T 101: Data Cleaning and Merging

This notebook serves as the solution of first part of **Task 1: Data Preparation and Customer Analytics** of **Quantium Data Analytics Virtual Experience Program.**

The goal of this notebook is to clean the data, generate new variables and merge two dataframes into a single data frame. The whole process will be performed in 2 phases

- **Phase 1:** In the first phase, we will clean the dataset and generating new features from exisiting features. We will be deriving
  - Brand Name
  - Flavour
  - Packet Size
- **Phase 2:** Once we done with phase 1, we will move on to phase 2. In this, we will merge transaction and customer segment dataframe into a single datafrmae. This dataframe will be used in the next notebook for futher analysis

In the next notebook, `T 102: Data Exploration, Analysis and Insights` we will explore and analyse the dataset and finally we will derive intersting and useful insights which could help Category Manager to target specific customer segments in prder to increase the sales,

Importing Libraries

In [115]:
```
#To import the dataset as dataframe
import pandas as pd

#To play sound (typically used as an alarm for long code executions)
from playsound import playsound #playsound("path\Sound.mp3")

#To clear the output screen -> used in defined functions
from IPython.display import clear_output

#To not print warning messages
import warnings
warnings.filterwarnings("ignore")
```

Loading Datatsets

In [36]:
```
transations = pd.read_excel("C:/Users/amitm/Jupyter Notebools/Quantium/Task 1/QVI_transaction_data.xls
x")
behaviour = pd.read_csv("C:/Users/amitm/Jupyter Notebools/Quantium/Task 1/QVI_purchase_behaviour.csv")

playsound("C:/Users/amitm/Jupyter Notebools/Sound.mp3")
```

In [37]:
```
transations.head()
```

Out[37]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8

In [38]:
```
behaviour.head()
```

Out[38]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	YOUNG SINGLES/COUPLES	Premium
1	1002	YOUNG SINGLES/COUPLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES/COUPLES	Mainstream
4	1005	MIDAGE SINGLES/COUPLES	Mainstream

We can observe that datasets are loaded successfully

Phase 1: Cleaning and Generating new Features

In [39]:
```
transations["PROD_NAME"].nunique()
```

Out[39]: 114

In [40]:
```
transations["PROD_NBR"].nunique()
```

Out[40]: 114

Sub Task 1

Derive weight of the chips packets from product name

In [41]:
```
transations["PROD_WT"] = transations["PROD_NAME"].str.slice(-4,-1)
```

We took last three elements (excluding last element) from product name to get the weight.

In [46]:
```
transations[transations["PROD_WT"].str.contains(r"[a-z,A-Z]",regex=True)]
```

Out[46]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PROD_WT
65	2019-05-20	83	83008	82099	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Sal
153	2019-05-17	208	208139	206906	63	Kettle 135g Swt Pot Sea Salt	1	4.2	Sal
174	2018-08-20	237	237227	241132	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Sal
177	2019-05-17	243	243070	246706	63	Kettle 135g Swt Pot Sea Salt	1	4.2	Sal
348	2018-10-26	7	7077	6604	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Sal
...	...	...	...	...	...	...	...	...	...
264564	2018-10-08	260	260240	259480	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Sal
264574	2019-06-12	261	261035	259860	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Sal
264725	2018-07-20	266	266413	264246	63	Kettle 135g Swt Pot Sea Salt	1	4.2	Sal
264767	2019-06-08	269	269133	265839	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Sal
264823	2019-03-17	272	272156	269855	63	Kettle 135g Swt Pot Sea Salt	2	8.4	Sal

3257 rows × 9 columns

In [47]:
```
transations[transations["PROD_WT"].str.contains(r"[a-z,A-Z]",regex=True)]["PROD_NBR"].unique()
```

Out[47]: array([63], dtype=int64)

We can see that for product number 63, weight was given in between and not at the end

Therefore, we need to update the correct weight corrosponding to product number 63

In [95]:
```
transations.loc[transations["PROD_NBR"]==63,"PROD_WT"] = transations[transations["PROD_NBR"]==63]["PROD
_NAME"].str.slice(7,10)
```

In [99]:
```
transations["PROD_WT"] = transations["PROD_WT"].astype(int)
```

Weight of the chips packets are derived Successfully

Sub Task 2

Derive Brand Name of the chips packets from product name

In [103]:
```
transations["BRAND_NAME"] = transations["PROD_NAME"].str.split(pat=" ",n=1,expand=True)[0]
```

In [104]:
```
transations.head()
```

Out[104]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PROD_WT	BRAND_NAME
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175	Natural
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	175	CCs
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	170	Smiths
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	175	Smiths
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	150	Kettle

In [111]:
```
brands = transations["BRAND_NAME"].unique()

brands
```

Out[111]:
```
array(['Natural', 'CCs', 'Smiths', 'Kettle', 'Old', 'Grain', 'Doritos',
       'Twisties', 'WW', 'Thins', 'Burger', 'NCC', 'Cheezels', 'Infzns',
       'Red', 'Pringles', 'Dorito', 'Infuzions', 'Smith', 'GrnWves',
       'Tyrrells', 'Cobs', 'Woolworths', 'French', 'RRD', 'Tostitos',
       'Cheetos', 'Snbts', 'Sunbites'], dtype=object)
```

We can observe that brands name were derived however there are some consistency issues in brand name

Issues:

- Some brand names are spelt incorrectly
- Brand name spelt differently for the same brand like Dorito and Doritos
- Abbreviation are used for some brand names
- Full names of the brands are not capurted

We need to fix all of the above issues and update the brand names correctly

In [132]:
```
def brands_check(data,brands):
    """
    This functions takes two arguements data frame and list of all brand names

    It runs a loop on brand names and asks if changes are required
    1: Changes required
    0: Changes not required
    e: Come out of the function

    When pressed 1, it asks for new name and user can input the correct brand name

    It returns the whole data frame with correct brand names stored in a new column
    """

    df = data.copy()

    for b in brands:
        dff = df[df["BRAND_NAME"] == b][["PROD_NAME","BRAND_NAME"]]

        clear_output()
        print(dff)

        inp = input("Need Correction? (0/1): ")
        if inp == "e":
            return
        elif inp == "1":
            df.loc[df["BRAND_NAME"]==b,"Updated"] = 1
            name = input("Enter new name: ")
            df.loc[df["BRAND_NAME"]==b,"BRAND_NAME"] = name
        else:
            df.loc[df["BRAND_NAME"]==b,"Updated"] = 0

    return df
```

In [138]:
```
df = brands_check(transations,brands)
```

```
                                      PROD_NAME BRAND_NAME
224         Snbts Whlgrn Crisps Cheddr&Mstrd 90g   Sunbites
333      Sunbites Whlegrn    Crisps Frch/Onin 90g   Sunbites
414      Sunbites Whlegrn    Crisps Frch/Onin 90g   Sunbites
493         Snbts Whlgrn Crisps Cheddr&Mstrd 90g   Sunbites
525         Snbts Whlgrn Crisps Cheddr&Mstrd 90g   Sunbites
...                                          ...        ...
264751      Snbts Whlgrn Crisps Cheddr&Mstrd 90g   Sunbites
264786      Snbts Whlgrn Crisps Cheddr&Mstrd 90g   Sunbites
264791   Sunbites Whlegrn    Crisps Frch/Onin 90g   Sunbites
264802   Sunbites Whlegrn    Crisps Frch/Onin 90g   Sunbites
264817   Sunbites Whlegrn    Crisps Frch/Onin 90g   Sunbites

[3008 rows x 2 columns]
Need Correction? (0/1): 1
Enter new name: Sunbites
```

In [141]:
```
df.head()
```

Out[141]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PROD_WT	BRAND_NAME	Up
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175	Naturals	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	175	Corn Chips	
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	170	Smiths	
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	175	Smiths	
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	150	Kettle	

In [142]:
```
df["BRAND_NAME"].value_counts()
```

Out[142]:
```
Kettle            41288
Smiths            31823
Doritos           28147
Pringles          25102
Red Rock Deli     17779
Woolworths        14757
Infuzions         14201
Thins             14075
Cobs               9693
Tostitos           9471
Twisties           9454
Old El Paso        9324
Grain Waves        7740
Naturals           7469
Tyrrells           6442
Cheezels           4603
Corn Chips         4551
Sunbites           3008
Cheetos            2927
Burger Rings       1564
French Fries       1418
Name: BRAND_NAME, dtype: int64
```

Brand Name of the chips are derived Successfully

Sub Task 3

Derive Flavour of the chips packets from product name

In [144]:
```
transations = df.copy()
```

In [154]:
```
def flavour_check(data,flavours):
    """
    This function takes two arguments, data frame and list of all flavours (unique product name)

    It run a for loop on flavours and shows product name column from the dataframe

    It then ask the user name of the flavour and user can enter it

    Finally it returns the whole dataframe with the correct flavour names stored in a new column

    """

    df = data.copy()

    for f in flavours:
        dff = df[df["PROD_NAME"] == f]["PROD_NAME"]

        clear_output()
        print(dff)

        inp = input("Enter Flavour: ")
        if inp == "e":
            return df
        else:
            df.loc[df["PROD_NAME"]==f,"Flavour"] = inp

    return df
```

In [155]:
```
flav = transations["PROD_NAME"].unique()

df = flavour_check(transations,flav)
```

```
689       Doritos Salsa Mild  300g
913       Doritos Salsa Mild  300g
1459      Doritos Salsa Mild  300g
1474      Doritos Salsa Mild  300g
1581      Doritos Salsa Mild  300g
                    ...
264343    Doritos Salsa Mild  300g
264346    Doritos Salsa Mild  300g
264528    Doritos Salsa Mild  300g
264655    Doritos Salsa Mild  300g
264734    Doritos Salsa Mild  300g
Name: PROD_NAME, Length: 1472, dtype: object
Enter Flavour: Salsa Mild
```

In [156]:
```
df.head()
```

Out[156]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PROD_WT	BRAND_NAME	Up
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175	Naturals	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	175	Corn Chips	
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	170	Smiths	
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	175	Smiths	
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	150	Kettle	

In [158]:
```
df.drop("Updated",axis=1,inplace=True)
```

In [159]:
```
df.head()
```

Out[159]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PROD_WT	BRAND_NAME	Fl
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175	Naturals	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	175	Corn Chips	N Cl
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	170	Smiths	Ch
3	2018-08-17	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0	175	Smiths	C &
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	150	Kettle	C H

Flavour of the chips packets are derived Successfully

Phase 2: Merging Dataframes

Finally we are merging the dataframe we got above with customer behaviour dataframe and create a single dataframe

In [165]:
```
#merging df and behaviour dataset on LYLTY_CARD_NBR and making single dataset

data = df.merge(behaviour,on="LYLTY_CARD_NBR")
```

In [166]:
```
data.head()
```

Out[166]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PROD_WT	BRAND_NAME	Flav
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175	Naturals	
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	175	Corn Chips	Na Che
2	2018-11-10	1	1307	346	96	WW Original Stacked Chips 160g	2	3.8	160	Woolworths	Orig Stac
3	2019-03-09	1	1307	347	54	CCs Original 175g	1	2.1	175	Corn Chips	Orig
4	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	170	Smiths	Chic

In [167]:
```
data.shape
```

Out[167]: (264836, 13)

In [168]:
```
df.shape
```

Out[168]: (264836, 11)

In [170]:
```
df.isnull().sum()
```

Out[170]:
```
DATE              0
STORE_NBR         0
LYLTY_CARD_NBR    0
TXN_ID            0
PROD_NBR          0
PROD_NAME         0
PROD_QTY          0
TOT_SALES         0
PROD_WT           0
BRAND_NAME        0
Flavour           0
dtype: int64
```

We can observe that dataframes are merged successfully and there are no missing values.

Now we can export this final dataframe to a CSV file which will be used further to do anlysis and draw insights.

In [169]:
```
#Export the final dataframe to a CSV file which will used in the next notebook for further analysis

data.to_csv("C:/Users/amitm/Jupyter Notebools/Quantium/Task 1/QVI_merged_data.csv",index=False)
```

Conclusion

Let's summaries what all we have done.

- We corrected the datatypes of the variables
- We derived three new features from product name
  - Brand Name
  - Flavour
  - Packet Size
- Finally we merged both dataframes into a single dataframe and exported it, which will be used for further analysis

First part of Task 1 of Quantium Virtual Internship Program ends here. Second part is in the next notebook `T 102: Data Exploration, Analysis and Insights`