```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

Preliminary Analysis of dataset: Product Launch Dataset

```
print("creating the dataframe from Product_Launch_Dataset.csv file..")
df_product_launch = pd.read_csv('Product_Launch_Dataset.csv')
print('Done.\n')
```

```
creating the dataframe from Product_Launch_Dataset.csv file..
Done.
```

```
df_product_launch.head(5)
```

```
   Product id                                        Flavor  \
0           1  Herbs, not specified; Fruit, not specified
1           2                        Fruit, not specified
2           3                        Lemon; Honey; Ginger
3           4                        Mango; Passion Fruit
4           5                                       Mango

     Market Subcategory Launch Date Country       Region  \
0     Other Soft Drinks  01-01-2001     WE1  West Europe
1            Carbonates  01-01-2001     WE2  West Europe
2  Juice & Juice Drinks  01-01-2001     WE2  West Europe
3  Juice & Juice Drinks  01-01-2001     WE2  West Europe
4  Juice & Juice Drinks  01-01-2001     WE2  West Europe

                                    Positioning
0                            Low Calorie, Natural
1                         Convenience - Consumption
2  100% Not from Concentrate, Convenience - Consu...
3  100% Not from Concentrate, Convenience - Consu...
4  100% Not from Concentrate, Convenience - Consu...
```

```
df_product_launch.shape
```

```
(114689, 7)
```

Basic Stats:

```
print("Total data ")
print("-"*50)
print("\nTotal No of
products :",len(np.unique(df_product_launch["Product id"])))
print("Total No of Market Subcategories:",
len(np.unique(df_product_launch["Market Subcategory"])))
```

```
print("Total No of Countries  :",
len(np.unique(df_product_launch["Country"])))
print("Total No of Regions  :",
len(np.unique(df_product_launch["Region"])))
```

```
Total data
-------------------------------------------------

Total No of products : 114689
Total No of Market Subcategories: 14
Total No of Countries  : 108
Total No of Regions  : 8
```

Data Quality Check:

- Check for duplicates: if product ids are same for two products
- Check and validate NaN values

```
dup_bool = df_product_launch.duplicated(["Product id"])
dups = sum(dup_bool) # by considering Product ids...
print("There are {} duplicate rating entries in the
data..".format(dups))
```

```
There are 0 duplicate rating entries in the data..
```

```
print("No of columns found with NaNs : ",
sum(df_product_launch.isnull().any()))
```

```
No of columns found with NaNs :  2
```

```
df_product_launch.isnull().any()
```

```
Product id            False
Flavor                 True
Market Subcategory    False
Launch Date           False
Country               False
Region                False
Positioning            True
dtype: bool
```

- Found Flavor & Positioning columns with NaNs
- Deep dive into data to get better understanding on NaNs

```
df_product_launch [df_product_launch.isnull().Flavor == True]
```

```
        Product id Flavor         Market Subcategory Launch Date
Country  \
11              12    NaN       Juice & Juice Drinks  01-01-2001
WE3
12              13    NaN       Juice & Juice Drinks  01-01-2001
WE3
32              33    NaN                  Iced Tea  01-01-2001
A1
```

```
129            130    NaN          Sports Drinks RTD  01-01-2001
NA1
130            131    NaN              Energy Drinks  01-01-2001
NA1
...            ...    ...                        ...         ...     .
..
78006        78007    NaN                  Iced Tea  01-01-2007
WE10
78358        78359    NaN                Carbonates  01-01-2010
WE4
94438        94439    NaN                Carbonates  01-01-2007
AF14
100214      100215    NaN  Bottled Water - Flavoured  01-01-2008
NA1
110293      110294    NaN                Carbonates  01-01-2010
A13

               Region
Positioning
11         West Europe  100% Not from Concentrate, No Added Sugar,
Vit...
12         West Europe  Nectars (25-99% juice), Energy/Alertness,
Vita...
32                Asia  Convenience - Consumption, No
Additives/Preser...
129      North America       Sports & Recovery, Convenience -
Consumption
130      North America        Convenience - Consumption,
Energy/Alertness
...                ...                                          .
..
78006      West Europe                          Convenience -
Consumption
78358      West Europe                               Low Calorie,
Natural
94438          Africa  Low Calorie, Sugar Free, Convenience -
Consump...
100214  North America  Allergy, Antioxidant, Gluten Free, Low
Calorie...
110293           Asia  Traditional, Sugar Free, Convenience -
Consump...

[1976 rows x 7 columns]
```

```
df_product_launch [df_product_launch.isnull().Positioning == True]
```

```
        Product id                                 Flavor  \
36              37             Grapefruit, not specified
88              89  Grape, red; Raspberry, not specified
111            112                          Elderflower
```

```
119            120                          Unflavored
160            161                   Berry, Cranberry
...            ...                               ...
114438      114439                Citrus, Not Specified
114439      114440                Citrus, Not Specified
114505      114506                 Berry, Not Specified
114512      114513                 Berry, Not Specified
114670      114671                                 Cola

                Market Subcategory Launch Date Country         Region
\
36        Bottled Water - Flavoured  01-01-2001     EE3    East Europe

88                       Carbonates  01-01-2001     LA2  Latin America

111            Juice & Juice Drinks  01-01-2001     WE8    West Europe

119     Bottled Water - Unflavoured  01-01-2001     EE3    East Europe

160                      Carbonates  01-01-2001     NA1  North America

...                             ...         ...     ...            ...

114438                   Carbonates  01-01-2010     NA1  North America

114439                   Carbonates  01-01-2010     NA1  North America

114505                Energy Drinks  01-01-2010     NA1  North America

114512                Energy Drinks  01-01-2010     NA1  North America

114670                   Carbonates  01-01-2010    AF14         Africa


       Positioning
36             NaN
88             NaN
111            NaN
119            NaN
160            NaN
...            ...
114438         NaN
114439         NaN
114505         NaN
114512         NaN
114670         NaN

[4312 rows x 7 columns]
```

Observation:

- 1976/114689 (0.02%) products have Flavor data missing
- 4312/114689 (0.04%) products have Positioning data missing

Pre-processing:

Exploding Flavor Column to multiple rows:

- Split multiple flavors of a product from Flavor column (semi-colon separated) to Product id x Flavor pair . i.e., one row must contain only 1 product and one of its flavors only

```
df_product_launch.head(5)
```

```
   Product id                                      Flavor  \
0           1  Herbs, not specified; Fruit, not specified
1           2                        Fruit, not specified
2           3                        Lemon; Honey; Ginger
3           4                        Mango; Passion Fruit
4           5                                       Mango

     Market Subcategory Launch Date Country       Region  \
0     Other Soft Drinks  01-01-2001     WE1  West Europe
1            Carbonates  01-01-2001     WE2  West Europe
2  Juice & Juice Drinks  01-01-2001     WE2  West Europe
3  Juice & Juice Drinks  01-01-2001     WE2  West Europe
4  Juice & Juice Drinks  01-01-2001     WE2  West Europe

                                    Positioning
0                           Low Calorie, Natural
1                        Convenience - Consumption
2  100% Not from Concentrate, Convenience - Consu...
3  100% Not from Concentrate, Convenience - Consu...
4  100% Not from Concentrate, Convenience - Consu...
```

```
df_product_launch["Flavor (modified)"] = df_product_launch["Flavor"]

df_product_launch["Flavor (modified)"] = df_product_launch["Flavor
(modified)"].apply( lambda x: (str(x).replace(',','-')))

df_product_launch["Flavor (modified)"] = df_product_launch["Flavor
(modified)"].apply( lambda x: (str(x).split(';')))

df_product_launch.head(5)
```

```
   Product id                                      Flavor  \
0           1  Herbs, not specified; Fruit, not specified
1           2                        Fruit, not specified
2           3                        Lemon; Honey; Ginger
3           4                        Mango; Passion Fruit
4           5                                       Mango
```

```
     Market Subcategory Launch Date Country       Region  \
0     Other Soft Drinks  01-01-2001     WE1  West Europe
1            Carbonates  01-01-2001     WE2  West Europe
2  Juice & Juice Drinks  01-01-2001     WE2  West Europe
3  Juice & Juice Drinks  01-01-2001     WE2  West Europe
4  Juice & Juice Drinks  01-01-2001     WE2  West Europe

                                      Positioning  \
0                            Low Calorie, Natural
1                          Convenience - Consumption
2  100% Not from Concentrate, Convenience - Consu...
3  100% Not from Concentrate, Convenience - Consu...
4  100% Not from Concentrate, Convenience - Consu...

                              Flavor (modified)
0  [Herbs- not specified,  Fruit- not specified]
1                          [Fruit- not specified]
2                       [Lemon,  Honey,  Ginger]
3                        [Mango,  Passion Fruit]
4                                        [Mango]
```

```
df_product_launch = df_product_launch.explode('Flavor (modified)')
```

```
df_product_launch.head(5)
```

```
   Product id                                      Flavor  \
0           1  Herbs, not specified; Fruit, not specified
0           1  Herbs, not specified; Fruit, not specified
1           2                        Fruit, not specified
2           3                        Lemon; Honey; Ginger
2           3                        Lemon; Honey; Ginger

     Market Subcategory Launch Date Country       Region  \
0     Other Soft Drinks  01-01-2001     WE1  West Europe
0     Other Soft Drinks  01-01-2001     WE1  West Europe
1            Carbonates  01-01-2001     WE2  West Europe
2  Juice & Juice Drinks  01-01-2001     WE2  West Europe
2  Juice & Juice Drinks  01-01-2001     WE2  West Europe

                                      Positioning      Flavor
(modified)
0                            Low Calorie, Natural   Herbs- not
specified
0                            Low Calorie, Natural   Fruit- not
specified
1                          Convenience - Consumption   Fruit- not
specified
2  100% Not from Concentrate, Convenience - Consu...
Lemon
```

```
2  100% Not from Concentrate, Convenience - Consu...
Honey
```

```
df_product_launch[['Flavor (Calculated)','Flavor type (Calculated)']]
= df_product_launch['Flavor (modified)'].str.split('-',n=1,
expand=True)
```

```
df_product_launch.head(5)
```

```
   Product id                                      Flavor  \
0           1  Herbs, not specified; Fruit, not specified
0           1  Herbs, not specified; Fruit, not specified
1           2                        Fruit, not specified
2           3                        Lemon; Honey; Ginger
2           3                        Lemon; Honey; Ginger

     Market Subcategory Launch Date Country       Region  \
0     Other Soft Drinks  01-01-2001     WE1  West Europe
0     Other Soft Drinks  01-01-2001     WE1  West Europe
1            Carbonates  01-01-2001     WE2  West Europe
2  Juice & Juice Drinks  01-01-2001     WE2  West Europe
2  Juice & Juice Drinks  01-01-2001     WE2  West Europe

                                       Positioning      Flavor
(modified)  \
0                            Low Calorie, Natural   Herbs- not
specified
0                            Low Calorie, Natural   Fruit- not
specified
1                       Convenience - Consumption   Fruit- not
specified
2  100% Not from Concentrate, Convenience - Consu...
Lemon
2  100% Not from Concentrate, Convenience - Consu...
Honey

  Flavor (Calculated) Flavor type (Calculated)
0               Herbs            not specified
0               Fruit            not specified
1               Fruit            not specified
2               Lemon                     None
2               Honey                     None
```

```
df_product_launch = df_product_launch.drop(columns = 'Flavor
(modified)')
```

Exploding Positioning Column to multiple rows:

- Split multiple positioning sub-categories of a product from Positioning column (comma separated) to Product id x positioning sub-category pair . i.e., one row must

contain only 1 positioning subcategory so we can assign respective positioning group by joining tables later in this activity.

```
df_product_launch["Positioning Subcategory (Calculated)"] =
df_product_launch["Positioning"]

df_product_launch["Positioning Subcategory (Calculated)"] =
df_product_launch["Positioning Subcategory (Calculated)"].apply(
lambda x: (str(x).split(',')))

df_product_launch.head(5)
```

```
   Product id                                      Flavor  \
0           1  Herbs, not specified; Fruit, not specified
0           1  Herbs, not specified; Fruit, not specified
1           2                       Fruit, not specified
2           3                       Lemon; Honey; Ginger
2           3                       Lemon; Honey; Ginger

     Market Subcategory Launch Date Country       Region  \
0     Other Soft Drinks  01-01-2001     WE1  West Europe
0     Other Soft Drinks  01-01-2001     WE1  West Europe
1            Carbonates  01-01-2001     WE2  West Europe
2  Juice & Juice Drinks  01-01-2001     WE2  West Europe
2  Juice & Juice Drinks  01-01-2001     WE2  West Europe

                                        Positioning Flavor
(Calculated)  \
0                            Low Calorie, Natural
Herbs
0                            Low Calorie, Natural
Fruit
1                       Convenience - Consumption
Fruit
2  100% Not from Concentrate, Convenience - Consu...
Lemon
2  100% Not from Concentrate, Convenience - Consu...
Honey

  Flavor type (Calculated)              Positioning Subcategory
(Calculated)
0            not specified                              [Low Calorie,
Natural]
0            not specified                              [Low Calorie,
Natural]
1            not specified                         [Convenience -
Consumption]
2                     None  [100% Not from Concentrate,  Convenience -
Con...
2                     None  [100% Not from Concentrate,  Convenience -
Con...
```

```
df_product_launch = df_product_launch.explode('Positioning Subcategory
(Calculated)')

df_product_launch.head(5)
```

```
   Product id                                        Flavor Market
Subcategory  \
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
1           2                        Fruit, not specified
Carbonates

  Launch Date Country       Region                Positioning  \
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
1  01-01-2001     WE2  West Europe  Convenience - Consumption

  Flavor (Calculated) Flavor type (Calculated)  \
0               Herbs            not specified
0               Herbs            not specified
0               Fruit            not specified
0               Fruit            not specified
1               Fruit            not specified

  Positioning Subcategory (Calculated)
0                           Low Calorie
0                               Natural
0                           Low Calorie
0                               Natural
1            Convenience - Consumption
```

Checking for Data Inconsitency...

```
print(df_product_launch["Flavor (Calculated)"].unique())
```

```
['Herbs' ' Fruit' 'Fruit' 'Lemon' ' Honey' ' Ginger' 'Mango'
 ' Passion Fruit' 'Apple' ' Pineapple' ' Guava' 'Cherry' ' Vanilla'
 'Orange' 'Superfruit' ' Berry' 'nan' ' Apricot' ' Herbs' 'Passion
Fruit'
 'Guarana' 'Coffee' ' Grape' ' Grapefruit' 'Peach' 'Cola' ' Cherry'
'Rose'
 ' Melon' ' Lemon' 'Berry' ' Tea' 'Vegetables' 'Tea' 'Strawberry'
'Grape'
 'Grapefruit' 'Raspberry' 'Cranberry' 'Mandarin' ' Mint' 'Wildberry'
```

```
 'Fennel' ' Coriander' ' Superfruit' ' Cinnamon' ' Pear' ' Plum'
 'Unflavored' 'Citrus' 'Tropical Fruit' ' Raspberry' ' Japanese' '
Rice'
 'Prune' ' Orange' 'Pineapple' ' Mango' ' Carrot' 'Guava' 'Lime' '
Citrus'
 ' Caramel' 'Aloe Vera' 'Spices' ' Cola' 'Vanilla' 'Cocoa'
'Elderflower'
 'Lemonade' ' Strawberry' 'Blackcurrant' ' Banana' ' Apple' ' Lime'
 ' Prickly Pear' 'Chocolate' 'Ginger Beer' ' Fig' ' Peach' '
Blackcurrant'
 ' Boysenberry' ' Kiwi' 'Honey' ' Ginseng' 'Exotic Fruit' 'Coconut'
 'Carrot' ' Soy' 'Beer' 'Cinnamon' ' Coffee' ' Tangerine' 'Ginseng'
 ' Lemonade' 'Banana' ' Pumpkin' 'Nougat' ' Chocolate' 'Tangerine'
 ' Vinegar' 'Soy' ' Fish' ' Vegetables' ' Mandarin' ' Beetroot'
 ' Blueberry' ' Balm' ' Broccoli' ' Licorice' ' Guarana' ' Aloe Vera'
 ' Echinacea' ' Cranberry' 'Forest Fruit' 'Apricot' 'Kiwi'
'Peppermint'
 ' Coconut' 'Vinegar' 'Bilberry' 'Papaya' 'Mint' 'Tomato' ' Chili'
 ' Pepper' ' Watermelon' ' Cloves' ' Tomato' 'Ginger' 'Pear'
'Licorice'
 'Tamarind' ' Thyme' ' Nata De Coco' ' Elderflower' 'Elderberry' '
Milk'
 'Gingko' ' Cream' ' Rhubarb' 'Chocolate Fudge' 'Root Beer' 'Cream
Soda'
 ' Cider' ' Cherry Blossom (Sakura)' ' Rose' 'Fig' 'Onion' 'Melon'
 'Honeydew' 'Water Chestnut' 'Sweet & Sour' ' Lemongrass' 'Plum'
 'Watermelon' ' Sugar' 'Grains' ' Basil' ' Grains' 'Date' 'Cappuccino'
 ' Mexican' ' Artichoke' ' Nut' 'Ginger Ale' 'Nectarine' ' Tropical
Fruit'
 'Bubble Gum' ' Honeydew' 'Cocktail' 'Liqueur' 'Cider' 'Nut' ' Malt'
 ' Thai' ' Tiramisu' 'Lotus' ' Oriental' ' Redcurrant' ' Jasmine'
 ' Chinese' 'Chinese' ' Date' 'Wine' ' Radish' 'Malt' ' Olive'
 'Cherry Blossom (Sakura)' ' Exotic Fruit' ' Cactus' 'Bean'
 ' Orange Blossom' 'Raisin' ' Bilberry' 'Noni' ' Fenugreek' '
Lavender'
 ' Spearmint' ' Chrysanthemum' 'Spearmint' 'Barley' 'Rosehip'
'Watercress'
 'Chrysanthemum' ' Beer' 'Hibiscus' ' Toffee' ' Indian' 'Jasmine'
 'Lavender' ' Hibiscus' 'Galangal' ' Cardamon' 'Rhubarb' ' Peppermint'
 'Clementine' 'Thyme' ' Passion Flower' ' Bamboo Shoot' ' Juniper'
 'Caramel' ' Elderberry' 'Mushroom' ' Pimento' ' Honeysuckle' 'Sugar'
 'Orange Blossom' 'Lemongrass' 'Caramel ' ' Papaya' ' Cooling
Sensation'
 ' Sweet & Sour' ' Melissa' 'Cumin' ' Salt' 'Pepper' ' Turnip Greens'
 'Japanese' 'Toffee' 'Sesame' 'Cucumber' 'Corn' 'Vodka' 'Lemon '
 'Tutti Frutti' 'Turmeric' ' Tamarind' ' Champagne' ' Nectarine' '
Corn'
 'Birch Tree' 'Rice' 'Bergamot' 'Sage' 'Persimmon' 'Echinacea' '
Woodruff'
 'Milk' 'Blueberry' ' Asian' ' Forest Fruit' ' Wine' ' Sesame' '
```

Meringue'
 ' Feijoa' 'Longan' 'Chili' 'Saffron' ' Pandan' ' Herb' 'Mexican'
'Olive'
 'Seeds' ' Seeds' ' Burdock' ' Dandelion' ' Gingko' ' Barley' 'Basil'
 'Yogurt' 'Gingerbread' ' Saffron' ' Root Beer' ' Custard'
'Boysenberry'
 ' Cassis' ' Spices' ' Moroccan' ' Yogurt' ' Cucumber' 'Beetroot'
 ' Spinach' ' Sage' ' Geranium' ' Asparagus' ' banana' 'Kale'
'Melissa'
 ' Beetroot Greens' ' Worcestershire' 'Indian' ' Egg' 'Wheat' '
Celery'
 ' Anise' 'Cassis' 'Rum' ' Bergamot' 'Bluecurrant' ' Guarana '
'Garlic'
 'Dandelion' ' Spirit' ' Clementine' 'Feijoa' 'Turnip' ' Cabbage'
'Asian'
 'Oriental' ' Wildberry' 'Pie' 'Oregano' 'Cookies & Cream' 'Asparagus'
 'Jaeggermeister' 'Pumpkin' 'Nettle' 'Candy Corn' ' Bluecurrant'
 ' Physalis' 'Cream' ' Rosemary' 'Syrup' ' Syrup' 'Prickly Pear' 'Ube'
 ' Nutmeg' 'Tamarillo' 'Brussels Sprout' ' Turkey' 'Tiramisu'
'Lettuce' ''
 'Redcurrant' 'Champagne' ' Tumeric' 'Thai' ' Noni' 'Pandan' ' Yam'
 ' Oregano' ' Maple' ' Lemon ' ' Cheese' 'Marshmallow' 'Buttermilk'
 'Saffrom Milk' ' Liqueur' ' Raisin' ' Beans' ' Shellfish' 'Egg Nog'
 ' Wintergreen' ' Lotus' 'Wintergreen' ' Eucalyptus' ' Menthol'
 ' Cream Soda' 'Kombacha' ' Mushroom' 'Potato' 'Cake' 'Tayberry'
'Salt'
 'Sorrel' ' Spanish' ' Oil' ' Cocoa' ' Raspberry' ' Paprika'
'Woodruff'
 'Carob' ' Rosehip' 'Yam' ' Prune' ' Persimmon' 'Moroccan' ' Onion'
 ' Cucumber' ' Eggplant' ' Cumin' ' Marshmallow' ' Kombacha'
 ' Unflavored' 'Menthol' 'Cabbage' ' Cocktail' ' Curry' ' Arrowroot'
 ' Lotus Root' 'Barbecue' ' Potato' 'Nuts' ' Squash' ' Sea' ' Italian'
 'Akee' ' Caraway' ' Kiwi' 'Pitanga' ' Seaweed' ' Marjoram' ' Nuts'
 'Peanut Butter' 'Curacao' 'Fish' ' Kiwano' 'Cupuacu' ' Ginger Ale'
 'Shaddock' ' Nettle' 'Cardamon' 'Loquat' ' Tutti Frutti'
 'Giant Granadilla' 'Hogplum' ' Bloody Mary' ' Pie' ' Pitanga'
 ' Dulce De Leche' 'Rosewater' ' Rum' ' Valerian' 'Rose Petal'
 'Whortleberry' ' Giant Granadilla' ' Roasted' 'Triple Sec' '
Caribbean'
 'Spirit' ' Marrow' ' Turnip' ' Galanga' 'Masala' ' Wheat' ' Greek'
 ' Turmeric' ' Korean' 'Spinach' 'Courgette/Squash' 'Anise' ' Longan'
 'Martini' ' Zucchini' 'Peach Melba' ' Peach Palm' 'Water' 'Beans'
 'Cranberry ' 'Mandarin ' 'Peach ' 'indian' 'Beetroot Greens' ' Vodka'
 'Medlar' ' Grapefruit' ' Martini' 'Maple' 'Lavender ' 'Greengage'
 'Broccoli' ' Kale' 'Oil' 'Mediterranean' 'Greek' ' Rose '
 'Collard Greens' ' Butterscotch' ' Birch Tree' 'Jamaica Cherry' '
Fennel'
 ' Loquat' ' Praline' ' Parsnip' 'Curcumin' ' Acerola' 'Nata De Coco'
 'Sake' ' Sake' 'Garden Greens' ' Bean' ' Garlic' ' Butter' ' Dill'
 ' Avocado' 'Violet' ' Cookies & Cream' 'Celeriac' 'Canteloupe'

 'appletest' ' Sorrel' ' Naartjie' ' Tangelo' ' Curcumin' 'Osmanthus'
 ' Yeast' 'Rambutan' 'Physalis' 'Dulce De Leche' 'Cookie Dough'
 'Huckleberry' 'Praline' 'Naartjie' ' Cupuacu' 'Sweetsop' '
Elderberry'
 ' Chokeberry' ' Acacia' 'Balm' 'Yeast' 'Vermouth' 'Graham Cracker'
 ' Cappuccino' 'Arrowroot' ' Curacao' ' Taro' 'Tequila' ' Chocolate
Chip'
 ' Rose Petal' 'Roasted' 'Tarragon' 'Cassia' ' Cake' 'grape' 'Satsuma'
 ' Ginger Beer' 'Dana' ' Berry' ' Grains' ' Oreo' 'Brambleberry'
 'Cotton Candy' 'Celery' ' French' ' Parsley' ' Orange' 'Oreo'
 'Chocolate Chip Cookie' 'Creme Brulee' 'Pitahaya' ' Violet' ' Oats'
 ' Strawberry' 'Cajun' 'Bael Fruit' 'Parsley' 'Pea' 'Caribbean'
 ' Galangal' ' Yuca' 'Laurel' ' Brandy' 'Sala' ' Molasses' '
Canteloupe'
 ' Currant' 'Taro' 'Tumeric' ' Bay Leaf' ' Milk' 'Bacon' 'Italian'
 ' Gingerbread' 'Jaboticaba' 'Honeysuckle' ' Seafood' 'Chokeberry'
 ' Mint' 'Shellfish' 'Buffalo Wing' ' Chicken' 'Pork' ' Spare Ribs'
 'Beef' ' Hickory' 'Chicken' ' Teriyaki' 'Sausage' ' Sausage' 'Duck'
 ' Garam Masala' 'Veal' 'Egg' ' Bacon' 'Ham' 'Lamb' ' Pork' 'Meat'
 ' Bubble Gum' 'Gourd' 'Indian Jujube' ' Masala' 'Bamboo Shoot'
 'Butterscotch' 'Agave' 'Black Sapote' 'Butter' ' Fruit' 'French'
 ' chokeberry' 'Aniseed' ' Cayenne' 'Acacia' ' Triple Sec' ' Tabasco'
 'Litchi' ' exotic fruit' ' Savory' 'Jonagold' 'Eggplant' ' Ube'
'Seaweed'
 ' Barbados Cherry' 'Kefir' ' Shortbread' ' Wasabi' 'Currant'
'Zucchini'
 ' Moringa' 'Gin' ' Grape' 'Cowberry' ' Tamarillo' ' red fruits'
'Squash'
 'Fudge' ' Kefir' ' Pitahaya' 'Moringa' ' Wax Jambu' 'Artichoke'
 'Rosemary' ' Cauliflower' 'Butter Pecan' ' Matrimony Vine' 'Hickory'
 ' Smoked' 'Avocado' ' Linden Blossom' ' Gourd' 'Shortbread'
 'Chocolate Chip' ' Cookie Dough' ' Alfalfa' ' Rose Apple' ' Tandoori'
 ' Caramel Salted' ' Pomegranate' 'Pink Grapefruit' ' Sloeberry'
 ' Tarragon' 'Nutmeg' 'Salak' ' Superfruit' 'Custard' ' Egg Nog'
 ' Cherry Blossom' 'Barbados Cherry' 'Caffeine' ' Plantain' 'Burdock'
 'Cherry Blossom' 'Lotus Root' 'Caramel Salted' 'Apple Cobbler' '
Honey'
 ' Lettuce' ' Brussels Sprout' ' Allspice' ' Mango' ' Banana' '
Medlar'
 'Cheese' ' Star Anise' ' Jamaica Cherry' ' Water Chestnut' 'Waxberry'
 ' Bubble Gum' ' Mocha' ' Huckleberry' ' Osmanthus' 'Galanga' 'Oats'
 'Honey Fungus' ' Carrot' ' Pea' ' Whiskey' ' aroma' ' Aroma' '
Sprouts'
 'Hawthorn' 'Whiskey' ' Peanut Butter' ' Grand Marnier' 'Bayberry'
 'Jamaica' 'Geranium' 'Cloves' 'Bignay' 'Mocha' ' Mabolo'
 ' Vegetable Marrow' 'Cape Gooseberry' ' Buttermilk' ' Aniseed' '
Peas'
 'Nashi Pear' 'Sweet Grenadilla' ' Molokhia' "S'Mores" ' Water'
 'Cherry Apple' 'Chocolate Hazelnut' ' Celeriac' 'Cayenne' 'Molokhia'
 'Morel' ' Gin' 'Tomatillo' ' Collard Greens' ' Cotton Candy'

```
 'Tree Tomato' ' Guyabano' ' Watercress' 'Mesquite' 'Sea' ' Swiss
Chard'
 ' Ham' ' water' 'Valerian' 'Peas' ' Sriracha' 'Pickle' 'Soda']
```

```
print(df_product_launch['Flavor type (Calculated)'].unique())
```

```
[' not specified' None ' Red' ' Not specified' ' Pomegranate'
 ' Blackberry' ' red' ' acai' ' Blueberry' ' Not Specified' ' blood'
 ' green' ' Lychee' ' cappuccino' ' mixed' ' pomegranate' ' Cranberry'
 'Lime' ' blue' ' pink' ' Berry- Blackberry' ' Green' ' White' '
mocha'
 ' latte macchiatto' ' Cappuccino' ' black' ' Aronia (Chokeberry)'
 ' Chinotto' ' white' ' lemon' ' gooseberry' ' espresso' ' french'
 ' latte' ' hawthorn' ' jujube' ' Milk' ' Bitter' ' Black' ' Blood'
 ' herbal' ' Wildberry' ' wild' ' oolong' ' Acerola' ' fermented'
 ' Cactus' ' Mulberry' ' Dragonfruit' ' rooibos' ' earl grey' '
mulberry'
 ' macadamia' ' Pina Colada' ' Irish Cream' ' almond' ' milk' '
amaretto'
 ' Mixed' ' arabica' ' Amaretto' ' rose' ' Mangosteen' ' Quince'
 ' Boysenberry' ' Agave' ' mung' ' walnut' ' peanut' ' chamomile'
 ' Pomelo' ' Oolong' ' mangosteen' ' Soursop & Guanabana' ' rosehip'
 ' Rooibos' ' Acai' ' Meringue' ' cashew' ' Pink' ' French' '
hazelnut'
 ' Java' ' Mocha' ' Wild' ' Brown' ' starfruit' ' Robusta' '
Dragonfruit '
 ' apple cider' ' dark' ' Espresso' ' mint' ' azuki' ' matcha' '
chestnut'
 ' yellow' ' Cashew' ' Elderberry' ' Yellow' ' calamansi' ' jasmine'
 ' Cafe au lait' ' Soy' ' bell red' ' brown' ' Habanero' ' Bloody
Mary'
 ' key' ' carambola' ' rosemary' ' jackfruit' ' pomelo' ' bitter'
 ' dark roast' ' hemp' ' Chicory' ' not Specified' ' condensed' '
Latte'
 ' seabuckthorn' ' Jujube' ' Latte macchiatto' ' merlot' ' buckwheat'
 ' pawpaw' ' Sicilian' ' Malt' ' Carambola' ' soy' ' Yuzu' ' Herbal'
 ' mate' ' Durum' ' Goji' ' goji' ' Oats' ' Seabuckthorn' ' Assam'
 ' Martini' ' Bell red' ' loganberry' ' apple' ' Condensed' '
Margarita'
 ' cane' ' red wine' ' Woodberry' ' malt' ' Mate' ' Hazelnut'
 ' granny smith' ' Loganberry' ' Lager' ' poppy' ' Dutch' ' Muscat'
 ' kumquat' 'rooibos ' ' lingon' ' Colombia' ' ale' ' swiss' ' sour'
'Cap'
 ' bourbon' ' Mudslide' ' Clam' ' jalapeno' ' Gooseberry' ' Golden'
 ' sweet' ' osmanthus' ' blackforest' ' cherimoya' ' Cowberry' '
coconut'
 ' syrup' ' Dark' ' Sweet' ' Darjeeling' ' Arabica' ' yumberry'
 ' Calamansi' ' pistachio' ' Guatemala' ' Hawthorn' ' colombia'
 ' darjeeling' ' butternut' ' Sour' ' Salt' ' Mulled' ' café au lait'
 'red ' ' marula' ' rye' ' Cosmopolitan' ' mungo' ' cider' ' Mojito'
 ' sun-dried' ' Marula' ' Kabuse' ' Macadamia' ' Lingon' ' plum' '
```

baobab'
 ' Mint' ' Tequila' ' chai' ' multi' ' Dragonfruit- Mango- Pineapple'
 ' salt' ' Valencia' ' Earl grey' ' Key lime' ' Apple' ' mocha java'
 ' Hogplum' ' Latte ' ' Yumberry' ' Daiquiri' ' Vodka' ' Olive' '
assam'
 ' Lemon' ' Matcha' ' whipped' ' chardonnay' ' Marocchino' ' Rosehip'
 ' Key' ' Hemp' ' Mai Tai' ' Fennel' ' Flaxseed' ' caja' ' Chai' '
Almond'
 ' Shrimp' ' Osmanthus' ' Apple cider' ' Baobab' ' Kombucha' '
boletus'
 ' Cocoa' ' Jasmine' ' peri peri' ' Cloudberry' ' madagascan'
 ' Caipirinha' ' valencia' ' Blue' ' pine' ' Kahlua' ' pecan' '
Marion'
 ' Syrup' ' Sweetener' ' robusta' ' Caja' ' Verbena'
 ' Custard Apple (Noi Naa)' ' Jackfruit' ' Cane' ' Sangria' ' camu
camu'
 ' verbena' ' Barberry' ' soar' ' dutch' ' Acacia' ' Rye' ' fructose'
 ' Walnut' ' kombucha' ' Dark roast' ' Coconut' ' cheesecake' '
Chamomile'
 ' Irish Coffee' ' Mirabelle' ' Sambuca' ' guatemala' ' Starfruit'
 ' Buckwheat' ' muscat' ' Irish' ' senna' ' roasted'
 ' carambola/starfruit' ' Oatmeal' ' belgian' ' morello' ' Madagascan'
 ' mascarpone' ' apple\xa0' ' Plum' ' Mint Julep' ' Wholegrain' '
Durian'
 ' Peanut' ' Camu camu' ' Granny smith' ' irish' ' habanero' '
maraschino'
 ' Pawpaw' ' Aronia (Chokeberry(' ' Sauvignon blanc' ' Rose' '
Curacao'
 ' cabernet sauvignon' ' Pear' ' Merlot' ' Rice' ' Kumquat' ' golden'
 ' hamburger' ' cheddar' ' shiitake' ' prosciutto' ' champignon'
 ' gorgonzola' ' brie' ' oyster' ' clam' ' Moringa' ' sicilian'
 ' Chardonnay' ' Mung' ' Belgian' ' Purple' ' Multi' ' Goldenberry'
 ' Ginger' ' Sunflower' ' Long grain' ' Poppy' ' Whipped' '
Wheatgrass'
 ' Pistachio' ' Natural' ' Cherry' ' Swiss' ' Stevia' ' Sponge'
 ' Chestnut' ' Marzen' ' Moringa' ' amaranth' ' ruby' ' Romaine'
 ' Jalapeno' ' Ruby' ' Chia' ' Doum Palm' ' caramelized' ' Cider'
 ' Bourbon' ' Linseed' ' java' ' Yacon' ' Camu Camu' ' marsala' '
truffle'
 ' Pecan' ' Truffle' ' Ethiopia' ' Morello' ' Roasted' ' Fructose'
 ' Senna' ' Blonde' 'Apple-red' ' Quinoa' ' Apple\xa0' ' Buckthorn' ''
 ' Cream' ' Wineberry' ' Azuki' ' celery' ' Granola' ' chicory'
'Grape'
 ' Crowberry' ' Red velvet' ' bean' ' Lucuma' ' Pinot' ' Millet'
 ' Butternut' ' Maqui (Chilean Wineberry)' ' Amaranth' ' Bran' '
Pumpkin'
 ' Fried' ' Ale' ' Pale ale' ' Pinot grigio' ' Tahitian' ' Beach'
 ' Mocha java' ' Bean' ' Shiraz' ' Pine' ' Cherimoya'
 ' Cabernet sauvignon' ' Pin-head' ' Blackforest' ' Chinese']

```
print(df_product_launch['Positioning Subcategory
(Calculated)'].unique())
```

```
['Low Calorie' ' Natural' 'Convenience - Consumption'
 '100% Not from Concentrate' ' Convenience - Consumption'
 ' No Added Sugar' ' No Additives/Preservatives' 'Nectars (25-99%
juice)'
 ' Low Calorie' ' Vitamin/Mineral Fortified' ' Digestive/Gut Health'
 ' High/Source of Fibre' 'Dry' ' Low Fat' ' Low Sodium' ' Low Carb'
 ' Antioxidant' ' Immune Health' 'Convenience - Packaging' ' Low
Sugar'
 'Female' ' Sugar Free' ' Anti-Aging/Aging-Well' ' Skin Health'
 ' Energy/Alertness' ' Convenience - Easy-to-Prepare' 'Heart Health'
 'Juice Drinks (up to 25% juice)' 'Vegetarian'
 ' Convenience - Time Saving' 'Low Fat' 'Antioxidant' 'Economy'
 '100% Reconstituted' ' Ethical - Environment' 'Ethical - Packaging'
 'Sports & Recovery' 'Novel and Fun' ' Dry' 'nan' ' Children (5-12
years)'
 ' Convenience - Ready Prepared' 'Sugar Free' ' Weight Management'
 ' Added Protein' ' High/Source of Protein' 'Indulgent and Premium'
 ' Nectars (25-99% juice)' ' Ethical - Packaging' 'Ethnic and Exotic'
 'No Added Sugar' ' Added Calcium' ' Traditional' ' Gluten Free'
 ' Allergy' ' Eye Health' ' Bone Health' ' Lactose Free' ' Added
Fibre'
 ' Sports & Recovery' ' Sticks' 'Convenience - Easy-to-Prepare' '
Organic'
 'Natural' ' 100% Reconstituted' ' Heart Health' ' Low GI'
'Functional'
 ' not specified' ' Functional' 'Added Calcium' 'Traditional'
 ' Convenience - Packaging' 'High/Source of Protein'
'Energy/Alertness'
 ' Novel and Fun' 'Single Shot' 'Vitamin/Mineral Fortified'
 ' 100% Not from Concentrate' ' Soy Foods' ' Vegetarian' ' GMO Free'
 ' Single Shot' 'Convenience - Ready Prepared' 'Convenience - Time
Saving'
 ' Economy' 'Low Sodium' ' Brain Health' ' DHA'
 ' Juice Drinks (up to 25% juice)' 'No Additives/Preservatives'
 'Joint Health' 'Soy Foods' 'GMO Free' ' Diabetic'
 'Seasonal/In-Out Products' 'Male' 'Weight Management' 'Organic'
 ' Low Cholesterol' 'Allergy' 'Children (5-12 years)' 'Low Carb'
 'Added Fibre' ' Added Iron' 'Digestive/Gut Health' 'Low Cholesterol'
 ' Indulgent and Premium' ' Seasonal/In-Out Products' 'Microwaveable'
 'Gluten Free' 'Low Sugar' 'Co-Branding' 'Halal' ' Co-Branding'
'Kosher'
 ' Ethnic and Exotic' 'High/Source of Fibre' ' Female' ' Kosher'
 ' HFCS Free' ' Omega-3' 'Immune Health' ' Ethical - Human' ' Male'
 ' Refill' ' Prebiotic' 'Ethical - Environment' ' Wholegrain'
 ' Joint Health' ' economy' 'Bone Health' 'Novel and fun' ' Novel and
fun'
 'Eye Health' 'Ethical - Human' '100% Frozen' 'HFCS Free' 'Sticks'
```

```
 ' Oral Health' 'Skin Health' ' No Trans Fats' 'Oral Health'
'Diabetic'
 'Refill' 'Lactose Free' 'Anti-Aging/Aging-Well' ' Halal' 'Low GI'
 'Omega-3' ' Nectars (25-99% Juice)' '100% Not from concentrate'
 'Brain Health' 'Added Iron' ' Packaging' 'Seniors (55+)' ' 100%
Frozen'
 'Packaging' ' Seniors (55+)' '100% not from Concentrate'
 'Ethical - Not Specific' ' Ethical - Animal/Fish & Bird'
 ' Ethical - Not Specific' 'Digestive/Liver Health (Supplements)'
 ' Children (Supplements)' 'nectars (25-99% juice)'
 ' Convenience -  Consumption' 'Ethical - Animal/Fish & Bird'
 ' Microwaveable' ' Growing-Up Milk (1+ year)' 'Consumption'
 ' Time Saving' 'Wholegrain' 'Price Premium > or equal to 150%'
 ' Price Premium > or equal to 200%' 'Added Protein' ' Consumption'
 ' Ready Prepared' ' Easy-to-Prepare' 'No Trans Fats'
 ' Convenience -  Packaging' 'Convenience -  Consumption'
 'Energy and Stamina (Supplements)' 'DHA' 'Pre-prepared'
 ' Price Premium > or equal to 150%' 'Convenience of Usage'
 ' Pre-prepared' 'Time Saving' ' Convenience of Usage'
 ' Convenience -  Easy-to-Prepare' 'Convenience -  Easy-to-Prepare'
 'Brain-Mood Health (Supplements)' ' Energy and Stamina (Supplements)'
 ' Immune Health (Supplements)' ' Mental Acuity (Supplements)']
```

Observation:

- Noticed data errors like ' Fruit' & 'Fruit' (spacing issue) ' Pomegranate'& ' pomegranate' (lower case & upper case). Need to rectify them before analysis

```
#Converting all flovors to lower case to rectify case-sensitive errors
df_product_launch["Flavor (Calculated)"] = df_product_launch["Flavor
(Calculated)"].apply( lambda x: (str(x).lower()))

df_product_launch.head(5)
```

```
   Product id                                    Flavor Market
Subcategory  \
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
1           2                       Fruit, not specified
Carbonates

  Launch Date Country       Region                Positioning  \
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
```

```
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
1  01-01-2001     WE2  West Europe  Convenience - Consumption

  Flavor (Calculated) Flavor type (Calculated)  \
0               herbs            not specified
0               herbs            not specified
0               fruit            not specified
0               fruit            not specified
1               fruit            not specified

  Positioning Subcategory (Calculated)
0                          Low Calorie
0                              Natural
0                          Low Calorie
0                              Natural
1            Convenience - Consumption
```

```
#Strip leading and tailing spaces in data
df_product_launch["Flavor (Calculated)"]  = df_product_launch["Flavor
(Calculated)"].str.strip()

print("After Data Cleaning:")
print("-"*50)
print(df_product_launch["Flavor (Calculated)"].unique())
```

```
After Data Cleaning:
--------------------------------------------------
['herbs' 'fruit' 'lemon' 'honey' 'ginger' 'mango' 'passion fruit'
'apple'
 'pineapple' 'guava' 'cherry' 'vanilla' 'orange' 'superfruit' 'berry'
 'nan' 'apricot' 'guarana' 'coffee' 'grape' 'grapefruit' 'peach'
'cola'
 'rose' 'melon' 'tea' 'vegetables' 'strawberry' 'raspberry'
'cranberry'
 'mandarin' 'mint' 'wildberry' 'fennel' 'coriander' 'cinnamon' 'pear'
 'plum' 'unflavored' 'citrus' 'tropical fruit' 'japanese' 'rice'
'prune'
 'carrot' 'lime' 'caramel' 'aloe vera' 'spices' 'cocoa' 'elderflower'
 'lemonade' 'blackcurrant' 'banana' 'prickly pear' 'chocolate'
 'ginger beer' 'fig' 'boysenberry' 'kiwi' 'ginseng' 'exotic fruit'
 'coconut' 'soy' 'beer' 'tangerine' 'pumpkin' 'nougat' 'vinegar'
'fish'
 'beetroot' 'blueberry' 'balm' 'broccoli' 'licorice' 'echinacea'
 'forest fruit' 'peppermint' 'bilberry' 'papaya' 'tomato' 'chili'
'pepper'
 'watermelon' 'cloves' 'tamarind' 'thyme' 'nata de coco' 'elderberry'
 'milk' 'gingko' 'cream' 'rhubarb' 'chocolate fudge' 'root beer'
 'cream soda' 'cider' 'cherry blossom (sakura)' 'onion' 'honeydew'
 'water chestnut' 'sweet & sour' 'lemongrass' 'sugar' 'grains' 'basil'
 'date' 'cappuccino' 'mexican' 'artichoke' 'nut' 'ginger ale'
'nectarine'
```

'bubble gum' 'cocktail' 'liqueur' 'malt' 'thai' 'tiramisu' 'lotus'
'oriental' 'redcurrant' 'jasmine' 'chinese' 'wine' 'radish' 'olive'
'cactus' 'bean' 'orange blossom' 'raisin' 'noni' 'fenugreek'
'lavender'
'spearmint' 'chrysanthemum' 'barley' 'rosehip' 'watercress'
'hibiscus'
'toffee' 'indian' 'galangal' 'cardamon' 'clementine' 'passion flower'
'bamboo shoot' 'juniper' 'mushroom' 'pimento' 'honeysuckle'
'cooling sensation' 'melissa' 'cumin' 'salt' 'turnip greens' 'sesame'
'cucumber' 'corn' 'vodka' 'tutti frutti' 'turmeric' 'champagne'
'birch tree' 'bergamot' 'sage' 'persimmon' 'woodruff' 'asian'
'meringue'
'feijoa' 'longan' 'saffron' 'pandan' 'herb' 'seeds' 'burdock'
'dandelion'
'yogurt' 'gingerbread' 'custard' 'cassis' 'moroccan' 'spinach'
'geranium'
'asparagus' 'kale' 'beetroot greens' 'worcestershire' 'egg' 'wheat'
'celery' 'anise' 'rum' 'bluecurrant' 'garlic' 'spirit' 'turnip'
'cabbage'
'pie' 'oregano' 'cookies & cream' 'jaeggermeister' 'nettle' 'candy
corn'
'physalis' 'rosemary' 'syrup' 'ube' 'nutmeg' 'tamarillo'
'brussels sprout' 'turkey' 'lettuce' '' 'tumeric' 'yam' 'maple'
'cheese'
'marshmallow' 'buttermilk' 'saffrom milk' 'beans' 'shellfish' 'egg
nog'
'wintergreen' 'eucalyptus' 'menthol' 'kombacha' 'potato' 'cake'
'tayberry' 'sorrel' 'spanish' 'oil' 'paprika' 'carob' 'eggplant'
'curry'
'arrowroot' 'lotus root' 'barbecue' 'nuts' 'squash' 'sea' 'italian'
'akee' 'caraway' 'pitanga' 'seaweed' 'marjoram' 'peanut butter'
'curacao'
'kiwano' 'cupuacu' 'shaddock' 'loquat' 'giant granadilla' 'hogplum'
'bloody mary' 'dulce de leche' 'rosewater' 'valerian' 'rose petal'
'whortleberry' 'roasted' 'triple sec' 'caribbean' 'marrow' 'galanga'
'masala' 'greek' 'korean' 'courgette/squash' 'martini' 'zucchini'
'peach melba' 'peach palm' 'water' 'medlar' 'greengage'
'mediterranean'
'collard greens' 'butterscotch' 'jamaica cherry' 'praline' 'parsnip'
'curcumin' 'acerola' 'sake' 'garden greens' 'butter' 'dill' 'avocado'
'violet' 'celeriac' 'canteloupe' 'appletest' 'naartjie' 'tangelo'
'osmanthus' 'yeast' 'rambutan' 'cookie dough' 'huckleberry'
'sweetsop'
'chokeberry' 'acacia' 'vermouth' 'graham cracker' 'taro' 'tequila'
'chocolate chip' 'tarragon' 'cassia' 'satsuma' 'dana' 'oreo'
'brambleberry' 'cotton candy' 'french' 'parsley' 'chocolate chip
cookie'
'creme brulee' 'pitahaya' 'oats' 'cajun' 'bael fruit' 'pea' 'yuca'
'laurel' 'brandy' 'sala' 'molasses' 'currant' 'bay leaf' 'bacon'
'jaboticaba' 'seafood' 'buffalo wing' 'chicken' 'pork' 'spare ribs'

```
 'beef' 'hickory' 'teriyaki' 'sausage' 'duck' 'garam masala' 'veal'
'ham'
 'lamb' 'meat' 'gourd' 'indian jujube' 'agave' 'black sapote'
'aniseed'
 'cayenne' 'tabasco' 'litchi' 'savory' 'jonagold' 'barbados cherry'
 'kefir' 'shortbread' 'wasabi' 'moringa' 'gin' 'cowberry' 'red fruits'
 'fudge' 'wax jambu' 'cauliflower' 'butter pecan' 'matrimony vine'
 'smoked' 'linden blossom' 'alfalfa' 'rose apple' 'tandoori'
 'caramel salted' 'pomegranate' 'pink grapefruit' 'sloeberry' 'salak'
 'cherry blossom' 'caffeine' 'plantain' 'apple cobbler' 'allspice'
 'star anise' 'waxberry' 'mocha' 'honey fungus' 'whiskey' 'aroma'
 'sprouts' 'hawthorn' 'grand marnier' 'bayberry' 'jamaica' 'bignay'
 'mabolo' 'vegetable marrow' 'cape gooseberry' 'peas' 'nashi pear'
 'sweet grenadilla' 'molokhia' "s'mores" 'cherry apple'
 'chocolate hazelnut' 'morel' 'tomatillo' 'tree tomato' 'guyabano'
 'mesquite' 'swiss chard' 'sriracha' 'pickle' 'soda']
```

```
#Converting all flovors to lower case to rectify case-sensitive errors
df_product_launch["Flavor type (Calculated)"] =
df_product_launch["Flavor type (Calculated)"].apply( lambda x:
(str(x).lower()))
#Strip leading and tailing spaces in data
df_product_launch["Flavor type (Calculated)"]  =
df_product_launch["Flavor type (Calculated)"].str.strip()
print("After Data Cleaning:")
print("-"*50)
print(df_product_launch['Flavor type (Calculated)'].unique())
```

```
After Data Cleaning:
--------------------------------------------------
['not specified' 'none' 'red' 'pomegranate' 'blackberry' 'acai'
 'blueberry' 'blood' 'green' 'lychee' 'cappuccino' 'mixed' 'cranberry'
 'lime' 'blue' 'pink' 'berry- blackberry' 'white' 'mocha'
 'latte macchiatto' 'black' 'aronia (chokeberry)' 'chinotto' 'lemon'
 'gooseberry' 'espresso' 'french' 'latte' 'hawthorn' 'jujube' 'milk'
 'bitter' 'herbal' 'wildberry' 'wild' 'oolong' 'acerola' 'fermented'
 'cactus' 'mulberry' 'dragonfruit' 'rooibos' 'earl grey' 'macadamia'
 'pina colada' 'irish cream' 'almond' 'amaretto' 'arabica' 'rose'
 'mangosteen' 'quince' 'boysenberry' 'agave' 'mung' 'walnut' 'peanut'
 'chamomile' 'pomelo' 'soursop & guanabana' 'rosehip' 'meringue'
'cashew'
 'hazelnut' 'java' 'brown' 'starfruit' 'robusta' 'apple cider' 'dark'
 'mint' 'azuki' 'matcha' 'chestnut' 'yellow' 'elderberry' 'calamansi'
 'jasmine' 'cafe au lait' 'soy' 'bell red' 'habanero' 'bloody mary'
'key'
 'carambola' 'rosemary' 'jackfruit' 'dark roast' 'hemp' 'chicory'
 'condensed' 'seabuckthorn' 'merlot' 'buckwheat' 'pawpaw' 'sicilian'
 'malt' 'yuzu' 'mate' 'durum' 'goji' 'oats' 'assam' 'martini'
'loganberry'
 'apple' 'margarita' 'cane' 'red wine' 'woodberry' 'granny smith'
'lager'
```

```
 'poppy' 'dutch' 'muscat' 'kumquat' 'lingon' 'colombia' 'ale' 'swiss'
 'sour' 'cap' 'bourbon' 'mudslide' 'clam' 'jalapeno' 'golden' 'sweet'
 'osmanthus' 'blackforest' 'cherimoya' 'cowberry' 'coconut' 'syrup'
 'darjeeling' 'yumberry' 'pistachio' 'guatemala' 'butternut' 'salt'
 'mulled' 'café au lait' 'marula' 'rye' 'cosmopolitan' 'mungo' 'cider'
 'mojito' 'sun-dried' 'kabuse' 'plum' 'baobab' 'tequila' 'chai'
'multi'
 'dragonfruit- mango- pineapple' 'valencia' 'key lime' 'mocha java'
 'hogplum' 'daiquiri' 'vodka' 'olive' 'whipped' 'chardonnay'
'marocchino'
 'mai tai' 'fennel' 'flaxseed' 'caja' 'shrimp' 'kombucha' 'boletus'
 'cocoa' 'peri peri' 'cloudberry' 'madagascan' 'caipirinha' 'pine'
 'kahlua' 'pecan' 'marion' 'sweetener' 'verbena' 'custard apple (noi
naa)'
 'sangria' 'camu camu' 'barberry' 'soar' 'acacia' 'fructose'
'cheesecake'
 'irish coffee' 'mirabelle' 'sambuca' 'irish' 'senna' 'roasted'
 'carambola/starfruit' 'oatmeal' 'belgian' 'morello' 'mascarpone'
 'mint julep' 'wholegrain' 'durian' 'maraschino' 'aronia (chokeberry('
 'sauvignon blanc' 'curacao' 'cabernet sauvignon' 'pear' 'rice'
 'hamburger' 'cheddar' 'shiitake' 'prosciutto' 'champignon'
'gorgonzola'
 'brie' 'oyster' 'moringa' 'purple' 'goldenberry' 'ginger' 'sunflower'
 'long grain' 'wheatgrass' 'natural' 'cherry' 'stevia' 'sponge'
'marzen'
 'amaranth' 'ruby' 'romaine' 'chia' 'doum palm' 'caramelized'
'linseed'
 'yacon' 'marsala' 'truffle' 'ethiopia' 'blonde' 'apple-red' 'quinoa'
 'buckthorn' '' 'cream' 'wineberry' 'celery' 'granola' 'grape'
'crowberry'
 'red velvet' 'bean' 'lucuma' 'pinot' 'millet' 'maqui (chilean
wineberry)'
 'bran' 'pumpkin' 'fried' 'pale ale' 'pinot grigio' 'tahitian' 'beach'
 'shiraz' 'pin-head' 'chinese']
```

```
#Strip leading and tailing spaces in data
df_product_launch["Positioning Subcategory (Calculated)"]  =
df_product_launch["Positioning Subcategory (Calculated)"].str.strip()
print("After Data Cleaning:")
print("-"*50)
print(df_product_launch['Positioning Subcategory
(Calculated)'].unique())
```

```
After Data Cleaning:
--------------------------------------------------
['Low Calorie' 'Natural' 'Convenience - Consumption'
 '100% Not from Concentrate' 'No Added Sugar' 'No
Additives/Preservatives'
 'Nectars (25-99% juice)' 'Vitamin/Mineral Fortified'
 'Digestive/Gut Health' 'High/Source of Fibre' 'Dry' 'Low Fat'
 'Low Sodium' 'Low Carb' 'Antioxidant' 'Immune Health'
```

```
 'Convenience - Packaging' 'Low Sugar' 'Female' 'Sugar Free'
 'Anti-Aging/Aging-Well' 'Skin Health' 'Energy/Alertness'
 'Convenience - Easy-to-Prepare' 'Heart Health'
 'Juice Drinks (up to 25% juice)' 'Vegetarian' 'Convenience - Time
Saving'
 'Economy' '100% Reconstituted' 'Ethical - Environment'
 'Ethical - Packaging' 'Sports & Recovery' 'Novel and Fun' 'nan'
 'Children (5-12 years)' 'Convenience - Ready Prepared'
 'Weight Management' 'Added Protein' 'High/Source of Protein'
 'Indulgent and Premium' 'Ethnic and Exotic' 'Added Calcium'
'Traditional'
 'Gluten Free' 'Allergy' 'Eye Health' 'Bone Health' 'Lactose Free'
 'Added Fibre' 'Sticks' 'Organic' 'Low GI' 'Functional' 'not
specified'
 'Single Shot' 'Soy Foods' 'GMO Free' 'Brain Health' 'DHA' 'Joint
Health'
 'Diabetic' 'Seasonal/In-Out Products' 'Male' 'Low Cholesterol'
 'Added Iron' 'Microwaveable' 'Co-Branding' 'Halal' 'Kosher' 'HFCS
Free'
 'Omega-3' 'Ethical - Human' 'Refill' 'Prebiotic' 'Wholegrain'
'economy'
 'Novel and fun' '100% Frozen' 'Oral Health' 'No Trans Fats'
 'Nectars (25-99% Juice)' '100% Not from concentrate' 'Packaging'
 'Seniors (55+)' '100% not from Concentrate' 'Ethical - Not Specific'
 'Ethical - Animal/Fish & Bird' 'Digestive/Liver Health (Supplements)'
 'Children (Supplements)' 'nectars (25-99% juice)'
 'Convenience -  Consumption' 'Growing-Up Milk (1+ year)'
'Consumption'
 'Time Saving' 'Price Premium > or equal to 150%'
 'Price Premium > or equal to 200%' 'Ready Prepared' 'Easy-to-Prepare'
 'Convenience -  Packaging' 'Energy and Stamina (Supplements)'
 'Pre-prepared' 'Convenience of Usage' 'Convenience -  Easy-to-
Prepare'
 'Brain-Mood Health (Supplements)' 'Immune Health (Supplements)'
 'Mental Acuity (Supplements)']
```

```
df_product_launch.head(5)
```

```
   Product id                                     Flavor Market
Subcategory  \
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
1           2                       Fruit, not specified
Carbonates
```

```
  Launch Date Country       Region                Positioning  \
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
0  01-01-2001     WE1  West Europe       Low Calorie, Natural
1  01-01-2001     WE2  West Europe  Convenience - Consumption

  Flavor (Calculated) Flavor type (Calculated)  \
0               herbs            not specified
0               herbs            not specified
0               fruit            not specified
0               fruit            not specified
1               fruit            not specified

  Positioning Subcategory (Calculated)
0                          Low Calorie
0                              Natural
0                          Low Calorie
0                              Natural
1            Convenience - Consumption
```

Checking for data errors in other columns

```
print(df_product_launch['Market Subcategory'].unique())
```

```
['Other Soft Drinks' 'Carbonates' 'Juice & Juice Drinks'
 'Drink Concentrates & Mixes' 'Bottled Water - Flavoured' 'Energy
Drinks'
 'Iced Coffee' 'Iced Tea' 'Sports Drinks RTD'
 'Bottled Water - Unflavoured' 'Sports Others' 'Sports Powders'
 'Other Soft Drinks  ' 'Sports Supplements']
```

```
print(df_product_launch['Country'].unique())
```

```
['WE1' 'WE2' 'NA1' 'AU1' 'WE3' 'NA2' 'EE1' 'WE4' 'WE5' 'WE6' 'EE2'
'WE7'
 'A1' 'LA1' 'EE3' 'WE8' 'WE9' 'AF1' 'LA2' 'A2' 'A3' 'LA3' 'LA4' 'A4'
'ME1'
 'A5' 'EE4' 'EE5' 'A6' 'AF2' 'WE10' 'A7' 'ME2' 'WE11' 'EE6' 'A8' 'LA5'
 'A9' 'EE7' 'AF3' 'A10' 'A11' 'EE8' 'ME3' 'EE9' 'EE10' 'WE12' 'LA6'
'WE13'
 'A12' 'WE14' 'ME4' 'A13' 'ME5' 'WE15' 'A14' 'AU2' 'LA7' 'EE11' 'LA8'
 'EE12' 'LA9' 'WE16' 'A15' 'AF4' 'LA10' 'ME6' 'EE13' 'LA11' 'EE14'
'AF5'
 'LA12' 'WE17' 'WE18' 'WE19' 'EE15' 'ME7' 'ME8' 'LA13' 'ME9' 'LA14'
'LA15'
 'LA16' 'EE16' 'A16' 'EE17' 'LA17' 'NA3' 'AF6' 'ME10' 'ME11' 'AF7'
'AF8'
 'AF9' 'AF10' 'ME12' 'EE18' 'AF11' 'AF12' 'WE20' 'LA18' 'AF13' 'AF14'
 'LA19' 'A17' 'EE19' 'ME13' 'AF15']
```

```
print(df_product_launch['Region'].unique())
```

```
['West Europe' 'North America' 'Australasia' 'East Europe' 'Asia'
 'Latin America' 'Africa' 'Middle East']
```

```
print(df_product_launch['Positioning'].unique())
```

```
['Low Calorie, Natural' 'Convenience - Consumption'
 '100% Not from Concentrate, Convenience - Consumption, Natural' ...
 'Indulgent and Premium, Heart Health, Kosher'
 'Low Calorie, Ethical - Packaging, Convenience - Consumption, GMO
Free'
 'Kosher, Juice Drinks (up to 25% juice), Allergy, Gluten Free,
Organic, Omega-3, GMO Free']
```

Calculating Month and Year for Launch date

```
#type(df_product_launch['Launch Date'][1])
df_product_launch['Launch Date'] =
pd.to_datetime(df_product_launch['Launch Date'], format='%d-%m-%Y')

df_product_launch['Month (Calculated)'] = df_product_launch['Launch
Date'].dt.month_name()

df_product_launch['Year (Calculated)'] = df_product_launch['Launch
Date'].dt.year

df_product_launch.head(5)
```

```
   Product id                                     Flavor Market
Subcategory  \
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
1           2                       Fruit, not specified
Carbonates

  Launch Date Country       Region                Positioning  \
0  2001-01-01     WE1  West Europe       Low Calorie, Natural
0  2001-01-01     WE1  West Europe       Low Calorie, Natural
0  2001-01-01     WE1  West Europe       Low Calorie, Natural
0  2001-01-01     WE1  West Europe       Low Calorie, Natural
1  2001-01-01     WE2  West Europe  Convenience - Consumption

  Flavor (Calculated) Flavor type (Calculated)  \
0               herbs            not specified
0               herbs            not specified
0               fruit            not specified
```

```
0               fruit            not specified
1               fruit            not specified

  Positioning Subcategory (Calculated) Month (Calculated)  Year
(Calculated)
0                          Low Calorie            January
2001
0                              Natural            January
2001
0                          Low Calorie            January
2001
0                              Natural            January
2001
1             Convenience - Consumption            January
2001
```

Preprocessed Data (Product Launch dataset):

```
print("Pre-processed Data: (product_launch)")
df_product_launch.head(15)
```

```
Pre-processed Data: (product_launch)

   Product id                                        Flavor  \
0           1  Herbs, not specified; Fruit, not specified
0           1  Herbs, not specified; Fruit, not specified
0           1  Herbs, not specified; Fruit, not specified
0           1  Herbs, not specified; Fruit, not specified
1           2                         Fruit, not specified
2           3                         Lemon; Honey; Ginger
2           3                         Lemon; Honey; Ginger
2           3                         Lemon; Honey; Ginger
2           3                         Lemon; Honey; Ginger
2           3                         Lemon; Honey; Ginger
2           3                         Lemon; Honey; Ginger
2           3                         Lemon; Honey; Ginger
2           3                         Lemon; Honey; Ginger
2           3                         Lemon; Honey; Ginger
3           4                        Mango; Passion Fruit

     Market Subcategory Launch Date Country       Region  \
0     Other Soft Drinks  2001-01-01     WE1  West Europe
0     Other Soft Drinks  2001-01-01     WE1  West Europe
0     Other Soft Drinks  2001-01-01     WE1  West Europe
0     Other Soft Drinks  2001-01-01     WE1  West Europe
1            Carbonates  2001-01-01     WE2  West Europe
2  Juice & Juice Drinks  2001-01-01     WE2  West Europe
2  Juice & Juice Drinks  2001-01-01     WE2  West Europe
2  Juice & Juice Drinks  2001-01-01     WE2  West Europe
2  Juice & Juice Drinks  2001-01-01     WE2  West Europe
```

```
2  Juice & Juice Drinks  2001-01-01     WE2  West Europe
2  Juice & Juice Drinks  2001-01-01     WE2  West Europe
2  Juice & Juice Drinks  2001-01-01     WE2  West Europe
2  Juice & Juice Drinks  2001-01-01     WE2  West Europe
2  Juice & Juice Drinks  2001-01-01     WE2  West Europe
3  Juice & Juice Drinks  2001-01-01     WE2  West Europe

                                          Positioning Flavor
(Calculated)  \
0                              Low Calorie, Natural
herbs
0                              Low Calorie, Natural
herbs
0                              Low Calorie, Natural
fruit
0                              Low Calorie, Natural
fruit
1                          Convenience - Consumption
fruit
2  100% Not from Concentrate, Convenience - Consu...
lemon
2  100% Not from Concentrate, Convenience - Consu...
lemon
2  100% Not from Concentrate, Convenience - Consu...
lemon
2  100% Not from Concentrate, Convenience - Consu...
honey
2  100% Not from Concentrate, Convenience - Consu...
honey
2  100% Not from Concentrate, Convenience - Consu...
honey
2  100% Not from Concentrate, Convenience - Consu...
ginger
2  100% Not from Concentrate, Convenience - Consu...
ginger
2  100% Not from Concentrate, Convenience - Consu...
ginger
3  100% Not from Concentrate, Convenience - Consu...
mango

  Flavor type (Calculated) Positioning Subcategory (Calculated)  \
0            not specified                           Low Calorie
0            not specified                               Natural
0            not specified                           Low Calorie
0            not specified                               Natural
1            not specified           Convenience - Consumption
2                     none           100% Not from Concentrate
2                     none           Convenience - Consumption
2                     none                               Natural
2                     none           100% Not from Concentrate
```

```
2                    none            Convenience - Consumption
2                    none                              Natural
2                    none            100% Not from Concentrate
2                    none            Convenience - Consumption
2                    none                              Natural
3                    none            100% Not from Concentrate

  Month (Calculated)  Year (Calculated)
0            January               2001
0            January               2001
0            January               2001
0            January               2001
1            January               2001
2            January               2001
2            January               2001
2            January               2001
2            January               2001
2            January               2001
2            January               2001
2            January               2001
2            January               2001
2            January               2001
2            January               2001
3            January               2001
```

```
#Exporting pre-processed data to csv file
preprocessed_productLaunch = df_product_launch
df_product_launch.to_csv("product_launch_dataset_preprocessed.csv")
```

Exporatory Data Analysis: Product Launch Dataset

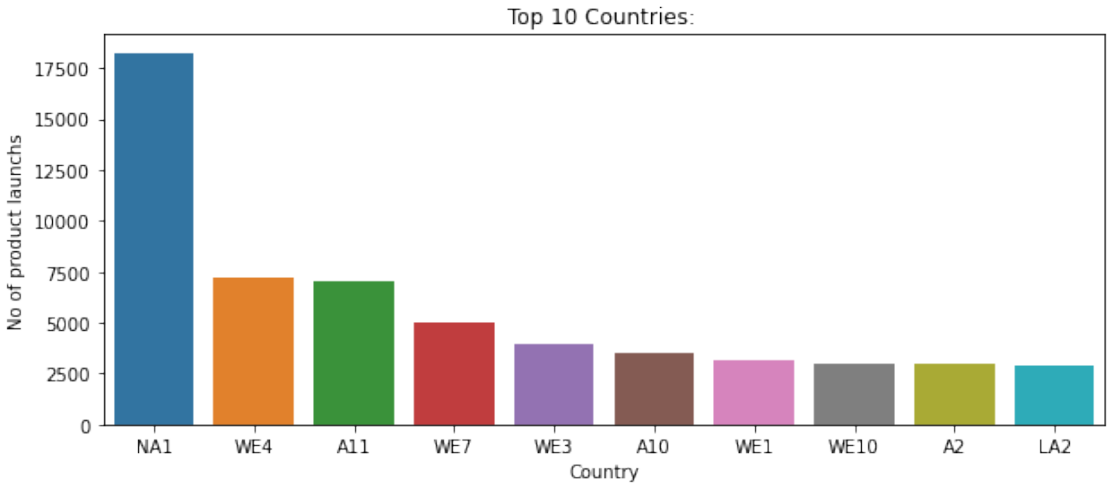
Univariant Analysis

```
print("Countries and their no of product launches")
df_countries = preprocessed_productLaunch[["Product
id","Country"]].groupby("Country",as_index = False).agg({"Product id":
"nunique"})
df_countries[["Country","No of product launchs"]]= df_countries
df_countries = df_countries.drop(columns = "Product id")
df_countries.sort_values(by = "No of product launchs", ascending =
False,inplace = True)
print(df_countries.columns)
df_countries
```

```
Countries and their no of product launches
Index(['Country', 'No of product launchs'], dtype='object')

    Country  No of product launchs
85      NA1                  18244
102     WE4                   7179
2       A11                   7015
```

```
105     WE7                   5016
101     WE3                   3906
..       ...                   ...
44     EE19                     1
100    WE20                     1
23     AF15                     1
87      NA3                     1
28      AF6                     1

[108 rows x 2 columns]
```

```
x = df_countries['Country'] [0:10]
y = df_countries['No of product launchs'][0:10]
sns.barplot(x,y)
fig = plt.gcf() #getcurrentfigure
fig.set_size_inches(10,4)
plt.title('Top 10 Countries: ')
plt.show()
```



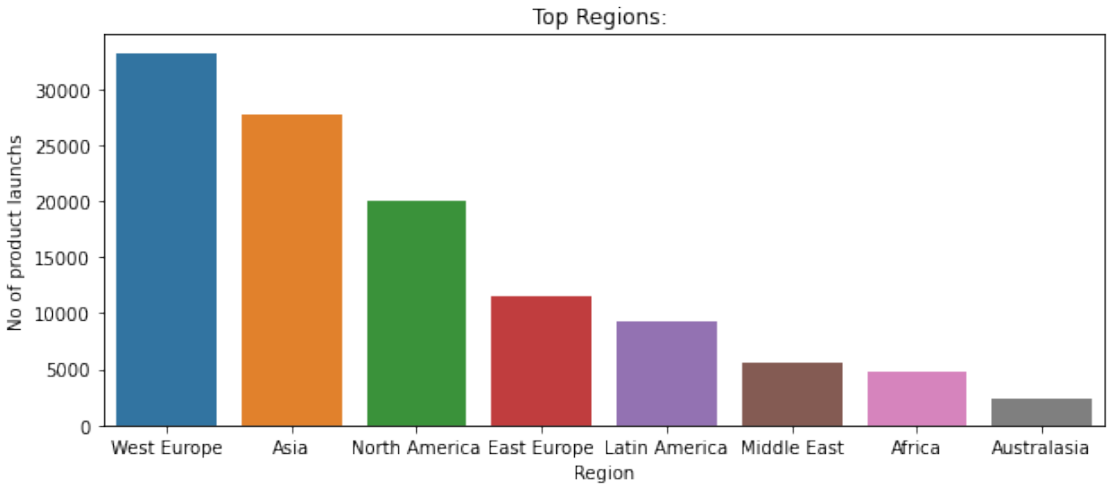

```
print("Regions and their no of product launches")
df_regions = preprocessed_productLaunch[["Product
id","Region"]].groupby("Region",as_index = False).agg({"Product id":
"nunique"})
df_regions[["Region","No of product launchs"]]= df_regions
df_regions = df_regions.drop(columns = "Product id")
df_regions.sort_values(by = "No of product launchs", ascending =
False,inplace = True)
print(df_regions.columns)
df_regions
```

```
Regions and their no of product launches
Index(['Region', 'No of product launchs'], dtype='object')
```

```
          Region  No of product launchs
7    West Europe                  33218
1           Asia                  27775
```

```
6  North America                  20085
3     East Europe                 11467
4  Latin America                   9348
5     Middle East                  5636
0          Africa                  4790
2     Australasia                  2370
```

```
x = df_regions['Region'] [0:10]
y = df_regions['No of product launchs'][0:10]
sns.barplot(x,y)
fig = plt.gcf() #getcurrentfigure
fig.set_size_inches(10,4)
plt.title('Top Regions: ')
plt.show()
```



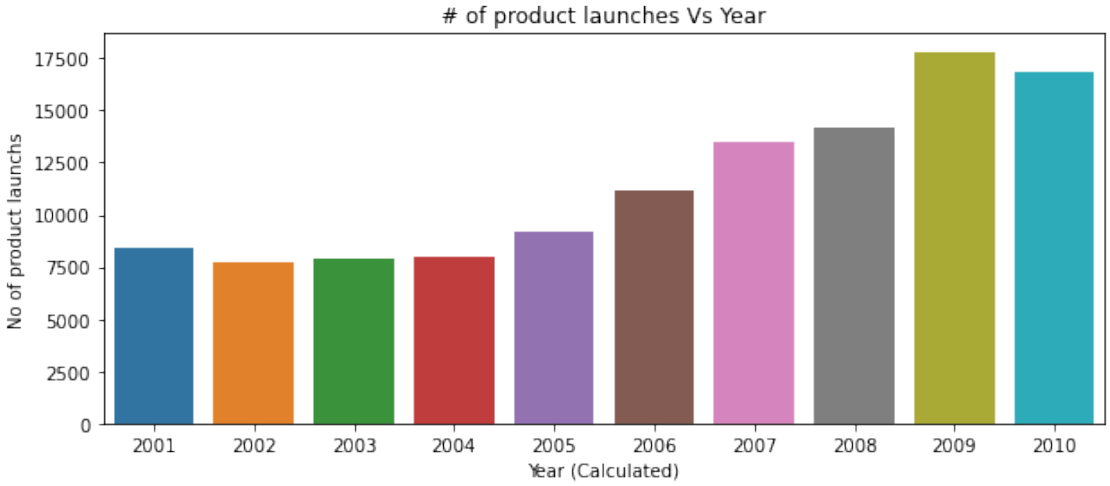

```
print("No of product launches Vs Year")
df_yearDistr = preprocessed_productLaunch[["Product id","Year
(Calculated)"]].groupby("Year (Calculated)",as_index =
False).agg({"Product id": "nunique"})
df_yearDistr[["Year (Calculated)","No of product launchs"]]=
df_yearDistr
df_yearDistr = df_yearDistr.drop(columns = "Product id")
df_yearDistr.sort_values(by = "Year (Calculated)", ascending =
False,inplace = True)
print(df_yearDistr.columns)
df_yearDistr
```

```
No of product launches Vs Year
Index(['Year (Calculated)', 'No of product launchs'], dtype='object')
```

```
   Year (Calculated)  No of product launchs
9               2010                  16834
8               2009                  17813
7               2008                  14135
6               2007                  13492
5               2006                  11151
```

```
4                2005                    9169
3                2004                    8005
2                2003                    7900
1                2002                    7757
0                2001                    8433
```

```
x = df_yearDistr['Year (Calculated)']
y = df_yearDistr['No of product launchs']
sns.barplot(x,y)
fig = plt.gcf() #getcurrentfigure
fig.set_size_inches(10,4)
plt.title('# of product launches Vs Year')
plt.show()
```



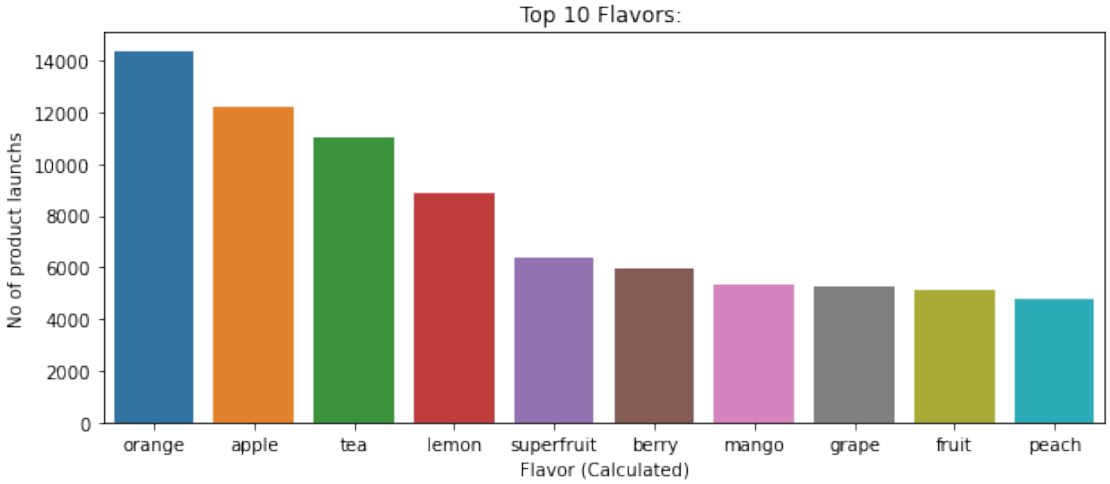

```
print("Flavors distribution across products")
df_flavorsDistr = preprocessed_productLaunch[["Product id","Flavor
(Calculated)"]][preprocessed_productLaunch['Flavor (Calculated)'] !=
'unflavored'].groupby("Flavor (Calculated)",as_index =
False).agg({"Product id": "nunique"})
df_flavorsDistr[["Flavor (Calculated)","No of product launchs"]]=
df_flavorsDistr
df_flavorsDistr = df_flavorsDistr.drop(columns = "Product id")
df_flavorsDistr.sort_values(by = "No of product launchs", ascending =
False,inplace = True)
print(df_flavorsDistr.columns)
df_flavorsDistr
```

```
Flavors distribution across products
Index(['Flavor (Calculated)', 'No of product launchs'],
dtype='object')
```

```
    Flavor (Calculated)  No of product launchs
265              orange                  14386
10                apple                  12201
373                 tea                  11020
208               lemon                   8912
```

```
358          superfruit                    6357
..                  ...                     ...
206              laurel                       1
214      linden blossom                       1
216              litchi                       1
221              mabolo                       1
166      graham cracker                       1

[421 rows x 2 columns]
```

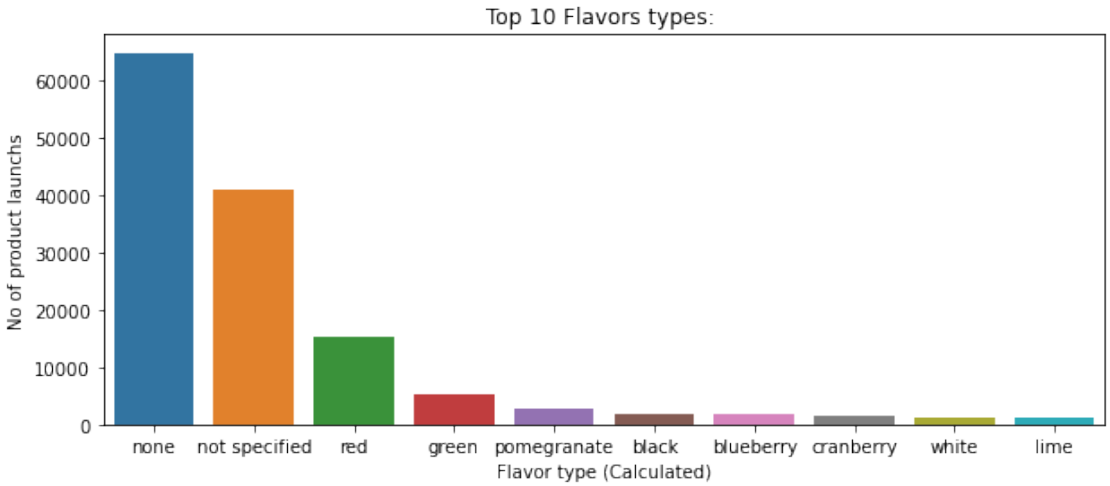
```
x = df_flavorsDistr['Flavor (Calculated)'] [0:10]
y = df_flavorsDistr['No of product launchs'][0:10]
sns.barplot(x,y)
fig = plt.gcf() #getcurrentfigure
fig.set_size_inches(10,4)
plt.title('Top 10 Flavors: ')
plt.show()

top10_flavors = df_flavorsDistr['Flavor (Calculated)'] [0:10]
```




```
print("Flavors types distribution across products")
df_flavorsTypeDistr = preprocessed_productLaunch[["Product id","Flavor
type (Calculated)"]].groupby("Flavor type (Calculated)",as_index =
False).agg({"Product id": "nunique"})
df_flavorsTypeDistr[["Flavor type (Calculated)","No of product
launchs"]]= df_flavorsTypeDistr
df_flavorsTypeDistr = df_flavorsTypeDistr.drop(columns = "Product id")
df_flavorsTypeDistr.sort_values(by = "No of product launchs",
ascending = False,inplace = True)
print(df_flavorsTypeDistr.columns)
df_flavorsTypeDistr

Flavors types distribution across products
Index(['Flavor type (Calculated)', 'No of product launchs'],
dtype='object')
```

```
    Flavor type (Calculated)  No of product launchs
186                     none                  64716
187            not specified                  40835
216                      red                  15143
114                    green                   5321
208              pomegranate                   2826
..                       ...                    ...
39                 buckthorn                      1
162                mascarpone                     1
161                    marzen                     1
158                   marsala                     1
168                    millet                     1

[271 rows x 2 columns]
```

```
x = df_flavorsTypeDistr['Flavor type (Calculated)'] [0:10]
y = df_flavorsTypeDistr['No of product launchs'][0:10]
sns.barplot(x,y)
fig = plt.gcf() #getcurrentfigure
fig.set_size_inches(10,4)
plt.title('Top 10 Flavors types: ')
plt.show()
```



Bi-Variant Analysis

```
print("No of product launches Vs Year Vs Month")
df_MonthDistr = preprocessed_productLaunch[["Product id","Year
(Calculated)","Month (Calculated)"]].groupby(["Year
(Calculated)","Month (Calculated)"],as_index = False,sort =
True).agg({"Product id": "nunique"})
#df_MonthDistr[["Year (Calculated)","No of product launchs"]]=
df_MonthDistr
#df_MonthDistr = df_MonthDistr.drop(columns = "Product id")
#df_MonthDistr.sort_values(by = "Year (Calculated)", ascending =
False,inplace = True)
```

```
#print(df_MonthDistr.columns)
df_MonthDistr
```

No of product launches Vs Year Vs Month

```
   Year (Calculated) Month (Calculated)  Product id
0               2001              January        8433
1               2002              January        7757
2               2003              January        7900
3               2004              January        8005
4               2005              January        9169
5               2006              January       11151
6               2007              January       13492
7               2008              January       14135
8               2009              January       17813
9               2010              January       16834
```

```
preprocessed_productLaunch["Month (Calculated)"].nunique()
```

```
1
```

Observation:

- Only January data is available in given dataset

```
preprocessed_productLaunch.head(4)
```

```
   Product id                                     Flavor Market
Subcategory  \
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks

  Launch Date Country       Region           Positioning Flavor
(Calculated)  \
0  2001-01-01     WE1  West Europe  Low Calorie, Natural
herbs
0  2001-01-01     WE1  West Europe  Low Calorie, Natural
herbs
0  2001-01-01     WE1  West Europe  Low Calorie, Natural
fruit
0  2001-01-01     WE1  West Europe  Low Calorie, Natural
fruit

  Flavor type (Calculated) Positioning Subcategory (Calculated)  \
0            not specified                          Low Calorie
0            not specified                              Natural
0            not specified                          Low Calorie
```

```
0            not specified                                Natural

  Month (Calculated)  Year (Calculated)
0            January               2001
0            January               2001
0            January               2001
0            January               2001
```

```
print("Understanding mapping of Flavor and Flavor types")
invalid_entries = ["", " ", "none", "not specified"]
df_flavor_flavortypes = preprocessed_productLaunch[["Product
id","Flavor (Calculated)","Flavor type (Calculated)"]]
[~preprocessed_productLaunch['Flavor type
(Calculated)'].isin(invalid_entries)]
#df_flavor_flavortypes = df_flavor_flavortypes
[~preprocessed_productLaunch['Flavor type
(Calculated)'].isin(invalid_entries)]
df_flavor_flavortypes= df_flavor_flavortypes.groupby(["Flavor
(Calculated)","Flavor type (Calculated)"],as_index = False,sort =
True, dropna = True).agg({"Product id": "nunique"})
df_flavor_flavortypes[["Flavor (Calculated)","Flavor type
(Calculated)","No of product launchs"]]= df_flavor_flavortypes
df_flavor_flavortypes = df_flavor_flavortypes.drop(columns = "Product
id")
#df_flavor_flavortypes.sort_values(by = "No of product launchs",
ascending = False,inplace = True)
print(df_flavor_flavortypes.columns)
df_flavor_flavortypes
```

```
Understanding mapping of Flavor and Flavor types
Index(['Flavor (Calculated)', 'Flavor type (Calculated)',
       'No of product launchs'],
      dtype='object')
```

```
    Flavor (Calculated) Flavor type (Calculated)  No of product
launchs
0                 apple                   golden
27
1                 apple             granny smith
30
2                 apple                    green
1667
3                 apple                      red
10544
4             asparagus                    green
2
..                  ...                      ...                    ..
.
338               wine                  sangria
3
339               wine          sauvignon blanc
```

```
1
340                wine                   shiraz
1
341                wine                    white
7
342                 yam                   purple
1

[343 rows x 3 columns]
```

```
top_flavors = list(top10_flavors[0:11])
```

```
df_flavor_flavortypes[df_flavor_flavortypes['Flavor
(Calculated)'].isin(top_flavors)].pivot(index='Flavor (Calculated)',
columns='Flavor type (Calculated)', values='No of product launchs')
```

```
Flavor type (Calculated)   acai  acerola  agave  aronia
(chokeberry(  \
Flavor (Calculated)

apple                       NaN      NaN    NaN                  NaN

berry                       NaN      NaN    NaN                  1.0

fruit                       NaN      NaN    NaN                  NaN

grape                       NaN      NaN    NaN                  NaN

lemon                       NaN      NaN    NaN                  NaN

mango                       NaN      NaN    NaN                  NaN

orange                      NaN      NaN    NaN                  NaN

peach                       NaN      NaN    NaN                  NaN

superfruit                841.0    327.0  127.0                  NaN

tea                         NaN      NaN    NaN                  NaN


Flavor type (Calculated)  aronia (chokeberry)  assam  baobab  barberry
\
Flavor (Calculated)

apple                                    NaN    NaN     NaN       NaN

berry                                  153.0    NaN     NaN       2.0
```

```
fruit                                    NaN    NaN     NaN       NaN

grape                                    NaN    NaN     NaN       NaN

lemon                                    NaN    NaN     NaN       NaN

mango                                    NaN    NaN     NaN       NaN

orange                                   NaN    NaN     NaN       NaN

peach                                    NaN    NaN     NaN       NaN

superfruit                               NaN    NaN    16.0       NaN

tea                                      NaN   50.0     NaN       NaN


Flavor type (Calculated)  berry- blackberry  bitter  ...  sweet
valencia  \
Flavor (Calculated)                                   ...

apple                                      NaN     NaN  ...    NaN
NaN
berry                                    428.0     NaN  ...    NaN
NaN
fruit                                      NaN     NaN  ...    NaN
NaN
grape                                      NaN     NaN  ...    NaN
NaN
lemon                                      NaN    86.0  ...    NaN
NaN
mango                                      NaN     NaN  ...    NaN
NaN
orange                                     NaN     NaN  ...   40.0
61.0
peach                                      NaN     NaN  ...    NaN
NaN
superfruit                                 NaN     NaN  ...    NaN
NaN
tea                                        NaN     NaN  ...    NaN
NaN

Flavor type (Calculated)  verbena  white  wildberry  wineberry
woodberry  \
Flavor (Calculated)

apple                          NaN    NaN        NaN        NaN
NaN
berry                          NaN    NaN      192.0        1.0
```

```
1.0
fruit                          NaN    NaN        NaN        NaN
NaN
grape                          NaN  681.0        NaN        NaN
NaN
lemon                          NaN    NaN        NaN        NaN
NaN
mango                          NaN    NaN        NaN        NaN
NaN
orange                         NaN    NaN        NaN        NaN
NaN
peach                          NaN   83.0        NaN        NaN
NaN
superfruit                     NaN    NaN        NaN        NaN
NaN
tea                           12.0  507.0        NaN        NaN
NaN

Flavor type (Calculated)  yacon  yumberry  yuzu
Flavor (Calculated)
apple                       NaN       NaN   NaN
berry                       NaN       NaN   NaN
fruit                       NaN       NaN   NaN
grape                       NaN       NaN   NaN
lemon                       NaN       NaN   NaN
mango                       NaN       NaN   NaN
orange                      NaN       NaN   NaN
peach                       NaN       NaN   NaN
superfruit                  1.0      50.0  62.0
tea                         NaN       NaN   NaN

[10 rows x 93 columns]
```

```
print("Understanding mapping of Flavor and No of Flavor types")
invalid_entries = ["", " ", "none", "not specified"]
df_flavor_flavortypes2 = preprocessed_productLaunch[["Flavor
(Calculated)","Flavor type (Calculated)"]]
[~preprocessed_productLaunch['Flavor type
(Calculated)'].isin(invalid_entries)]
#df_flavor_flavortypes = df_flavor_flavortypes
[~preprocessed_productLaunch['Flavor type
(Calculated)'].isin(invalid_entries)]
df_flavor_flavortypes2= df_flavor_flavortypes2.groupby(["Flavor
(Calculated)"],as_index = False, dropna = True).agg({"Flavor type
(Calculated)": "nunique"})
df_flavor_flavortypes2[["Flavor (Calculated)","No of Flavor types"]]=
df_flavor_flavortypes2
df_flavor_flavortypes2 = df_flavor_flavortypes2.drop(columns = "Flavor
type (Calculated)")
df_flavor_flavortypes2.sort_values(by = "No of Flavor types",
```

```
ascending = False,inplace = True)
print(df_flavor_flavortypes2.columns)
df_flavor_flavortypes2
```

```
Understanding mapping of Flavor and No of Flavor types
Index(['Flavor (Calculated)', 'No of Flavor types'], dtype='object')
```

```
   Flavor (Calculated)  No of Flavor types
68          superfruit                  42
9                berry                  20
69                 tea                  20
20              coffee                  18
75                wine                  13
..                 ...                 ...
36             lettuce                   1
35            lemonade                   1
15             chicken                   1
57                 rum                   1
76                 yam                   1

[77 rows x 2 columns]
```

```
df_flavor_flavortypes[df_flavor_flavortypes['Flavor (Calculated)'] ==
'superfruit'].pivot(index='Flavor (Calculated)', columns='Flavor type
(Calculated)', values='No of product launchs')
```

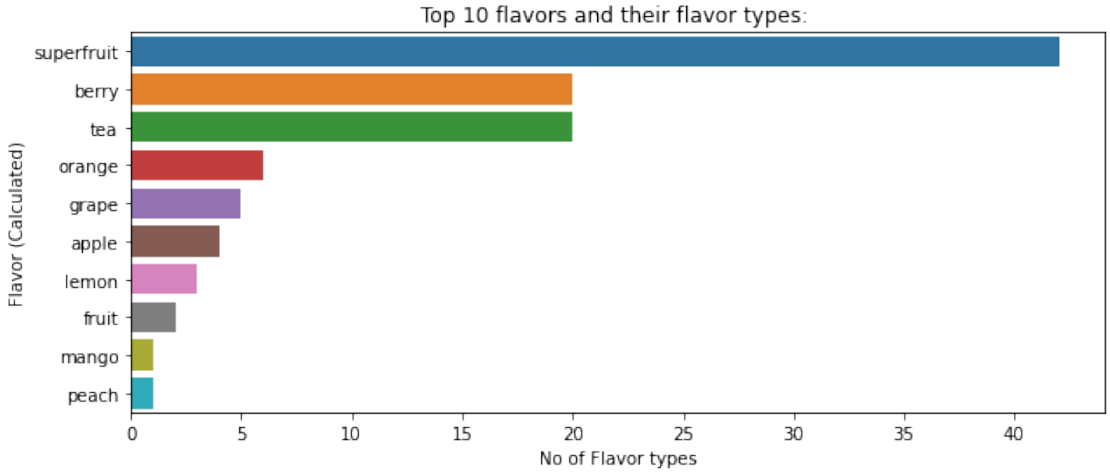
```
Flavor type (Calculated)  acai  acerola  agave  baobab  cactus
caja  \
Flavor (Calculated)

superfruit                 841      327    127      16      69     9


Flavor type (Calculated)  calamansi  camu camu  carambola  \
Flavor (Calculated)
superfruit                      108          9         16

Flavor type (Calculated)  carambola/starfruit  ...  pawpaw
pomegranate  \
Flavor (Calculated)                             ...

superfruit                                    2  ...       9
2826

Flavor type (Calculated)  pomelo  quince  seabuckthorn  soursop &
guanabana  \
Flavor (Calculated)

superfruit                   109      71            90
173
```

```
Flavor type (Calculated)  starfruit  yacon  yumberry  yuzu
Flavor (Calculated)
superfruit                       28      1        50    62

[1 rows x 42 columns]
```

```
print("Top 10 flavors and no of types they have")
df_flavor_flavortypes2[df_flavor_flavortypes2['Flavor
(Calculated)'].isin(top_flavors)]
```

```
Top 10 flavors and no of types they have

   Flavor (Calculated)  No of Flavor types
68          superfruit                  42
9                berry                  20
69                 tea                  20
47              orange                   6
28               grape                   5
0                apple                   4
34               lemon                   3
24               fruit                   2
39               mango                   1
49               peach                   1
```

```
x = df_flavor_flavortypes2[df_flavor_flavortypes2['Flavor
(Calculated)'].isin(top_flavors)]['Flavor (Calculated)']
y = df_flavor_flavortypes2[df_flavor_flavortypes2['Flavor
(Calculated)'].isin(top_flavors)]['No of Flavor types']
sns.barplot(y,x)
fig = plt.gcf() #getcurrentfigure
fig.set_size_inches(10,4)
plt.title('Top 10 flavors and their flavor types: ')
plt.show()
```



Preliminary Analysis of dataset: Positional_Category_Mapping_Dataset

```
print("creating the dataframe from
Positional_Category_Mapping_Dataset.csv file..")
df_positonal_category_mapping =
pd.read_csv('Positional_Category_Mapping_Dataset.csv')
print('Done.\n')
```

```
creating the dataframe from Positional_Category_Mapping_Dataset.csv
file..
Done.
```

```
df_positonal_category_mapping.head(10)
```

	Positioning Group	Positioning Subcategory
0	Age	Children (5-12 years)
1	Age	Seniors (55+)
2	Others	Economy
3	Choice	Halal
4	Choice	Kosher
5	Choice	Vegetarian
6	Co-Branding	Co-Branding
7	Supplements	Anti-Cancer
8	Supplements	Blood Pressure
9	Supplements	Bone Health (Supplements)

Basic Stats:

```
print("Total data ")
print("-"*50)
print("No of unique Positioning SubCategories
available:" ,df_positonal_category_mapping["Positioning
Subcategory"].nunique())
print("No of unique Positioning Groups
available:" ,df_positonal_category_mapping["Positioning
Group"].nunique())
```

```
Total data
--------------------------------------------------
No of unique Positioning SubCategories available: 113
No of unique Positioning Groups available: 16
```

Data Quality Check:

- Check for duplicates: if positioning subcategory have duplicates
- Check and validate NaN values

```
df_positonal_category_mapping['dup_bool'] =
df_positonal_category_mapping.duplicated(["Positioning Subcategory"])
dups = sum(df_positonal_category_mapping['dup_bool']) # by considering
Product ids...
```

```
print("There are {} duplicate rating entries in the data w.r.t sub-
categories..".format(dups))
```

```
There are 1 duplicate rating entries in the data w.r.t sub-
categories..
```

```
df_positonal_category_mapping
```

```
    Positioning Group             Positioning Subcategory  dup_bool
0                 Age               Children (5-12 years)     False
1                 Age                       Seniors (55+)     False
2              Others                             Economy     False
3              Choice                               Halal     False
4              Choice                              Kosher     False
..                ...                                 ...       ...
109          Ethical              Ethical - Not Specific     False
110          Ethical                 Ethical - Packaging     False
111           Choice                             GMO Free     False
112              Age  Pregnancy/Breastfeeding - Formulas     False
113      Supplements  Pregnancy/Breastfeeding - Formulas      True

[114 rows x 3 columns]
```

```
preprocessed_productLaunch [preprocessed_productLaunch['Positioning
Subcategory (Calculated)'] == 'Pregnancy/Breastfeeding - Formulas']
```

```
Empty DataFrame
Columns: [Product id, Flavor, Market Subcategory, Launch Date,
Country, Region, Positioning, Flavor (Calculated), Flavor type
(Calculated), Positioning Subcategory (Calculated), Month
(Calculated), Year (Calculated)]
Index: []
```

```
df_positonal_category_mapping =
df_positonal_category_mapping.drop(columns = "dup_bool")
```

Observation:

- sub-category:"Pregnancy/Breastfeeding - Formulas" is mapped to 2 groups:"Age, Supplements" but fortunately "Pregnancy/Breastfeeding - Formulas" is not found in our Product Lauch Dataset. Hence no data correction need in mapping data

```
print("No of columns found with NaNs : ",
sum(df_positonal_category_mapping.isnull().any()))
```

```
No of columns found with NaNs :  0
```

Analysis:

```
print("No of subcategories of each Positioning Group")
df_subcategories = df_positonal_category_mapping[["Positioning Group",
"Positioning Subcategory"]].groupby("Positioning Group",as_index =
False).agg({"Positioning Subcategory": "nunique"})
df_subcategories[["Positioning Group", "No of sub-categories"]]=
```

```
df_subcategories
df_subcategories = df_subcategories.drop(columns = "Positioning
Subcategory")
df_subcategories.sort_values(by = "No of sub-categories", ascending =
False,inplace = True)
print(df_subcategories.columns)
df_subcategories
```

```
No of subcategories of each Positioning Group
Index(['Positioning Group', 'No of sub-categories'], dtype='object')
```

	Positioning Group	No of sub-categories
15	Supplements	24
8	Health (Active)	22
9	Health (Passive)	22
11	Others	7
0	Age	6
3	Convenience	5
6	Ethical	5
10	Juice	5
1	Choice	4
12	Pleasure	4
4	Convenience – Packaging	2
5	Convenience – Preparation	2
7	Gender	2
13	Price Positioning	2
2	Co-Branding	1
14	Seasonal/In-Out Products	1

Preliminary Analysis of dataset: Flavor_Classification_Dataset

```
print("creating the dataframe from
Positional_Category_Mapping_Dataset.csv file..")
df_flavor_classification =
pd.read_csv('Flavor_Classification_Dataset.csv')
print('Done.\n')
```

```
creating the dataframe from Positional_Category_Mapping_Dataset.csv
file..
Done.
```

```
df_flavor_classification.head(10)
```

	Flavor_Group	Flavor
0	Alcohol	Vodka, Citron
1	Alcohol	Bacardi
2	Alcohol	Bacardi, Gold
3	Alcohol	Bacardi, Silver
4	Alcohol	Beer, Ale
5	Alcohol	Beer, Amber

```
6      Alcohol  Beer, Amber Ale
7      Alcohol     Beer, Amstel
8      Alcohol      Beer, Black
9      Alcohol     Beer, Blonde
```

Pre-processing

Split Flavor column to Flovor and its type

```
df_flavor_classification[['Flavor (Calculated)','Flavor type
(Calculated)']] =
df_flavor_classification['Flavor'].str.split(',',n=1, expand=True)

df_flavor_classification.head(10)
```

```
  Flavor_Group           Flavor Flavor (Calculated) Flavor type
(Calculated)
0      Alcohol    Vodka, Citron               Vodka
Citron
1      Alcohol          Bacardi             Bacardi
None
2      Alcohol    Bacardi, Gold             Bacardi
Gold
3      Alcohol  Bacardi, Silver             Bacardi
Silver
4      Alcohol        Beer, Ale                Beer
Ale
5      Alcohol      Beer, Amber                Beer
Amber
6      Alcohol  Beer, Amber Ale                Beer
Amber Ale
7      Alcohol     Beer, Amstel                Beer
Amstel
8      Alcohol      Beer, Black                Beer
Black
9      Alcohol     Beer, Blonde                Beer
Blonde
```

```
df_flavor_flavor_group =
df_flavor_classification[['Flavor_Group','Flavor
(Calculated)']].groupby("Flavor (Calculated)",as_index = False).max()

df_flavor_flavor_group["Flavor (Calculated)"] =
df_flavor_flavor_group["Flavor (Calculated)"].apply( lambda x:
(str(x).lower()))

df_flavor_flavor_group
```

```
    Flavor (Calculated)   Flavor_Group
0                  abiu          Fruit
1                acacia          Herbs
2                  akee          Fruit
3             al pastor         Ethnic
```

```
4               alfalfa      Vegetable
..                  ...            ...
629              yogurt  Dairy Flavors
630                yuca      Vegetable
631               yucca      Vegetable
632             zambuca        Alcohol
633            zucchini      Vegetable

[634 rows x 2 columns]
```

Basic Stats:

```
print("Total data ")
print("-"*50)
print("No of unique Flavors
Groups:" ,df_flavor_classification["Flavor_Group"].nunique())
print("No of unique Flavors
available:" ,df_flavor_classification["Flavor
(Calculated)"].nunique())
print("No of unique Flavors Types
available:" ,df_flavor_classification["Flavor type
(Calculated)"].nunique())
```

```
Total data
--------------------------------------------------
No of unique Flavors Groups: 22
No of unique Flavors available: 634
No of unique Flavors Types available: 632
```

Data Quality Check:

- Check for duplicates/Nulls: No point of checking duplicates as flavors and flavor type will definitely have duplicated and Nones

Analysis:

```
print("No of Flavors of each Flavor Group")
df_flavorGroups = df_flavor_classification[["Flavor_Group", "Flavor
(Calculated)"]].groupby("Flavor_Group",as_index = False).agg({"Flavor
(Calculated)": "nunique"})
df_flavorGroups[["Flavor_Group", "No of Flavors"]]= df_flavorGroups
df_flavorGroups = df_flavorGroups.drop(columns = "Flavor
(Calculated)")
df_flavorGroups.sort_values(by = "No of Flavors", ascending =
False,inplace = True)
print(df_flavorGroups.columns)
df_flavorGroups
```

```
No of Flavors of each Flavor Group
Index(['Flavor_Group', 'No of Flavors'], dtype='object')
```

```
          Flavor_Group  No of Flavors
7                Fruit            161
21           Vegetable            123
```

```
17       Spices & Seeds             52
4                Ethnic             47
1         Brown Flavors             42
9                 Herbs             39
13  Poultry, meat, fish             38
0               Alcohol             31
15    Sauce & Condiment             21
3         Dairy Flavors             19
6                Flowers             17
5       Fantasy Flavors             15
2    Cake, cookie & pie             12
8                 Grains              8
10       Mint & Menthol              6
16      Smoke & Roasted              5
12        Oil & Vinegar              2
11                 Nuts              2
14                 Rice              1
18                  Tea              1
19           Unflavored              1
20              Vanilla              1
```

Merging Datasets:

Merging Product Launch and Positioning Mapping Datasets

```
preprocessed_productLaunch.shape
```

```
(505612, 12)
```

```
df_merge_launchAndPositionMapping =
preprocessed_productLaunch.merge(df_positonal_category_mapping,how =
'left',left_on = 'Positioning Subcategory (Calculated)', right_on =
'Positioning Subcategory')
df_merge_launchAndPositionMapping.head(5)
```

```
   Product id                                      Flavor Market
Subcategory  \
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
1           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
2           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
3           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
4           2                       Fruit, not specified
Carbonates

  Launch Date Country       Region                Positioning  \
0  2001-01-01     WE1  West Europe       Low Calorie, Natural
```

```
1  2001-01-01     WE1  West Europe       Low Calorie, Natural
2  2001-01-01     WE1  West Europe       Low Calorie, Natural
3  2001-01-01     WE1  West Europe       Low Calorie, Natural
4  2001-01-01     WE2  West Europe  Convenience - Consumption

  Flavor (Calculated) Flavor type (Calculated)  \
0               herbs            not specified
1               herbs            not specified
2               fruit            not specified
3               fruit            not specified
4               fruit            not specified

  Positioning Subcategory (Calculated) Month (Calculated)  Year
(Calculated)  \
0                           Low Calorie            January
2001
1                               Natural            January
2001
2                           Low Calorie            January
2001
3                               Natural            January
2001
4              Convenience - Consumption            January
2001

  Positioning Group    Positioning Subcategory
0  Health (Passive)                 Low Calorie
1  Health (Passive)                     Natural
2  Health (Passive)                 Low Calorie
3  Health (Passive)                     Natural
4       Convenience  Convenience - Consumption
```

```
df_merge_launchAndPositionMapping[df_merge_launchAndPositionMapping['P
ositioning Subcategory'] == " "]
```

```
Empty DataFrame
Columns: [Product id, Flavor, Market Subcategory, Launch Date,
Country, Region, Positioning, Flavor (Calculated), Flavor type
(Calculated), Positioning Subcategory (Calculated), Month
(Calculated), Year (Calculated), Positioning Group, Positioning
Subcategory]
Index: []
```

- No nulls found. Which means merging successfull and we have successfully added Positioning Group info

```
df_merge_launchAndPositionMapping.shape
```

```
(505612, 14)
```

Merging Flavor Classification Datasets

```
df_final_preprocessed_data =
df_merge_launchAndPositionMapping.merge(df_flavor_flavor_group,how =
'left',left_on = 'Flavor (Calculated)', right_on = 'Flavor
(Calculated)')
df_final_preprocessed_data.head(5)
```

```
   Product id                                    Flavor Market
Subcategory  \
0           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
1           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
2           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
3           1  Herbs, not specified; Fruit, not specified  Other Soft
Drinks
4           2                       Fruit, not specified
Carbonates

  Launch Date Country       Region                Positioning  \
0  2001-01-01     WE1  West Europe       Low Calorie, Natural
1  2001-01-01     WE1  West Europe       Low Calorie, Natural
2  2001-01-01     WE1  West Europe       Low Calorie, Natural
3  2001-01-01     WE1  West Europe       Low Calorie, Natural
4  2001-01-01     WE2  West Europe  Convenience - Consumption

  Flavor (Calculated) Flavor type (Calculated)  \
0               herbs            not specified
1               herbs            not specified
2               fruit            not specified
3               fruit            not specified
4               fruit            not specified

  Positioning Subcategory (Calculated) Month (Calculated)  Year
(Calculated)  \
0                          Low Calorie            January
2001
1                              Natural            January
2001
2                          Low Calorie            January
2001
3                              Natural            January
2001
4            Convenience - Consumption            January
2001

  Positioning Group    Positioning Subcategory Flavor_Group
0  Health (Passive)                Low Calorie        Herbs
1  Health (Passive)                    Natural        Herbs
```

```
2  Health (Passive)                Low Calorie        Fruit
3  Health (Passive)                    Natural        Fruit
4       Convenience  Convenience - Consumption        Fruit
```

```
df_final_preprocessed_data[df_final_preprocessed_data['Flavor_Group']
== ""]
```

```
Empty DataFrame
Columns: [Product id, Flavor, Market Subcategory, Launch Date,
Country, Region, Positioning, Flavor (Calculated), Flavor type
(Calculated), Positioning Subcategory (Calculated), Month
(Calculated), Year (Calculated), Positioning Group, Positioning
Subcategory, Flavor_Group]
Index: []
```

Dropping off redundant columns in final preprocessed dataset

```
df_final_preprocessed_data = df_final_preprocessed_data.drop(columns
="Positioning Subcategory (Calculated)")
```

- No nulls found. Which means merging successfull and we have successfully added Flavor Group info

Final Preprocessed Data

```
df_final_preprocessed_data.head(10)
```

```
   Product id                                      Flavor  \
0           1  Herbs, not specified; Fruit, not specified
1           1  Herbs, not specified; Fruit, not specified
2           1  Herbs, not specified; Fruit, not specified
3           1  Herbs, not specified; Fruit, not specified
4           2                        Fruit, not specified
5           3                        Lemon; Honey; Ginger
6           3                        Lemon; Honey; Ginger
7           3                        Lemon; Honey; Ginger
8           3                        Lemon; Honey; Ginger
9           3                        Lemon; Honey; Ginger

     Market Subcategory Launch Date Country       Region  \
0     Other Soft Drinks  2001-01-01     WE1  West Europe
1     Other Soft Drinks  2001-01-01     WE1  West Europe
2     Other Soft Drinks  2001-01-01     WE1  West Europe
3     Other Soft Drinks  2001-01-01     WE1  West Europe
4            Carbonates  2001-01-01     WE2  West Europe
5  Juice & Juice Drinks  2001-01-01     WE2  West Europe
6  Juice & Juice Drinks  2001-01-01     WE2  West Europe
7  Juice & Juice Drinks  2001-01-01     WE2  West Europe
8  Juice & Juice Drinks  2001-01-01     WE2  West Europe
9  Juice & Juice Drinks  2001-01-01     WE2  West Europe
```

```
                                          Positioning Flavor
(Calculated)  \
0                              Low Calorie, Natural
herbs
1                              Low Calorie, Natural
herbs
2                              Low Calorie, Natural
fruit
3                              Low Calorie, Natural
fruit
4                         Convenience - Consumption
fruit
5  100% Not from Concentrate, Convenience - Consu...
lemon
6  100% Not from Concentrate, Convenience - Consu...
lemon
7  100% Not from Concentrate, Convenience - Consu...
lemon
8  100% Not from Concentrate, Convenience - Consu...
honey
9  100% Not from Concentrate, Convenience - Consu...
honey

  Flavor type (Calculated) Month (Calculated)  Year (Calculated)  \
0            not specified            January               2001
1            not specified            January               2001
2            not specified            January               2001
3            not specified            January               2001
4            not specified            January               2001
5                     none            January               2001
6                     none            January               2001
7                     none            January               2001
8                     none            January               2001
9                     none            January               2001

  Positioning Group    Positioning Subcategory   Flavor_Group
0  Health (Passive)                Low Calorie          Herbs
1  Health (Passive)                    Natural          Herbs
2  Health (Passive)                Low Calorie          Fruit
3  Health (Passive)                    Natural          Fruit
4       Convenience  Convenience - Consumption          Fruit
5             Juice  100% Not from Concentrate          Fruit
6       Convenience  Convenience - Consumption          Fruit
7  Health (Passive)                    Natural          Fruit
8             Juice  100% Not from Concentrate  Brown Flavors
9       Convenience  Convenience - Consumption  Brown Flavors
```

```
print("# of rows in final preprocessed data: ",
df_final_preprocessed_data.shape[0])
```

```
print("# of columns in final preprocessed data: ",
df_final_preprocessed_data.shape[1])
```

```
# of rows in final preprocessed data:  505612
# of columns in final preprocessed data:  14
```

```
#Exporting pre-processed data to csv file
#Uncomment below code this if you want to export csv
#df_final_preprocessed_data.to_csv("final_preprocessed_data.csv")
```