Exploratory Data Analysis of Big Data Set

```
# Importing important Libraries

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Importing dataset from google drive

from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive
```

```
Mounted at /gdrive
/gdrive
```

```
url = "/gdrive/MyDrive/Ahmedapps/en.openfoodfacts.org.products.tsv"
food = pd.read_csv(url,sep='\t')
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2882: DtypeWarni
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Step 1 of EDA analysis

```
food.head()
```

	code	url	creator	created_t	created_datetime	la
0	3087	http://world-en.openfoodfacts.org/product/0000...	openfoodfacts-contributors	1474103866	2016-09-17T09:17:46Z	
1	4530	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489069957	2017-03-09T14:32:37Z	

```
food.shape
```

```
(356027, 163)
```

3	16087	en.openfoodfacts.org/product/0000...	import	[illegible]	09T10:35:31Z	

```
rows, cols = food.shape
print("numbers of rows is :", rows) # instances
print("numbers of cols is :", cols) # series
```

```
numbers of rows is : 356027
numbers of cols is : 163
```

```
food.head(5)
```

	code	url	creator	created_t	created_datetime	la
0	3087	http://world-en.openfoodfacts.org/product/0000...	openfoodfacts-contributors	1474103866	2016-09-17T09:17:46Z	
1	4530	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489069957	2017-03-09T14:32:37Z	
2	4559	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489069957	2017-03-09T14:32:37Z	
3	16087	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489055731	2017-03-09T10:35:31Z	
4	16094	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489055653	2017-03-09T10:34:13Z	

5 rows × 163 columns

Step 2 of EDA Analysis

```
food.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 356027 entries, 0 to 356026
Columns: 163 entries, code to water-hardness_100g
dtypes: float64(107), object(56)
memory usage: 442.8+ MB
```

Step 3 of EDA analysis

find missing values

```
food.isnull()
```

	code	url	creator	created_t	created_datetime	last_modified_t	last_modifi
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
...	...	...	...	...	...	...	
356022	False	False	False	False	False	False	
356023	False	False	False	False	False	False	
356024	False	False	False	False	False	False	
356025	False	False	False	False	False	False	
356026	False	False	False	False	False	False	

356027 rows × 163 columns

```
food.isnull().sum()
```

```
code                          26
```

```
url                            26
creator                         3
created_t                       3
created_datetime               10
                              ...
carbon-footprint_100g      355749
nutrition-score-fr_100g    101171
nutrition-score-uk_100g    101171
glycemic-index_100g        356027
water-hardness_100g        356027
Length: 163, dtype: int64
```

```
# percent calculation of missing values
food.isnull().sum() /food.shape[0]*100
```

```
code                          0.007303
url                           0.007303
creator                       0.000843
created_t                     0.000843
created_datetime              0.002809
                               ...
carbon-footprint_100g        99.921916
nutrition-score-fr_100g      28.416665
nutrition-score-uk_100g      28.416665
glycemic-index_100g         100.000000
water-hardness_100g         100.000000
Length: 163, dtype: float64
```

Step 4 of EDA Analysis

split variables for new columns needed

```
food[['url type','second_part']] = food['url'].str.split(':', expand = True)
```

```
food.head()
```

	code	url	creator	created_t	created_datetime	la
0	3087	http://world-en.openfoodfacts.org/product/0000...	openfoodfacts-contributors	1474103866	2016-09-17T09:17:46Z	
1	4530	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489069957	2017-03-09T14:32:37Z	
2	4559	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489069957	2017-03-09T14:32:37Z	

Step 5 of EDA Analysis

4	16094	http://world-	usda-ndb-	1489055653	2017-03-

typecasting / conversion of datatype

```
food.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 356027 entries, 0 to 356026
Columns: 165 entries, code to second_part
dtypes: float64(107), object(58)
memory usage: 448.2+ MB
```

```
# to convert it into integers
```

```
food['nutrition-score-fr_100g'].dropna()
```

```
1         14.0
2          0.0
3         12.0
7          7.0
12        12.0
          ...
355982    17.0
355985    -1.0
356005    -4.0
356010     0.0
356022     0.0
Name: nutrition-score-fr_100g, Length: 254856, dtype: float64
```

```
food.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 356027 entries, 0 to 356026
```

```
Columns: 165 entries, code to second_part
dtypes: float64(107), object(58)
memory usage: 448.2+ MB
```

```
food[['nutrition-score-fr_100g']] = food[['nutrition-score-fr_100g']].astype('str')
food.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 356027 entries, 0 to 356026
Columns: 165 entries, code to second_part
dtypes: float64(106), object(59)
memory usage: 448.2+ MB
```

Step 6 of EDA Analysis

```
# Summary statistics of data
```

```
food.describe()
```

	no_nutriments	additives_n	ingredients_from_palm_oil_n	ingredients_from_palm
count	0.0	283867.000000	283867.000000	
mean	NaN	1.876851	0.023430	
std	NaN	2.501022	0.153094	
min	NaN	0.000000	0.000000	
25%	NaN	0.000000	0.000000	
50%	NaN	1.000000	0.000000	
75%	NaN	3.000000	0.000000	
max	NaN	30.000000	2.000000	

8 rows × 106 columns

Step 6 of EDA Analysis

```
# Value count of a specific columns


food['creator'].value_counts()
```

```
usda-ndb-import               169868
openfoodfacts-contributors     45805
kiliweb                        36379
date-limite-app                12679
openfood-ch-import             11469
                                ...
leleio                             1
bora                               1
sevede28                           1
brunoa                             1
climboxing                         1
Name: creator, Length: 3890, dtype: int64
```

```
# Finding unique value in a dataset
food['creator'].unique()
```

```
array(['openfoodfacts-contributors', 'usda-ndb-import', 'chris13', ...,
       'robopetr', 'mmarquesma', 'jerem26260'], dtype=object)
```

Step 8 of EDA Analysis

```
# deals with duplicate and/or Null values (meqn, median, mode or other methods)
food[food.creator =='usda-ndb-import']
```

	code	url	creator	created_t	created_dat
1	4530	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489069957	20 09T14:3
2	4559	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489069957	20 09T14:3
3	16087	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489055731	20 09T10:3
4	16094	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489055653	20 09T10:3
5	16100	http://world-en.openfoodfacts.org/product/0000...	usda-ndb-import	1489055651	20 09T10:3
...	...	...	...	...	
355968	9780803738782	http://world-en.openfoodfacts.org/product/9780...	usda-ndb-import	1489069944	20 09T14:3

```
print("the datatypes in our dataset are ", food.dtypes)
```

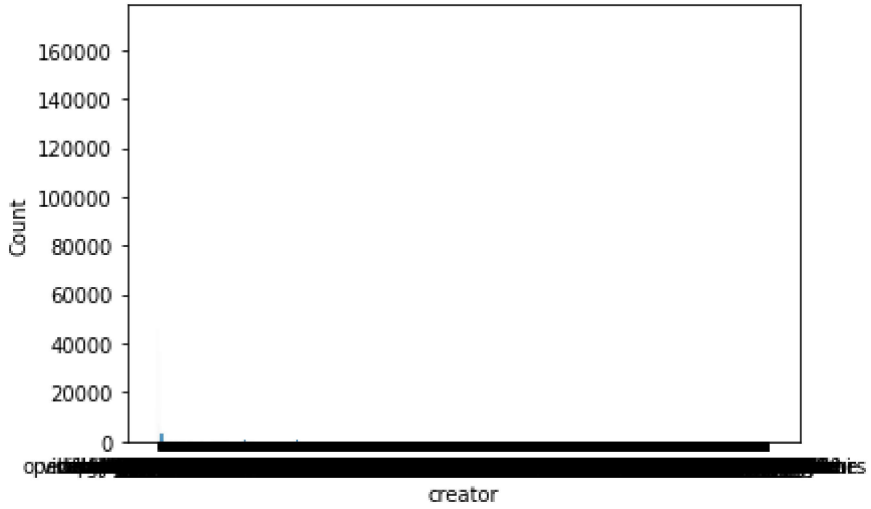
```
the datatypes in our dataset are  code                      object
url                          object
creator                      object
created_t                    object
created_datetime             object
                             ...
nutrition-score-uk_100g     float64
glycemic-index_100g         float64
water-hardness_100g         float64
url type                     object
second_part                  object
Length: 165, dtype: object
```

Step 9 of EDA Analysis

```
# check the normality and standard normal distribution


sns.histplot(food['creator'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f76ce1c10>
```




```
#sns.boxplot(food['creator'], color = 'red')
```

Step 10 of EDA Analysis

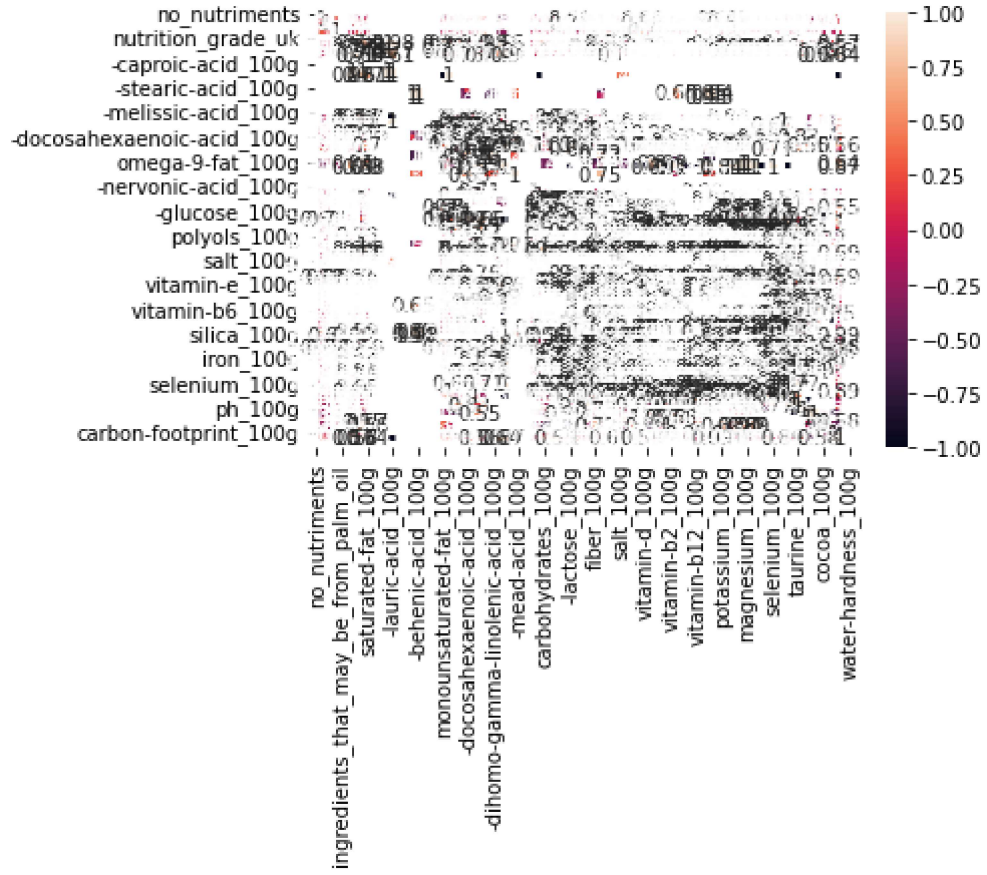
```
# Correlations

corr = food.corr(method = 'pearson')
corr # this will display a correlation matrix
```

	no_nutriments	additives_n	ingredients_from_pa
no_nutriments	NaN	NaN	

```
sns.heatmap(corr, annot= True)
#this will show numbers with colors
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f75e26950>
```



22s completed at 6:16 PM

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.