clean_spell

November 2, 2021

1 Data Intern Challenge

1.0.1 Written by Okiri Albert Cleaning Text Columns Based on Closeness of Spelling

[1]:
```python
import pandas as pd
import difflib
```

1.0.2 Load excel spreadsheet

[2]:
```python
df = pd.read_excel('copy_data_intern_challenge.xlsx', sheet_name="data")
df[:10]
```

[2]:
```
                 api_names       brand_sold_product    avg_price   count
0  Artemether / Lumefantrine                      NaN  113.636364    11.0
1  Artemether / Lumefantrine                       6T  200.000000     3.0
2  Artemether / Lumefantrine                       96    0.000000     3.0
3  Artemether / Lumefantrine                      ACT   67.000000    30.0
4  Artemether / Lumefantrine                   ACT AL   50.000000    66.0
5  Artemether / Lumefantrine                     ACTM   78.174557  2122.0
6  Artemether / Lumefantrine                   AJANTA  124.545455    77.0
7  Artemether / Lumefantrine                AJANTA AL   60.000000     7.0
8  Artemether / Lumefantrine              AJANTA AL S   84.444444    18.0
9  Artemether / Lumefantrine  AJANTA _VOUTURE PROGRAM   50.000000     1.0
```

[3]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 141 entries, 0 to 140
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   api_names           141 non-null    object
 1   brand_sold_product  140 non-null    object
 2   avg_price           141 non-null    float64
 3   count               141 non-null    float64
dtypes: float64(2), object(2)
memory usage: 4.5+ KB
```

1.1 List of correct brand_sold_products

```
[4]: brand_sold_product = df['brand_sold_product']
```

```
[5]: clean_list = {'UNNAMED','6T','96','ACT','ACTM','AJANTA','AJANTA VOUTURE␣
      ↪PROGRAM','A-L ',
                  'AL 25KG','ALFANTRINE','ALL BRANDS','LUFENART','LUMERAX DT','AL␣
      ↪ORIGINAL','PRO FANTRINE FORTE',
                  'AL S ','ALS FALCIZE','AL SYUP','ANTIMALARIAL','ARTEFAN','ARTEFAN␣
      ↪DISPERSIBLE','ARTEFENTRINE',
                  'ARTELAT','ARTEMETHER␣
      ↪LUMEFANTRINE','ARTEMETHER','BLISS','CARTER','CIPLA','COARINET','COARTEM',
                  'COARTESIANE','CO CORITHER','CO MALATHER','CO_MAX','COMAX␣
      ↪DPS','COMBIART','CROWN','DISPERSABLE',
                  'DOSE','ERITHER','FALCIZED','GAME','GENERIC''GENERIC APICA␣
      ↪LTD','GENERIC IPICA LABORATORIES',
                  'GENERIC S','GVITHER','IPCA','L␣
      ↪ARTEM','LEMERAX','LONART','LONEX','LONART DUSPERSIBLE','LORNAT',
                  'LOYALTY','LUFANATE','LUFENART','LUFENART','LUMARTEM','LUMARTEM␣
      ↪DT','LUMARTEM FORTE','LUMATEM',
                  'LUMATERM','LUMATERM DT','LUMEFAC','LUMERAX','LUMERAX␣
      ↪DT','LUMERAX DT S','LUMESOFT PLUS','LUMET',
                  'LUMET ADULTS','LUMETHER DRY','LUMET␣
      ↪S','LUMIART','LUMITER','LUMITER DT','MACLEODS','MALARATE',
                  'MALAREM','MALBETA','MALODAR','NOVARTIS','NOVATIS','ORANGE␣
      ↪FLAVOUR','P ALAXIN','PRO FANTRINE FORTE',
                  'SHAL ARTEM','SHAL ARTEM FORTE','SHAL ARTEM␣
      ↪TEMRIN','TEMTRIN','VETENARY AL GLOVES','WINART',
                  'WINART FORTE'}
```

1.1.1 Fill in Missing Values in brand_sold_product Column

```
[6]: df['brand_sold_product'] = df['brand_sold_product'].fillna('UNNAMED')
```

```
[7]: df.head()
```

```
[7]:                    api_names brand_sold_product   avg_price  count
     0  Artemether / Lumefantrine            UNNAMED  113.636364   11.0
     1  Artemether / Lumefantrine                 6T  200.000000    3.0
     2  Artemether / Lumefantrine                 96    0.000000    3.0
     3  Artemether / Lumefantrine                ACT   67.000000   30.0
     4  Artemether / Lumefantrine             ACT AL   50.000000   66.0
```

1.1.2 Spelling Function to Match With Closeness of Spelling

```
[8]: # Spelling function
     def clean_spell(df):
         return difflib.get_close_matches(df, clean_list, n=4, cutoff=0.48)[0]
```

```
[9]: df['corrected_brand_product'] = df['brand_sold_product'].apply(clean_spell)
```

```
[10]: df[:10]
```

```
[10]:                  api_names       brand_sold_product   avg_price   count  \
      0  Artemether / Lumefantrine                  UNNAMED  113.636364    11.0
      1  Artemether / Lumefantrine                       6T  200.000000     3.0
      2  Artemether / Lumefantrine                       96    0.000000     3.0
      3  Artemether / Lumefantrine                      ACT   67.000000    30.0
      4  Artemether / Lumefantrine                   ACT AL   50.000000    66.0
      5  Artemether / Lumefantrine                     ACTM   78.174557  2122.0
      6  Artemether / Lumefantrine                   AJANTA  124.545455    77.0
      7  Artemether / Lumefantrine                AJANTA AL   60.000000     7.0
      8  Artemether / Lumefantrine              AJANTA AL S   84.444444    18.0
      9  Artemether / Lumefantrine  AJANTA _VOUTURE PROGRAM   50.000000     1.0

        corrected_brand_product
      0                 UNNAMED
      1                      6T
      2                      96
      3                     ACT
      4                     ACT
      5                    ACTM
      6                  AJANTA
      7                  AJANTA
      8                  AJANTA
      9  AJANTA VOUTURE PROGRAM
```

1.1.3 Get Similarity Score

```
[11]: # Define the function that Scores   the spelling:
      def spell_diff(row):
          return difflib.SequenceMatcher(None, row['brand_sold_product'],␣
       ↪row['corrected_brand_product']).ratio()
```

```
[12]: df['score'] = df.apply(spell_diff, axis=1)
```

```
[13]: df[:20]
```

```
[13]:                   api_names       brand_sold_product   avg_price   count  \
      0   Artemether / Lumefantrine                  UNNAMED  113.636364    11.0
```

```
1   Artemether / Lumefantrine                       6T  200.000000     3.0
2   Artemether / Lumefantrine                       96    0.000000     3.0
3   Artemether / Lumefantrine                      ACT   67.000000    30.0
4   Artemether / Lumefantrine                   ACT AL   50.000000    66.0
5   Artemether / Lumefantrine                     ACTM   78.174557  2122.0
6   Artemether / Lumefantrine                   AJANTA  124.545455    77.0
7   Artemether / Lumefantrine                AJANTA AL   60.000000     7.0
8   Artemether / Lumefantrine              AJANTA AL S   84.444444    18.0
9   Artemether / Lumefantrine  AJANTA _VOUTURE PROGRAM   50.000000     1.0
10  Artemether / Lumefantrine   AJANTA VOUTURE PROGRAM   73.043478    23.0
11  Artemether / Lumefantrine                       AL   32.248713  9960.0
12  Artemether / Lumefantrine                      A L   35.810635   126.0
13  Artemether / Lumefantrine                 AL 25KG   50.410959    73.0
14  Artemether / Lumefantrine                     AL 3   36.551724    58.0
15  Artemether / Lumefantrine                     AL 6   47.142857    35.0
16  Artemether / Lumefantrine                   A L AL   96.666667     3.0
17  Artemether / Lumefantrine                    AL AL  100.000000     3.0
18  Artemether / Lumefantrine           AL DISPERSABLE   55.000000     2.0
19  Artemether / Lumefantrine               ALFANTRINE  252.371795    78.0

  corrected_brand_product     score
0                 UNNAMED  1.000000
1                      6T  1.000000
2                      96  1.000000
3                     ACT  1.000000
4                     ACT  0.666667
5                    ACTM  1.000000
6                  AJANTA  1.000000
7                  AJANTA  0.800000
8                  AJANTA  0.705882
9  AJANTA VOUTURE PROGRAM  0.977778
10 AJANTA VOUTURE PROGRAM  1.000000
11                   A-L   0.666667
12                   A-L   0.571429
13                AL 25KG  1.000000
14                   A-L   0.750000
15                   A-L   0.750000
16                   A-L   0.600000
17                   A-L   0.666667
18            DISPERSABLE  0.880000
19             ALFANTRINE  1.000000
```

1.1.4 Get Total Amount from Average Price and Counts of Brands Sold

[14]:
```
df['total_amount'] = df['avg_price'] * df['count']
```

[15]:
```
df_sum = pd.DataFrame()
```

1.1.5 Group the corrected_brand_name_product Column Entries and Sum their total_amount and count

[16]:
```
df_sum['total_revenue_brand'] = df.groupby('corrected_brand_product').
↪total_amount.apply(lambda g: g.sum())
```

[17]:
```
df_sum['total_count_brand'] = df.groupby('corrected_brand_product')['count'].
↪apply(lambda g: g.sum())
```

1.1.6 Get the Average Price

[18]:
```
df_sum['average_price'] = df_sum['total_revenue_brand'] /␣
↪df_sum['total_count_brand']
```

1.1.7 Round the Decimal Values for Currency (2) and Total counts

[19]:
```
df_sum = df_sum.round({'average_price': 2, 'total_count_brand': 0,␣
↪'total_revenue_brand': 2})
```

[20]:
```
df_sum = df_sum.reset_index()
```

[21]:
```
df_sum
```

[21]:
```
   corrected_brand_product  total_revenue_brand  total_count_brand  \
0                       6T               600.00                3.0
1                       96                 0.00                3.0
2                      A-L            330069.32            10185.0
3                      ACT              5310.00               96.0
4                     ACTM            165886.41             2122.0
..                     …                    …                  …
83                 TEMTRIN               770.04               52.0
84                 UNNAMED              1250.00               11.0
85      VETENARY AL GLOVES                59.00                4.0
86                  WINART              2280.00               19.0
87            WINART FORTE             31715.52              217.0

    average_price
0          200.00
1            0.00
2           32.41
3           55.31
4           78.17
..            …
83          14.81
84         113.64
85          14.75
86         120.00
```

```
87          146.15

[88 rows x 4 columns]
```

1.1.8 Write New Excel File for Submission

```
[22]: df_sum.to_excel("cleaned.xlsx",  index=False, sheet_name='data')
```

```
[ ]:
```