Assignment 1: Data Cleaning and Summarising

Jung-De Chiou (s4068959)

Data Preparation

Error Type 1: Typo -- Typographical Errors in Categorical Data

The "Income Group" column contains typographical errors, specifically, incorrect entries such as 'Lower middle income (LMM)' and 'Lower middle income (LLM)', which should have been recorded as 'Lower middle income (LM)'. These typos can cause incorrect categorization and analysis, as they create additional, unintended categories.

Identification:
To identify typographical errors, the "Income Group" column's unique values were examined using the unique() method in Pandas. These values were then compared against a predefined list of expected values (the correct data format I manually input). This approach can reveal any discrepancies and store them in a variable for further rectification in the following step. (detailed steps can be seen in ipynb file)

Correction:
I used the replace() method in Pandas to correct typographical errors. Specifically, I replaced the incorrect entries 'Lower middle income (LMM)' and 'Lower middle income (LLM)' with the correct value 'Lower middle income (LM)'.

Error Type 2: Missing Values

In the dataset, the following columns had missing values:

- **Residence (Rural)**: 5 missing values
- **Residence (Urban)**: 4 missing values
- **Wealth quintile (Poorest)**: 7 missing values
- **Wealth quintile (Richest)**: 7 missing values

Identification:
For missing values, I used isnull() method to identify along with sum() to count the total missing values in each relevant column. This method provides a clear count of missing data, allowing me to pinpoint the specific columns that need attention.

Correction:
For columns like "Residence" and "Wealth quintile" (which contain percentage values), I use fillna(-1) function to replace missing values. The reason for filling these missing values with -1 is that this dataset represents raw statistical results, and there are significant differences across countries or regions. Therefore, using an average or any statistical method to fill in a default value is not suitable. By inserting -1, it serves as an abnormal value, alerting analysts during later data analysis that an issue needs to be addressed or excluded.
For the "Time period" column, missing values were replaced with NaT (Not a Time) to ensure that any subsequent time-based data processing can properly identify and manage these abnormal values, allowing for targeted exclusion or correction as needed.

Error Type 3: Duplicate

Duplicate rows were identified in the dataset for certain entries:

- **Rows 30 and 31 for GTM (Guatemala)**
- **Rows 76, 77, and 78 for TGO (Togo)**

Identification:
Duplicates were detected using the duplicated() method in Pandas, which flagged identical rows as ‘True’ based on all columns.

Correction:
I retained only the first occurrence of each duplicated entry and removed the rest using the drop_duplicates() method. This step ensures that each observation is unique and correctly represented in the analysis.

Error Type 4: Out-of-range Values

In this dataset, such errors were identified in:

- **Wealth quintile (Richest)** – An anomaly was found in row 59 where the value significantly deviated from the expected range (110%).
- **Year**: The "Year" column contained values such as 3562, 2099, and 2076, which are outside the realistic range for this dataset.

Identification:
I defined acceptable ranges for certain types of data (like in percentage values is 0% - 100%) and checked whether any values fell outside these ranges. For instance, for the percentage values, I used the apply(lambda x: (x < 0) | (x > 1)) function to check the columns. If there are any out-of-range values detected, put them in an out_of_range variable.

For ‘Year” data, I create a check_time(value) method to identify the out-of-range values. After changing their data type from string to datetime, using the if-else statement and comparison operators, like if (2000 <= year <= current_year):, to select abnormalities and return them as -1.

Correction:
For percentage values, I used the clip(0, 1) function to correct out-of-range data. This method adjusts any values below 0 to 0, and any values above 1 to 1 (I already converted % to decimal, so the upper bond is 1 not 100), ensuring all percentage values fall within the acceptable range.

For the "Year" data, the previous identification step already replaced the erroneous values with -1. The next step involves using another method I defined, to_period(value) (which also handles structural errors), to process these values further. In this method, an if-else statement is used to convert any -1 values to pd.NaT, making them recognizable as missing or invalid date values in subsequent time-related analyses.

Error Type 5: Structural Error

Structural errors refer to inconsistencies in the format or organization of data within a column, making it difficult to process or analyze correctly. Two primary structural issues were noted:

- **Year Format Inconsistencies**: Different formats were used for years, such as "2015" and "2014-2015", leading to confusion during time-based analysis. Also, the original string type will impede further time-related data processing.
- **Percentage Values**: Some columns contained percentages formatted as strings, which prevented them from being used in numerical calculations.

Identification & Correction:
I processed the Year data by using a custom to_period() method to handle the "Time period" column. This method resolved inconsistencies where some entries represented a single year (e.g., "2015") and others represented a year range (e.g., "2014-2015"). To standardize the format, I split the "Time period" column into two new columns: "Start year" and "End year" (Single year data’s End year will be the same as the Start year).

After this, I converted all values from the string type to the Period type. Also, I set missing values to NaT (Not a Time) to ensure accurate time-based analysis during subsequent statistical processing.

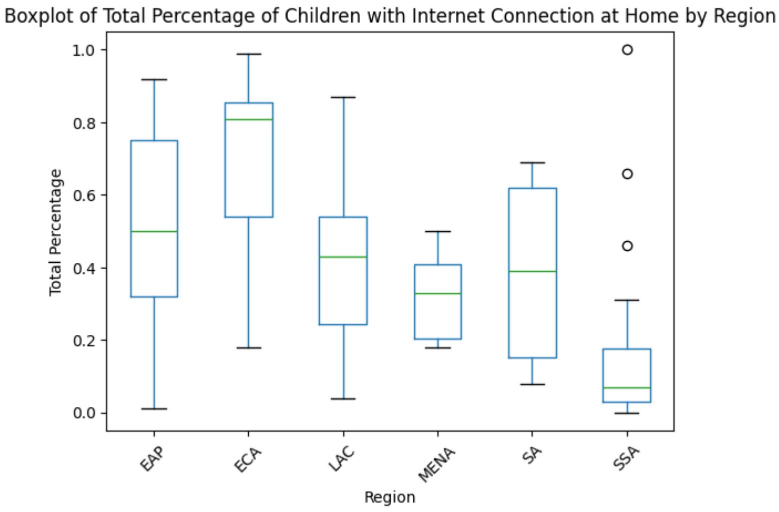
For the percentage data, I used a custom convert_to_numeric() method to process all the percentage columns. This method went through each percentage column and used the replace() function to remove the "%" symbol. Then, I converted the remaining values to floats and divided each value by 100 to present them in decimal format (e.g., "10%" became 0.1, "45%" became 0.45). This transformation ensures consistent numerical representation and facilitates percentage calculations in further analysis.

Data Exploration

Task 2.1

1. Exploration Process:

I used "df.boxplot(column='Total', by='Region', grid=False)" to create a boxplot illustrating the distribution of the "Total" percentage of children with internet access at home, grouped by the "Region" column. To enhance the presentation, I utilized "plt.suptitle(), plt.xlabel(), plt.ylabel(), and plt.xticks()" to adjust the display of titles and labels for better readability.



Additionally, I calculated the median values for each region using "df.groupby('Region')['Total'].median().reset_index()", allowing a precise representation of central tendencies within each region.

	Region	Median_Total_Percentage
0	EAP	0.50
1	ECA	0.81
2	LAC	0.43
3	MENA	0.33
4	SA	0.39
5	SSA	0.07

2. Results

(1) Central Tendencies (Medians):

- ECA has the highest median, indicating a generally high level of connectivity across the region.
- MENA and SSA have the lowest medians, suggesting that internet connectivity among children is much less prevalent in these regions.

(2) Spread and Variability:

- EAP, ECA, and LAC exhibit a wide spread, indicating high variability and significant disparities in internet access among children within these regions.
- MENA and SSA display tighter spreads, reflecting more uniformity in low internet access across these regions. However, SSA shows three prominent outliers, suggesting either data collection errors or the presence of a few countries in this region with significantly higher internet access than the average.

(3) Insights:

- ECA generally demonstrates the highest levels of internet connectivity among children, while EAP and LAC are more moderate but still show noticeable variability. This suggests that within these regions, while some countries are well connected, others significantly lag, highlighting the impact of development levels and income disparity.
- SSA stands out with the lowest overall connectivity and the narrowest range, indicating consistently low access across the region, but with a few notable exceptions where access is significantly higher.

- Regions like MENA and SA (South Asia) show mixed results with moderate levels of internet access, and their tighter spreads indicate more consistency within each country.

(4) Personal summary and reflection:

Overall, internet accessibility can reflect the level of development and wealth disparities within regions and countries. As detailed above, regions with higher variability in access also tend to have more pronounced economic disparities. If comparing these findings with specific country-level GDP or development indices could provide deeper insights, revealing how internet access among children correlates with economic inequality. This highlights the potential of internet access as an indicator of a country's development status.

Task 2.2

I first calculated the mean values for both the "Wealth quintile (Poorest)" and "Wealth quintile (Richest)" columns using the mean() function. The result as shown below:

Mean for Wealth quintile (Poorest): 0.09045

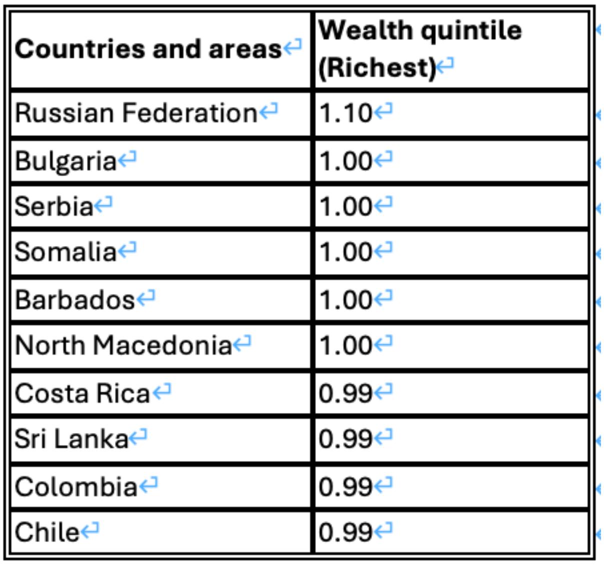
Mean for Wealth quintile (Richest): 0.4894

Then, I used sort_values() method by their respective wealth quintiles in descending order to identify the top 10 countries with the highest values in each category

Wealth Quintile (Poorest):

Countries and areas	**Wealth quintile (Poorest)**
Somalia	1.00
Russian Federation	0.88
Brazil	0.84
Tonga	0.83
Chile	0.75
Sri Lanka	0.71
North Macedonia	0.68
Serbia	0.65
Japan	0.64
Kyrgyzstan	0.56

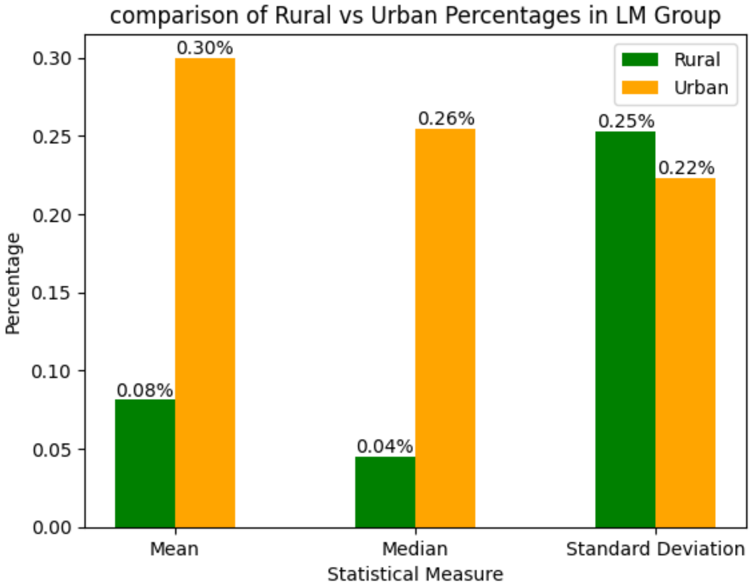
Wealth Quintile (Richest):



Countries and areas	**Wealth quintile (Richest)**
Russian Federation	1.10
Bulgaria	1.00
Serbia	1.00
Somalia	1.00
Barbados	1.00
North Macedonia	1.00
Costa Rica	0.99
Sri Lanka	0.99
Colombia	0.99
Chile	0.99

Task 2.3

1. Exploration Process:
   I first filtered the dataset to include only the relevant income group using df[df['Income Group'] == 'Lower middle income (LM)']. I then calculated the mean, median, and standard deviation for both Rural and Urban columns to understand the central tendencies and variability of internet access in these areas. Next, I visualized the results using a bar chart to compare rural and urban percentages across the three statistical measures (mean, median, and standard deviation).



2. Result:
   (1) Central Tendencies:

- The mean and median values indicate that internet access in urban areas is consistently higher than in rural areas within the Lower Middle Income group.
- Urban areas exhibit higher percentages, suggesting a significant disparity between rural and urban internet connectivity.

(2) Spread and Variability

- The standard deviation shows that variability in internet access is higher in rural areas, indicating that rural access levels are more inconsistent.

(3) Summary:

The comparative analysis underscores the persistent inequality in internet access between rural and urban settings in LM countries. Although this analysis does not include a comparison with other income groups, the absolute numbers clearly reveal a significant urban-rural divide within LM countries. Further comparison of rural and urban data across different income groups could provide deeper insights into "how the urban-rural gap varies with levels of wealth", highlighting potential differences in the degree of disparity between income groups.

Use of AI Tools

1. How does Val/ChatGPT help with completing Task1&2

During the Data Cleaning process in Task 1, although I was aware of the various errors present, I was uncertain about the correct Python code to address them. ChatGPT was instrumental in helping me write accurate code that aligned with my approach to handling the data. It provided detailed explanations of each function's purpose and usage, saving me significant time that would have otherwise been spent searching through official documentation. This greatly improved my efficiency.

For Task 2, ChatGPT played a key role by suggesting statistical techniques and visualizations to explore the data effectively. After discussing my intended data analysis methods with ChatGPT, it provided valuable feedback on the best ways to approach the analysis. For example, it recommended using specific types of charts for Task 2-3 and guided me on how to implement these visualizations in Python. ChatGPT also explained how to interpret boxplots and even shared relevant educational resources, enhancing my understanding. After writing the code, ChatGPT helped check for bugs and test the functionality, ensuring that the output matched my expectations.

2. How I utilize AI tool in this assignment

Beyond the code assistance and error debugging mentioned above, I used ChatGPT to discuss broader industry practices for handling and visualizing data, like those found in the CSV files used in this assignment. This helped me gain a deeper understanding of standard data science methods. ChatGPT also assisted with refining the report's text, enhancing the clarity and professionalism of my findings.

Overall, while ChatGPT is a powerful tool, it primarily serves as an auxiliary resource that improves the efficiency of identifying, correcting code issues, provides direction, and suggestions for analysis. However, the decision-making process and how to guide the AI remain my responsibility. One of ChatGPT's undeniable benefits is the time saved compared to the traditional method of slowly searching the web for information—it offers direct answers and sources that I can quickly verify.