# Bilenkin540_Term_Project_Milestone_3

April 29, 2025

## 1 DSC 540 - Project Milestone 3: Cleaning and Formatting Website Data

Website Source:** List of countries by alcohol consumption per capita

In this Milesonte, I will perform at least 5 data cleaning and transformational steps against the above website.

[36]:
```
# Importing necessary libraries
import pandas as pd
import requests
from bs4 import BeautifulSoup
```

### 1.0.1 Loading the Website HTML

[37]:
```
# Wikipedia URL for alcohol consumption per capita
url = "https://en.wikipedia.org/wiki/
 ↪List_of_countries_by_alcohol_consumption_per_capita"

# Fetching the page content
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")

# Finding all tables on the page
tables = soup.find_all("table", {"class": "wikitable"})

# Extracting table data
def extract_table_data(table):
    headers = [header.text.strip() for header in table.find_all("th")]
    data = []
    for row in table.find_all("tr")[1:]:  # Skipping header row
        cells = row.find_all("td")
        if len(cells) > 0:
            row_data = [cell.text.strip() for cell in cells]
            data.append(row_data)
    return pd.DataFrame(data, columns=headers)

# Getting the country-wise alcohol consumption table
```

```
df_countries = extract_table_data(tables[1])

# Showing the first five rows
df_countries.head()
```

[37]:

| | Country | 1996[9] | 2016[10] | 2019[6][a] |
|---|---|---|---|---|
| 0 | Afghanistan | - | 0.2 | 0.2 |
| 1 | Albania | 2.59 | 7.5 | 5.1 |
| 2 | Algeria | 0.27 | 0.9 | 0.6 |
| 3 | Andorra | - | 11.3 | 11.1 |
| 4 | Angola | 1.58 | 6.4 | 6.2 |

### 1.1 Step 1: Clean Column Names

Removing the references like [9], [10], [6][a].

Renaming columns to simple names: 'Country', 'Alcohol_1996', 'Alcohol_2016', 'Alcohol_2019'.

[38]:
```
# Renaming columns to remove footnote markers and make them more readable
df_countries.columns = ['Country', 'Alcohol_1996', 'Alcohol_2016',␣
 ↪'Alcohol_2019']

# Displaying the first 5 rows after renamed to confirm
df_countries.head()
```

[38]:

| | Country | Alcohol_1996 | Alcohol_2016 | Alcohol_2019 |
|---|---|---|---|---|
| 0 | Afghanistan | - | 0.2 | 0.2 |
| 1 | Albania | 2.59 | 7.5 | 5.1 |
| 2 | Algeria | 0.27 | 0.9 | 0.6 |
| 3 | Andorra | - | 11.3 | 11.1 |
| 4 | Angola | 1.58 | 6.4 | 6.2 |

### 1.2 Step 2: Replace "-" with NaN

The "-" symbol in cells means missing data.

I will replace it with np.nan so pandas can recognize missing values properly.

[39]:
```
import numpy as np

# Replacing "-" with NaN for better missing value handling.
df_countries.replace('-', np.nan, inplace=True)

# Displaying first 5 rows for confirmation
df_countries.head()
```

[39]:

| | Country | Alcohol_1996 | Alcohol_2016 | Alcohol_2019 |
|---|---|---|---|---|
| 0 | Afghanistan | NaN | 0.2 | 0.2 |
| 1 | Albania | 2.59 | 7.5 | 5.1 |

```
2      Algeria         0.27          0.9          0.6
3      Andorra          NaN         11.3         11.1
4       Angola         1.58          6.4          6.2
```

### 1.3 Step 3: Convert Alcohol Values to Numeric

At this point all columns are strings because of "-".

I need to convert 'Alcohol_1996', 'Alcohol_2016', and 'Alcohol_2019' into float numbers.

[40]:
```
# Converting data types to float.
for col in ['Alcohol_1996', 'Alcohol_2016', 'Alcohol_2019']:
    df_countries[col] = pd.to_numeric(df_countries[col], errors='coerce')

df_countries.dtypes
```

[40]:
```
Country          object
Alcohol_1996    float64
Alcohol_2016    float64
Alcohol_2019    float64
dtype: object
```

### 1.4 Step 4: Standardize Country Names

Ensuring that the country names are Title Case by capitalizing first letter of each word.

For example: 'united states" to "United States" for more professional readability.

[41]:
```
# Applying Title Case formatting to the 'Country' column.
df_countries['Country'] = df_countries['Country'].str.title()

# Displaying first 5 rows
df_countries.head()
```

[41]:
```
       Country  Alcohol_1996  Alcohol_2016  Alcohol_2019
0  Afghanistan           NaN           0.2           0.2
1      Albania          2.59           7.5           5.1
2      Algeria          0.27           0.9           0.6
3      Andorra           NaN          11.3          11.1
4       Angola          1.58           6.4           6.2
```

### 1.5 Step 5: Identify and Drop Duplicate Countries (if any)

Checking for any duplicate country names and drop them.

[42]:
```
# Ensuring each country appears only once.
df_countries = df_countries.drop_duplicates(subset='Country')
df_countries.reset_index(drop=True, inplace=True)

# Displaying the first 5 rows
```

```
df_countries.head()
```

[42]:

| | Country | Alcohol_1996 | Alcohol_2016 | Alcohol_2019 |
|---|---|---|---|---|
| 0 | Afghanistan | NaN | 0.2 | 0.2 |
| 1 | Albania | 2.59 | 7.5 | 5.1 |
| 2 | Algeria | 0.27 | 0.9 | 0.6 |
| 3 | Andorra | NaN | 11.3 | 11.1 |
| 4 | Angola | 1.58 | 6.4 | 6.2 |

### 1.6 After all the cleaning is done, printing the first 20 rows for preview.

[43]:
```
# Printing final cleaned dataset
df_countries.head(20)
```

[43]:

| | Country | Alcohol_1996 | Alcohol_2016 | Alcohol_2019 |
|---|---|---|---|---|
| 0 | Afghanistan | NaN | 0.2 | 0.2 |
| 1 | Albania | 2.59 | 7.5 | 5.1 |
| 2 | Algeria | 0.27 | 0.9 | 0.6 |
| 3 | Andorra | NaN | 11.3 | 11.1 |
| 4 | Angola | 1.58 | 6.4 | 6.2 |
| 5 | Antigua And Barbuda | NaN | 7.0 | 8.5 |
| 6 | Argentina | 9.58 | 9.8 | 8.0 |
| 7 | Armenia | 0.84 | 5.5 | 5.0 |
| 8 | Australia | 9.55 | 10.6 | 10.1 |
| 9 | Austria | 11.90 | 11.6 | 12.0 |
| 10 | Azerbaijan | 4.16 | 0.8 | 2.0 |
| 11 | Bahamas | NaN | 4.4 | 4.4 |
| 12 | Bahrain | NaN | 1.9 | 1.6 |
| 13 | Bangladesh | NaN | 0.0 | 0.1 |
| 14 | Barbados | 8.37 | 9.6 | 9.5 |
| 15 | Bhutan | NaN | 0.6 | 0.2 |
| 16 | Belarus | 8.14 | 11.2 | 10.9 |
| 17 | Belgium | 10.94 | 12.1 | 10.3 |
| 18 | Belize | 5.85 | 6.7 | 5.7 |
| 19 | Benin | 1.39 | 3.0 | 8.3 |

### 1.7 Ethical implications of data wrangling

In this milestone, I performed five cleaning and transformation steps on publicly available data from Wikipedia regarding alcohol consumption per capita by country. I renamed column headers to a more readable format, converted missing values represented by a minus symbol into recognized null values, converted textual data into numeric types, standardized country name casing, and checked for duplicate records to avoid counting the same country more than once.

Since this data comes from Wikipedia, a publicly accessible source, there are minimal—if any—legal or regulatory concerns. However, because Wikipedia can be edited by anyone, there is no guarantee of complete accuracy or reliability. Another potential risk in the data wrangling process is accidentally omitting valuable information or misrepresenting the data through incorrect trans-

formations. For example, the symbol "–" could be interpreted as zero in some contexts (such as accounting), rather than as a missing value. In this case, however, it was reasonable to assume the symbol indicated missing data.

To avoid ethical concerns, I did not fabricate or infer any data values; I only worked with the available data and applied standard cleaning techniques to improve its structure and usability. If this dataset were to be used for medical research or policy decisions, it would need to be cross-verified with an official source, such as the World Health Organization (WHO). Overall, the data cleaning and transformation process was performed ethically, transparently, and with careful documentation.