**Importing libraries** In [1]:

**import**

**pandas**

**as**

**pd**

**import**

**matplotlib.pyplot**

**as**

**plt**

**import**

**seaborn**

**as**

**sns**

**import**

**numpy**

**as**

**np**

*# for the Q-Q plots*

*#import scipy.stats as stats*

%

**matplotlib**

inline

**import**

**pandas**

**as**

**pd**

pd

.

options

.

display

.

float\_format

=

'

**{:.2f}**

'

.

format

*#from pandas.io.json import json\_normalize*

**Importing dataset for brands** In [2]:

brands

=

pd

.

read\_excel

(

"brands.xlsx"

)

In [3]:

brands

.

info

()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 1167 entries, 0 to 1166

Data columns (total 9 columns): \_id/$oid 1167 non-null object barcode 1167 non-null int64 category 1012 non-null object categoryCode 517 non-null object cpg/$id/$oid 1167 non-null object cpg/$ref 1167 non-null object name 1167 non-null object topBrand 555 non-null object brandCode 898 non-null object dtypes: int64(1), object(8) memory usage: 82.2+ KB

In [4]:

brands

[

"barcode"

]

=

brands

[

"barcode"

]

.

astype

(

str

)

In [5]:

brands

.

head

()

Out[5]:

**\_id/$oid barcode category categoryCode cpg**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 601ac115be37ce2ead437551 | 511111019862 | Baking | BAKING | 601ac114be37ce2e |
| 1 | 601c5460be37ce2ead43755f | 511111519928 | Beverages | BEVERAGES | 5332f5fbe4b03c9a |
| 2 | 601ac142be37ce2ead43755d | 511111819905 | Baking | BAKING | 601ac142be37ce2e |
| 3 | 601ac142be37ce2ead43755a | 511111519874 | Baking | BAKING | 601ac142be37ce2e |
| 4 | 601ac142be37ce2ead43755e | 511111319917 | Candy &  Sweets | CANDY\_AND\_SWEETS | 5332fa12e4b03c9a |

**Quantifying missing data** In [6]:

brands

.

isnull

()

.

sum

()

Out[6]:

\_id/$oid 0 barcode 0 category 155 categoryCode 650 cpg/$id/$oid 0 cpg/$ref 0 name 0 topBrand 612 brandCode 269 dtype: int64

***percentage of missing values in variables***

In [7]:

*# alternatively, we can use the mean() method after isnull() to visualise the percentage o f missing values for each variable*

percentage\_null\_values = brands.isnull().mean() **for** key,value **in** percentage\_null\_values.items(): **if** value >0: print(key,":",value\*100)

category : 13.281919451585262 categoryCode : 55.69837189374465 topBrand : 52.44215938303341 brandCode : 23.050556983718938

A considerate fraction of values(more than 50%) are missing from topBrand and categoryCode variables.

**Checking for redundant records**

In [8]:

duplicateRowsDF = brands[brands.duplicated()]

print("Duplicate Rows except first occurrence based on all columns are :") print(duplicateRowsDF)

Duplicate Rows except first occurrence based on all columns are :

Empty DataFrame

Columns: [\_id/$oid, barcode, category, categoryCode, cpg/$id/$oid, cpg/$ref, name, topBrand, brandCode] Index: []

No duplicate records found.

**Examining values of categorical varibales**

Here, the variable of my interest is brand 'category'.

In [10]:

brands

[

"category"

]

.

unique

()

Out[10]:

array(['Baking', 'Beverages', 'Candy & Sweets', 'Condiments & Sauces',

'Canned Goods & Soups', nan, 'Magazines', 'Breakfast & Cereal',

'Beer Wine Spirits', 'Health & Wellness', 'Beauty', 'Baby',

'Frozen', 'Grocery', 'Snacks', 'Household', 'Personal Care',

'Dairy', 'Cleaning & Home Improvement', 'Deli',

'Beauty & Personal Care', 'Bread & Bakery', 'Outdoor',

'Dairy & Refrigerated'], dtype=object)

**Examining percentage of different category values for categorical variables**

Here, the categorical variable of my interest is category,

In [11]:

freq\_category = 100\*(brands['category'].value\_counts() / len(brands)) print(freq\_category.map('**{:,.2f}** %'.format))

Baking 31.62 %

Beer Wine Spirits 7.71 %

Snacks 6.43 %

Candy & Sweets 6.08 %

Beverages 5.40 %

Health & Wellness 3.77 %

Magazines 3.77 %

Breakfast & Cereal 3.43 %

Grocery 3.34 %

Dairy 2.83 %

Condiments & Sauces 2.31 %

Frozen 2.06 %

Personal Care 1.71 %

Baby 1.54 %

Canned Goods & Soups 1.03 %

Beauty 0.77 %

Cleaning & Home Improvement 0.51 %

Deli 0.51 %

Beauty & Personal Care 0.51 %

Bread & Bakery 0.43 %

Dairy & Refrigerated 0.43 %

Household 0.43 %

Outdoor 0.09 % Name: category, dtype: object

Majority of brands belong to the 'Baking' category

No data quality issues found except large number of missing values in topBrand and categoryCode columns.