FinalProject_MS3_KanaparthiVenkata

March 3, 2022

[11]:
```
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
```

[4]:
```
url='https://en.wikipedia.org/wiki/List_of_breweries_in_Maryland'
dfs = pd.read_html(url)
df = dfs[1]
df
```

[4]:

	Brewery	Location
0	1623 Brewing Company	Eldersburg
1	1812 Brewery	Cumberland
2	7 Locks Brewing	Rockville
3	AleCraft Brewery	Bel Air
4	Antietam Brewery	Hagerstown
..	…	…
100	Waredaca Brewing Co.	Gaithersburg
101	Waverly Brewing Co.	Baltimore (Hampden)
102	Wet City Brewing	Baltimore (Mount Vernon)
103	White Marsh Brewing Co.	White Marsh
104	Wild Goose Brewery	Easton

[105 rows x 2 columns]

0.0.1 1. Replace the Headers

[5]:
```
names = []
with open('maryLand_names.txt', 'r') as f:
    for line in f:
        f.readline()
        var = line.split(":")[0]
        names.append(var)
names
```

[5]:
```
['brewery_name', 'location_name']
```

[6]:
```
df.columns = names
df.head()
```

[6]:
```
            brewery_name location_name
0  1623 Brewing Company    Eldersburg
1          1812 Brewery    Cumberland
2       7 Locks Brewing     Rockville
3      AleCraft Brewery       Bel Air
4      Antietam Brewery    Hagerstown
```

0.0.2 2. Create a data set with required columns

[7]:
```
df1=df[['brewery_name','location_name']]
df1
```

[7]:
```
                brewery_name             location_name
0       1623 Brewing Company                Eldersburg
1               1812 Brewery                Cumberland
2            7 Locks Brewing                 Rockville
3           AleCraft Brewery                   Bel Air
4           Antietam Brewery                Hagerstown
..                        …                       …
100     Waredaca Brewing Co.               Gaithersburg
101      Waverly Brewing Co.       Baltimore (Hampden)
102         Wet City Brewing  Baltimore (Mount Vernon)
103  White Marsh Brewing Co.               White Marsh
104       Wild Goose Brewery                    Easton

[105 rows x 2 columns]
```

0.0.3 3. Find duplicates

[8]:
```
print("Location Name is duplictaed - {}".format(any(df.location_name.
 ↪duplicated())))
```

```
Postal Code is duplictaed - True
```

0.0.4 4. Find Null values

[9]:
```
print("The column Location name contains NaN - %r " % df.location_name.isnull().
 ↪values.any())
```

```
The column Location name contains NaN - False
```

0.0.5 5. Identify outliers and bad data

[16]:
```
size_prev = df.shape
df = df.dropna()
size_after = df.shape
print("The size of previous data was - {prev[0]} rows and the size of the new␣
 ↪one is - {after[0]} rows".
```

```
      format(prev=size_prev, after=size_after))
```

```
The size of previous data was - 105 rows and the size of the new one is - 105
rows
```