Student Name: Krishnakanth Srikanth Student ID: s3959200

Data Preparation

As we all know, the most primary and the important step for a Data Scientist to show up the results is Data Preparation which includes exploring the data, analysing the data, cleaning and interpreting the information from the data to the world. For this process, I have first imported the data to Jupyter Notebook and made a copy from the original dataset so as to work on it without affecting the original data. Then, had a look at the data using **.head()** and **.tail()** functions, also, checked for any missing values or incorrect values using **.isnull()** and **.value_counts()** respectively. From the obtained results, I found some values in the data, which are misspelt, needs to be corrected and the missing values needs to be either removed or imputed with necessary values. I also found some features in the data which does not support me for my analysis and wanted to remove them as well. Finally, after all the potential issues and errors are rectified, I have exported the clean dataset to the new .csv files. The errors I corrected or rectified is explained below.

Error 1:

Check for typos:

The first error I rectified after analysing the whole data are the ones that were spelt wrong. For example, in **Total School Age.csv** I found the 'Lower middle income (LM)' of 'Income Group' was spelt wrong as 'Lowerr middle income (LM)' which might be a typo during data entry. So to clear this error I replaced 'Lowerr middle income (LM)' with 'Lower middle income (LM)' using **.replace()**

Also, in the **Secondary.csv**, I found the **ISO3** value for the country '**Angola**' as 'AGOA' which actually should be 'AGO'. Hence, I rectified it using **.replace()**

Error 2:

Check for extra white spaces:

For this error check, when importing the dataset, I had used **skipinitialspaces = True** (which by default is False) of **.read_csv()** to remove the extra whitespaces if present. So, here, using skipinitialspaces = True, I have removed the extra whitespaces.

Error 3:

Handling missing values:

Like always, no dataset is 100% clean and tidy and it is hard to find the one which is clean. The trickiest and the most wanted issue in cleaning the data is handling the missing values. In this problem, I handled the missing values in few features, which made my dataset look clean for my further process of analysis and exploration. The way these can be handled may vary from person to person. I had handled the missing values or NaN values in the following ways.

- First, the features where the missing values were, are of type 'object' and I had to use those features as part of my analysis to get better results for which I converted the data type those columns from '**object**' to '**float64**' by replacing the '%'. I used 'float64' because the **NaN values can only be represented in float type** and not as integer.
- After the data type conversion, I checked for the features with null or NaN values and found that some instances had 3 or more than 3 (out of 4) observations empty which in my view would not be good for final result if imputed, hence I removed those rows from the dataset by using **.loc()** and the values of the instances with mean less than 0.25 along columns (**axis=1**).

- Once the empty rows were removed, the resulted data frame had NaN values only in some of the instances and features. I imputed their respective means using **.fillna()** along the rows (**axis=0**). So, after this step, the data frame is clear with no missing values.
- Once after imputing, I rounded the values to 1 decimal using **.round()** and changed the data type from '**float64**' to '**integer**'

Error 4:

Checking for duplicates:

After cleaning the missing values, I checked for the **duplicates** in all the three csv files and found it to be **none**.

Error 5:

Sanity checks for impossible values:

The next issue I faced during data cleaning is with the feature '**Time Period**' because some values in this feature had values split by '-' and some did not. To overcome this, I created a new feature named '**Survey Year**' in which I added values from the 'Time Period' column one by one and for values with multiple years, I added the values of the most recent time period as per the information given in readme.txt ('* Time frame for survey: Household survey data as of year 2010 onwards are used to calculate the indicator. For countries with multiple years of data, the most recent dataset is used.') using **.apply()**

After this, when checked the values in the created new feature 'Survey Year', I found some of the years were irrelevant (e.g., 99, 2567, etc.,) and might be typos during data entry. I used **.replace()** and its **method='bfill'** to replace those years backwards (i.e., added the preceding value).

After cleaning the errors and issues, the new cleaned dataset is exported to csv files using **.to_csv()**

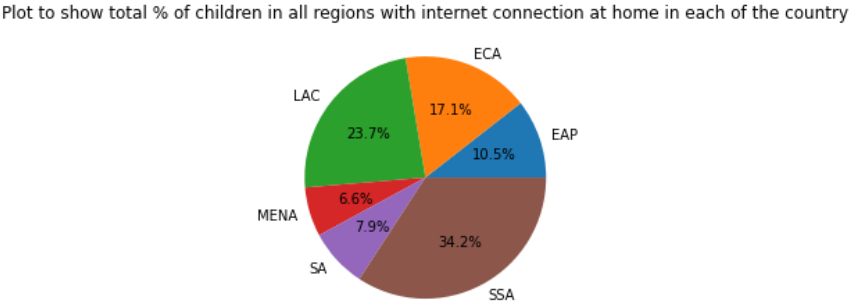
Data Exploration

Here, I used **.describe()** and **boxplots** from matplotlib (**kind='box'**) in '**Total**' to check for the outliers and **found no outliers**.

Task 2.1:

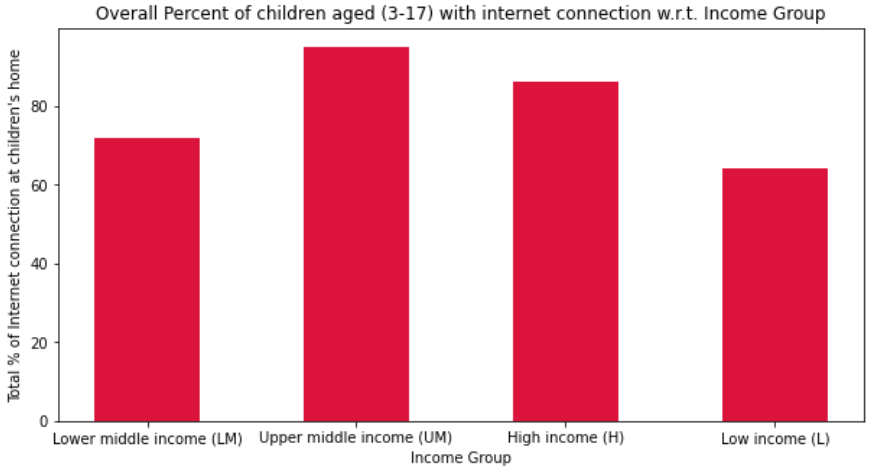
For this task, first I chose a **Nominal variable** which is **Region** and used a **Pie chart** to show the different categories of region with its percentage. As a whole, my plot is to show **Total percentage of children in all region with internet connection at home in each of the country**. For this, I have **grouped the data** of Primary school children **with respect to 'Region'** and made a plot of it using **matplotlib** with **kind='pie'.** From the below graph, we can see that, the region SSA tops with 34.2% of children with internet connections.

Plot:



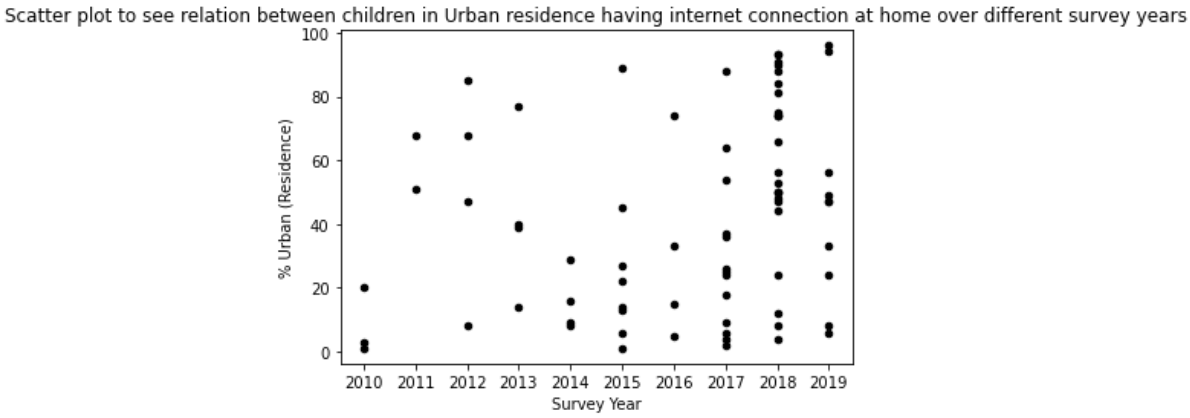
Secondly, I chose an **Ordinal variable** which is **Income Group** and used a **Bar plot** to **visualise the Overall percentage of children aged (3-17) with internet connection with respect to their Income Group.** For this purpose, I have plotted the **'Income Group' along x-axis** and **'Total' along y-axis** and made a bar plot using **plt.bar()** from **matplotlib.** From the graph, we can see that, children belonging to Upper Middle class group had most percentage of internet connection at home.

Plot:



For the final plot, I chose a **Numerical variable** which is **"Urban (Residence)"** and used a **Scatter plot** to see **relation between children in urban residence having internet connection at home over different survey years**. For this, I sorted the **'Survey Year'** column in ascending order using **sort_values()** and plotted the graph using **matplotlib** with **kind='scatter'** and chose **x** as **'Survey Years'** and **y** to be the **percentage of children in Urban residence with internet connection.** We can see from the graph that in the year 2018, most number of children in urban residence had internet connection at home.

Plot:



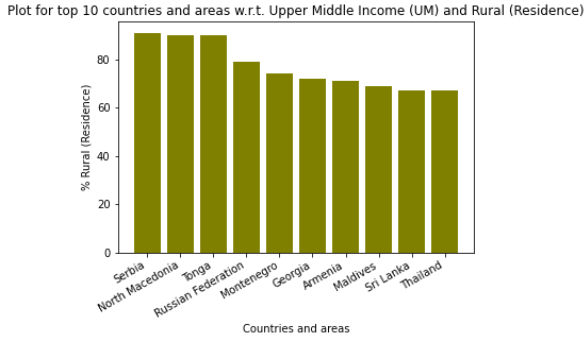
Task 2.2:

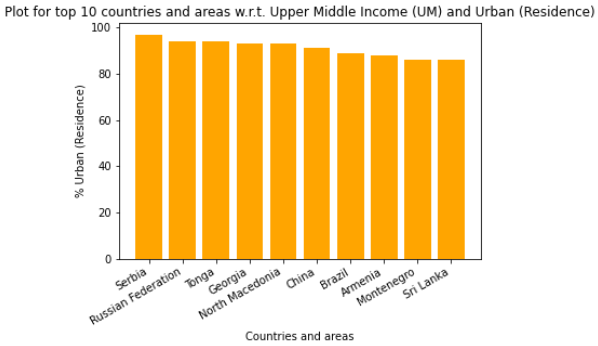
For this task, I have created different dataframes with respect to the Income Group and found the top 10 countries and region for each income group in accordance to Rural and Urban (Residence). Hence, I have got a total of 8 different plots (2 for each Income group) (i.e.,) for e.g. Top 10 countries and region for Lower middle income (LM) and Rural (Residence).

I have created **bar plots** for all the different conditions **by filtering out the data from 'Total School Age.csv' with respect to each of the Income Group, found the top 10 countries and region with highest Rural and Urban (Residence) each** and plotted using **plt.bar()** for each of the scenarios.

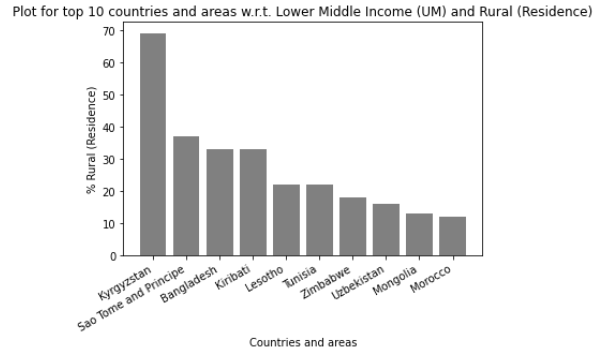
Plots:

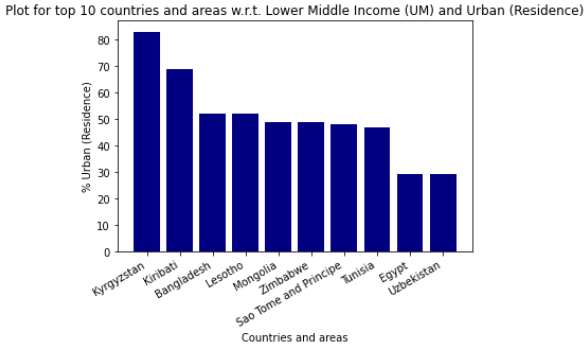
For Upper middle income (UM):



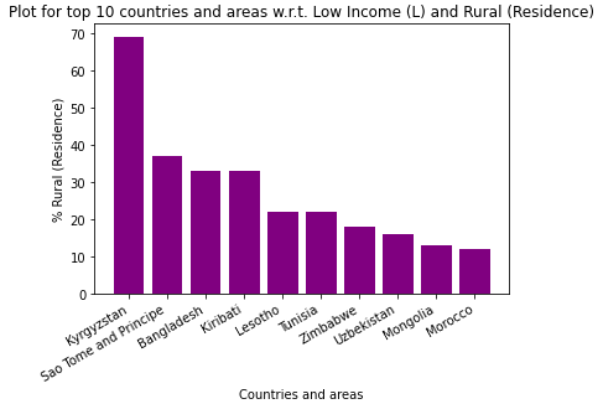


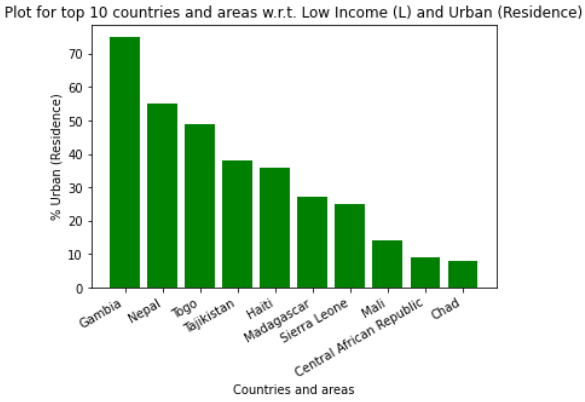
For Lower middle income (LM):



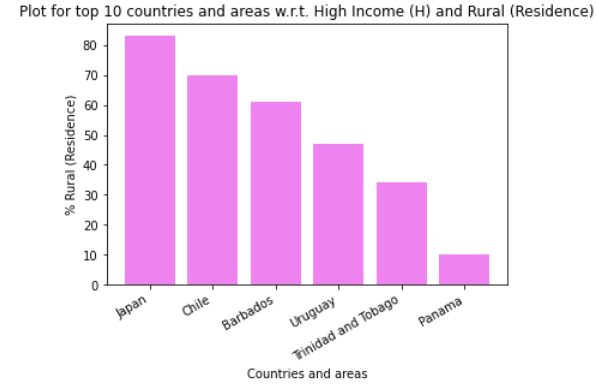


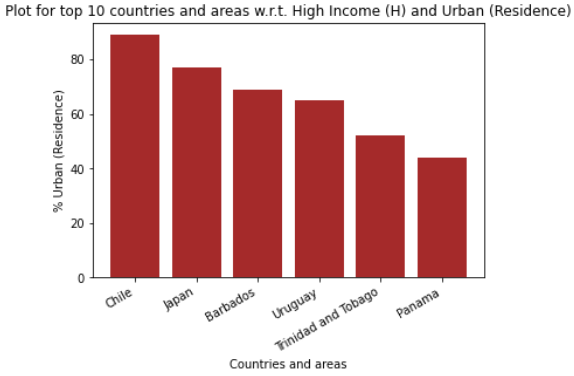
For Low Income (L):





For High Income (H):

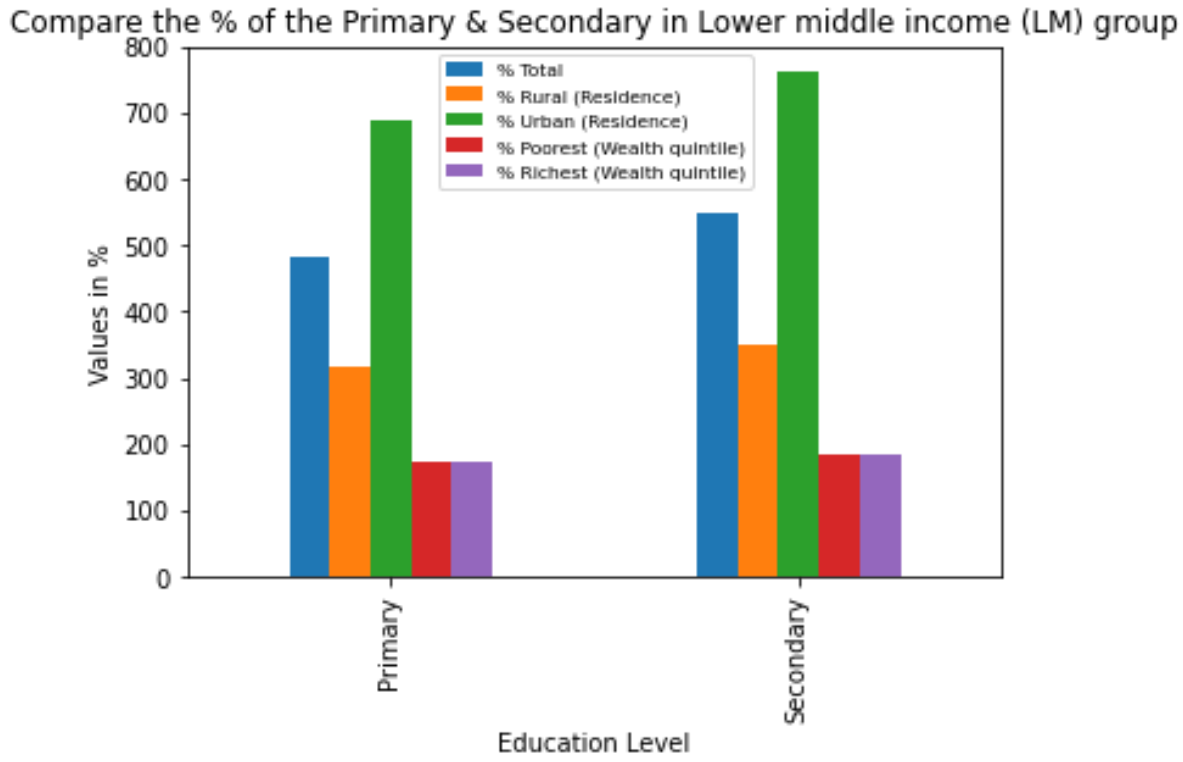




Task 2.3:

For this task, I have **filtered** out the data from both the Primary and Secondary based on **'Lower middle income (LM)'** income group and created new data frames with Income group using **.loc().** Then, filtered out only Total, Rural (Residence), Urban (Residence), Poorest (Wealth quintile), Richest (Wealth quintile) variables as their features and made a sum of all the percentages using **.sum()**, created a new column '**Education Level**' with values '**Primary**' and '**Secondary**'. Finally, merged the data frames using **.concat()**. Then, made a **bar plot** using the created new merged data frame to **compare the percentages between the Primary and Secondary school children in Lower middle income (LM) group**.

Plot:



From the above plot, we can confirm that children in the Urban (Residence) and Rural (Residence) at Secondary school have more percentage of internet connection at home when compared to those in Primary school. On the other hand, the percentages of both the Primary and Secondary school children remains the same with respect to Wealth (Poor or Rich). Thus, **children's Residence, the living area has more impact than their Wealth**. On the whole, the **total percentage of children at Secondary school have higher percentage of internet connection at home than the Primary school children**.

References:

- *Plot multiple lines in Matplotlib* (2020) *GeeksforGeeks*. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/plot-multiple-lines-in-matplotlib/ (Accessed: April 4, 2023).
- *Get N-largest values from a particular column in pandas DataFrame* (2018) *GeeksforGeeks*. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/get-n-largest-values-from-a-particular-column-in-pandas-dataframe/ (Accessed: April 4, 2023).
- Chen, B. (2022) *Creating a dual-axis combo chart in Python*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/creating-a-dual-axis-combo-chart-in-python-52624b187834 (Accessed: April 4, 2023).