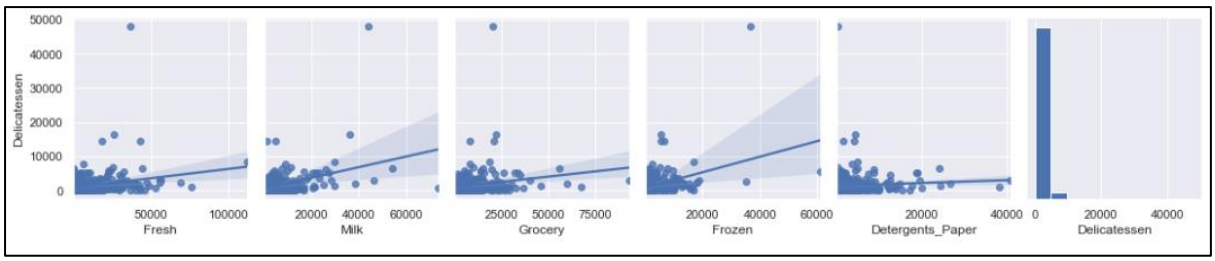
Now, we will try to plot pie charts/pairplots to see spendings,relations, the column Buyer/Spender is insignificant here so we will drop it by using drop command.

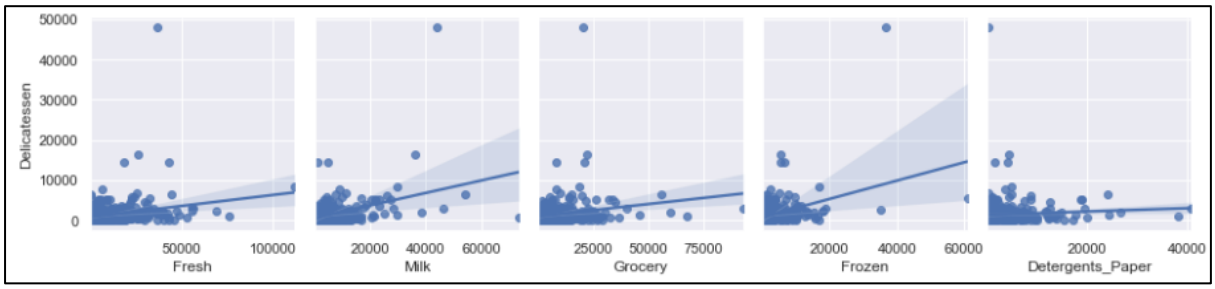
```
df_drop=df.drop(['Buyer/Spender','Channel','Region'],axis='columns')
df_drop
```

To plot pairplot amongst all variables, we will use below code:

```
sns.pairplot(data=df_drop,corner=True,kind='reg')
```






We can find correlation between varieties by using corr() function.

```
df_drop.corr()
```

| | Fresh | Milk | Grocery | Frozen | Detergents_Paper | Delicatessen |
|---|---|---|---|---|---|---|
| **Fresh** | 1.000000 | 0.100510 | -0.011854 | 0.345881 | -0.101953 | 0.244690 |
| **Milk** | 0.100510 | 1.000000 | 0.728335 | 0.123994 | 0.661816 | 0.406368 |
| **Grocery** | -0.011854 | 0.728335 | 1.000000 | -0.040193 | 0.924641 | 0.205497 |
| **Frozen** | 0.345881 | 0.123994 | -0.040193 | 1.000000 | -0.131525 | 0.390947 |
| **Detergents_Paper** | -0.101953 | 0.661816 | 0.924641 | -0.131525 | 1.000000 | 0.069291 |
| **Delicatessen** | 0.244690 | 0.406368 | 0.205497 | 0.390947 | 0.069291 | 1.000000 |

In above correlation table,the highest correlation is between Grocery and Detergents_Paper, so it can be assumed that buyer buying Detergent_paper also buy Grocery, chances of this combination is high and folllows linear relation and directly proportional.Similarly Milk and grocery has correlation index 0.7283 which is also good, so production of these items should be increased to increase Revenue.

Now we will print addition, and filter dataset by region.

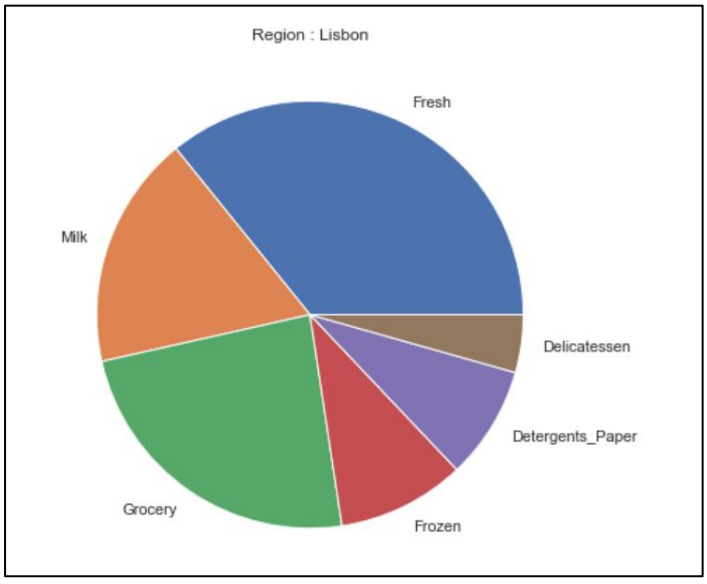
```
dfd1=dfd.groupby('Region').sum()
dfd1
```

| Region | Fresh | Milk | Grocery | Frozen | Detergents_Paper | Delicatessen |
|---|---|---|---|---|---|---|
| **Lisbon** | 854833 | 422454 | 570037 | 231026 | 204136 | 104327 |
| **Oporto** | 464721 | 239144 | 433274 | 190132 | 173311 | 54506 |
| **Other** | 3960577 | 1888759 | 2495251 | 930492 | 890410 | 512110 |

lets compare the total Regionwise spendings,lets plot for Region: Lisbon

To select row named as Lisbon, we use iloc function

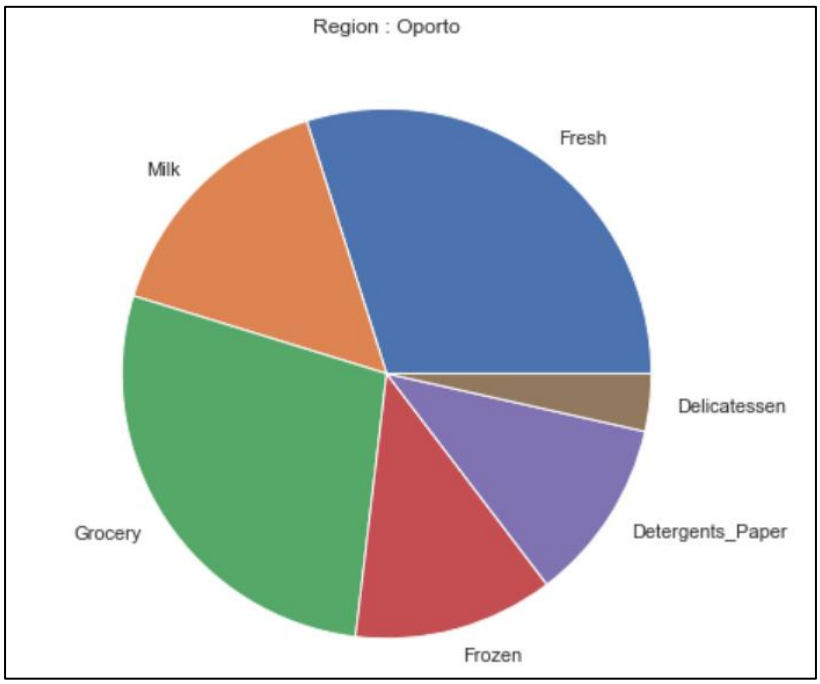
```
# To select row named as Lisbon, we use iloc function
dfd2=dfd1.iloc[0,:]
dfd2

fig = plt.figure(figsize =(10, 7))
varieties=['Fresh','Milk','Grocery','Frozen','Detergents_Paper','Delicatessen']
plt.pie(dfd2,labels=varieties)
plt.title('Region : Lisbon')
```



In Lisbon Region, Fresh variety seems to be highest spending. In the Lisbon region it is generating Highest revenue whereas Delicatessen has lowest spendings

Now lets plot All varieties Spendings in Oporto, we can see below pie chart for it

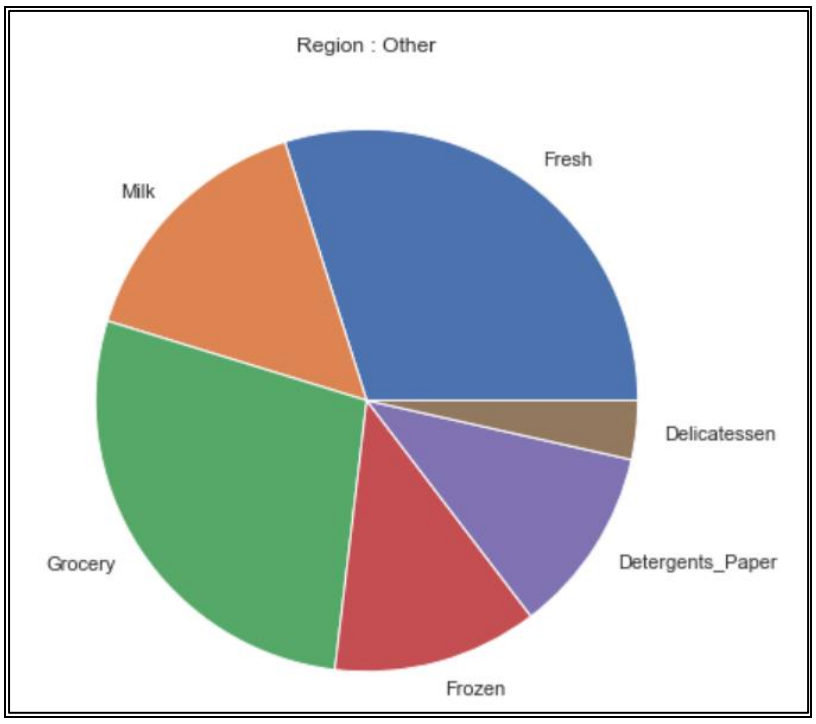
```
dfd3=dfd1.iloc[1,:]
dfd3

fig = plt.figure(figsize =(10, 7))
varieties=['Fresh','Milk','Grocery','Frozen','Detergents_Paper','Delicatessen']
plt.pie(dfd3,labels=varieties)
plt.title('Region : Oporto')
```




In Oporto Region, Fresh variety seems to be highest spending. In the Oporto region it is generating Highest revenue whereas Delicatessen has lowest spendings in this region

Now lets plot All varieties Spendings in other region,

```
dfd4=dfd1.iloc[1,:]
dfd4

fig = plt.figure(figsize =(10, 7))
varieties=['Fresh','Milk','Grocery','Frozen','Detergents_Paper','Delicatessen']
plt.pie(dfd4,labels=varieties)
plt.title('Region : Other')
```




In Other Region, Fresh variety seems to be highest spending. In Other region it is generating Highest revenue whereas Delicatessen has lowest spendings in this region

Overall, in all Region Fresh and Grocery are sources of Maximum income, so transport for this Varieties should be quickest,with high production and needed more focus, as compare to others