Student Name: Kishore Sudhir Student ID: s3971501

Data Preparation

First, I read all three file – Primary.csv, Secondary.csv and Total School Age.csv. All the files have headers as Column A, Column B to Column L. But their actual column starts from first row thus including “header=1” as a parameter in pd. read_csv() function helped to correctly identify the header of tables.

Before Cleaning

Since it wouldn't be feasible to cleanse all data for our needs in Task 2, focusing on a required feature is a good option. Moreover, combining Primary.csv and Secondary.csv will aid in decreasing the duplicated code of the cleaning process, which may be beneficial for Task 2.3, Countries and areas , Region, Sub-region, Income Group and Total is considered potentially useful features of Primary and Secondary csv files. While, Countries and areas, Income Group, Total, Rural (Residence), Urban (Residence) is considered important in Total School Age.csv file. The concat() function is used to combine primary and secondary csv data. Furthermore, a “key” attribute of this function will aid in determining whether the level of schooling is primary or secondary (B. Chen, 2020).

Cleaning

Since the required columns have already been sorted, cleaning these columns is sufficient.

Error 1

Datatypes – Percentages are represented as strings with the character "%" at the end. In our chosen data frame, I corrected this by utilising the "astype(int)" function, which type transforms the data type into integer. But, because we have the percentage "%" symbol, it must be removed before conversion. So, str.strip("%") can be used to remove the percentage symbol from the value.
Issue – There was “SettingWithCopyWarning” warning while

Error 2

Impossible values – As we have columns that represent percentage it is impossible to have values less than 0 and more than 100.

However, in this check one data in the combined data frame (needed_combined_school), there is an entity with percentage as 179 as shown in figure below.

```
# checking if any total perecntage is more than 100 - Found one
needed_combined_school[needed_combined_school["Total"]>100]
```

		Countries and areas	Region	Sub-region	Income Group	Total
Eduction level						
Secondary	75	Ukraine	ECA	EECA	Lower middle income	179

Fig1 – Sanity Check

It could be the result of human error. Where it is possible to type 179 instead of 17 or 79. As, I was unsure of the value. I filled with Null using “np.null”. Similar checks were performed on other percentage-related columns, and no anomalies were discovered.

Error 3

White spaces – White spaces are remove with str.split(“ “)

Error 4

Wrong value and typos

There is a country with wrong name in both data frame as “Bolivia (Plurinational State of)” instead of “Plurinational State of Bolivia”. Which is solved using mask operation. Example code snippet,

```
mask = needed_combined_school["Countries and areas"]=="Bolivia (Plurinational State of)"
needed_combined_school.loc[mask,"Countries and areas"] = "Plurinational State of Bolivia
```

Similarly, in the Total School Age file, there is a income group called “Lowerr middle income” with count as 1. It is typing error that represents “Lower middle income”. It is also fixed like above step.

Error 5

Null Values

The combined data had no null values until the inserted one from step Error 2 – sanity checks. Which is ensured by isna() function.

The are null values in the data frame of Total School Age csv file, their null percentage are as shown in the below code snippet picture.

```
# null percentage
needed_total_school_age.isna().mean()*100
```

```
Countries and areas     0.000000
Income Group            0.000000
Total                   0.000000
Rural                  11.494253
Urban                   8.045977
dtype: float64
```

Fig 2 – Null Percentage in Total School Age

Null values in Rural and Urban column are around 11 and 8 per cent respectively. Thus, handling them is necessary.

Solution – Null value is filled with mean value based on the income group. For example, a null in the rural column with the income group "High Income" is filled with the mean values of all other Rural resident percentages with the income group "High Income."

Code:

```
needed_total_school_age.loc[mask_High_income, 'Rural'] =
needed_total_school_age.loc[mask_High_income,
'Rural'].fillna(needed_total_school_age.loc[mask_High_income, 'Rural'].mean())
```

where mask_High_income is, mask that selects only income group with value as "High income".

Data Exploration

Task 2.1

The selected data

1. Nominal – Region
    a. Region is a categorical value, where order is not significant.

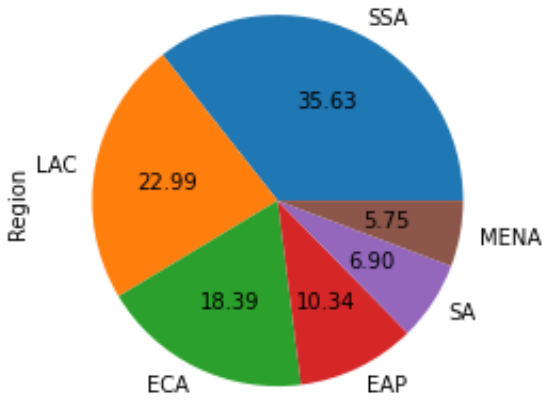

Fig 3 – Pie chart for Primary School Region

2. Ordinal – Income Group
    a. Income group is also a categorical value, but it could be ordered like high income, low income, etc.

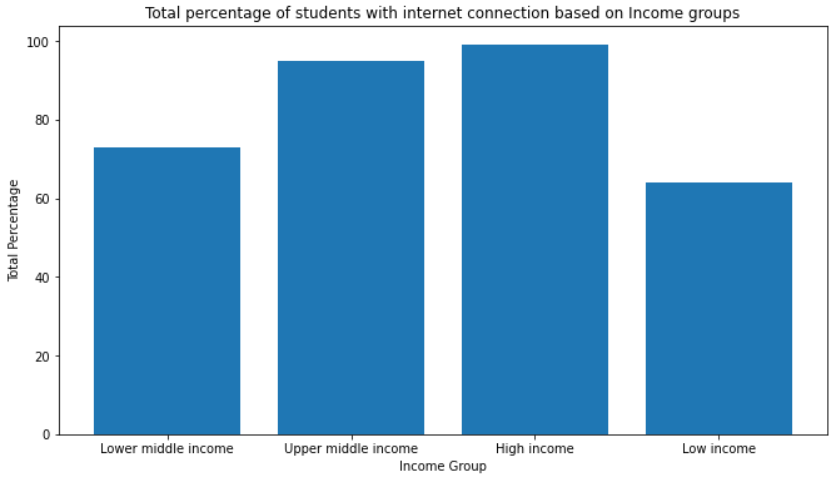

Fig 4 - Total percentage of students with internet connection based on Income groups.

    b. Two plots are made.
        i. Total vs Income Group bar graph – It shows the variation of total percentage among income groups.

ii. Pie chart – It show the variance in Income group. For example, in the below figure, there are 35.63% students in Upper middle income compared to only 9.20% in High income group.

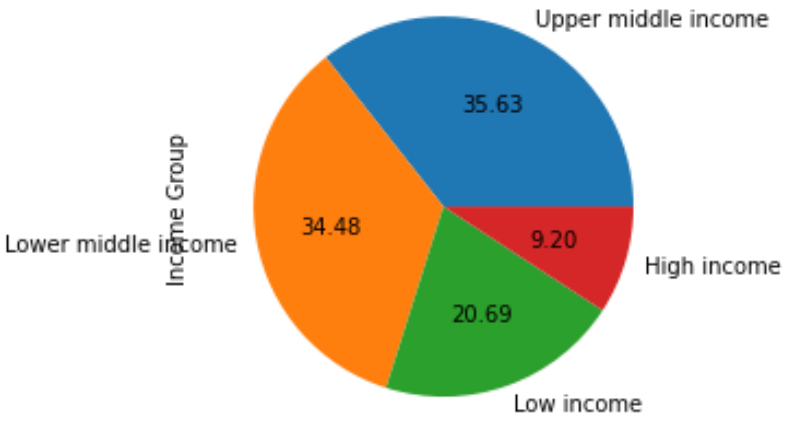

Fig 5 – Income Group pie chart

3. Numeric – Total
    a. Total is a percentage and perfect example for numeric type. Density plot is used to show the density of the total percentage. From graph below, it can be interpreted that there are many counties with percentage range from o to 25.

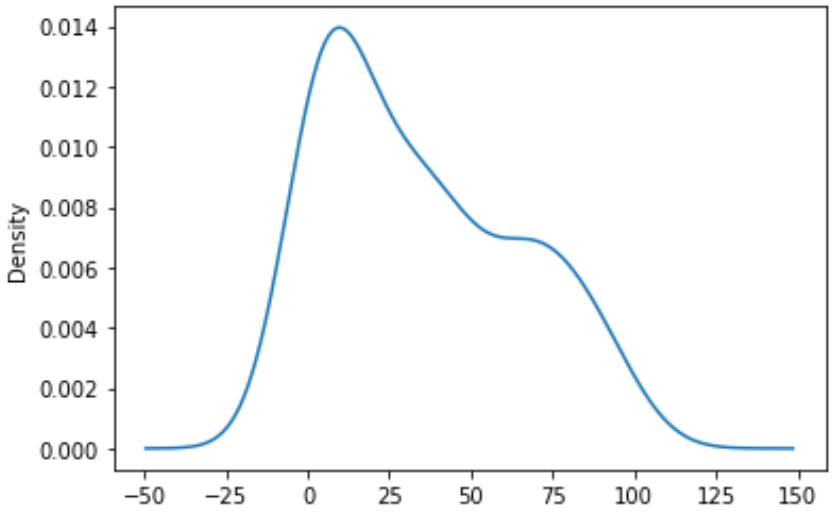

Fig 6 – Density plot for Total Percentage

Task 2.2

Plot 1 – From the line graph it can visible the percentage of students in Urban region in higher than Rural in all the top 10 countries.

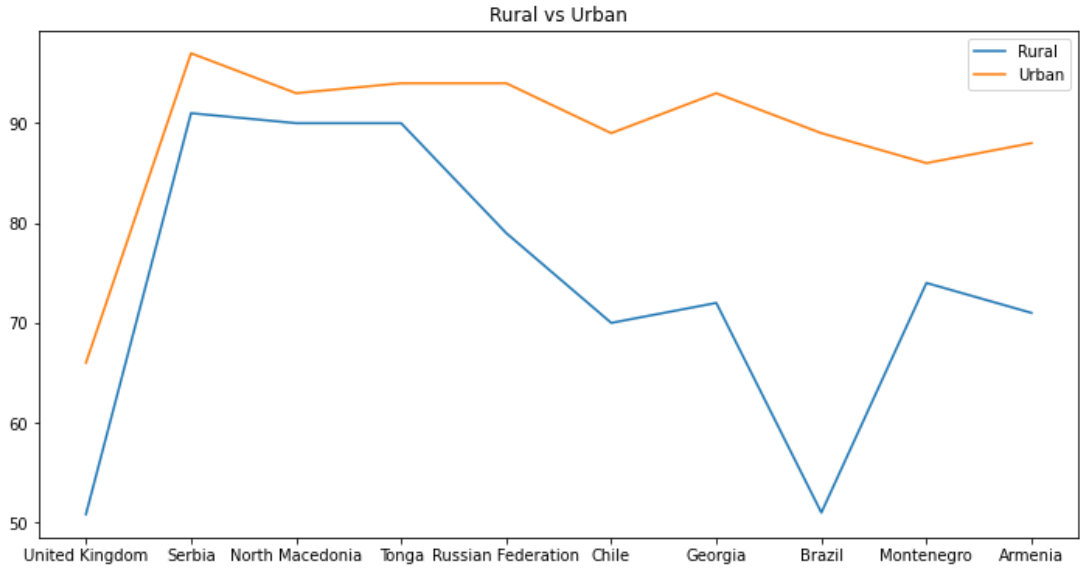

Fig 7 – Line Graph comparing Rural and Urban percentage in terms of top 10 countries.

Plot 2 – Pie chart for Income Group.

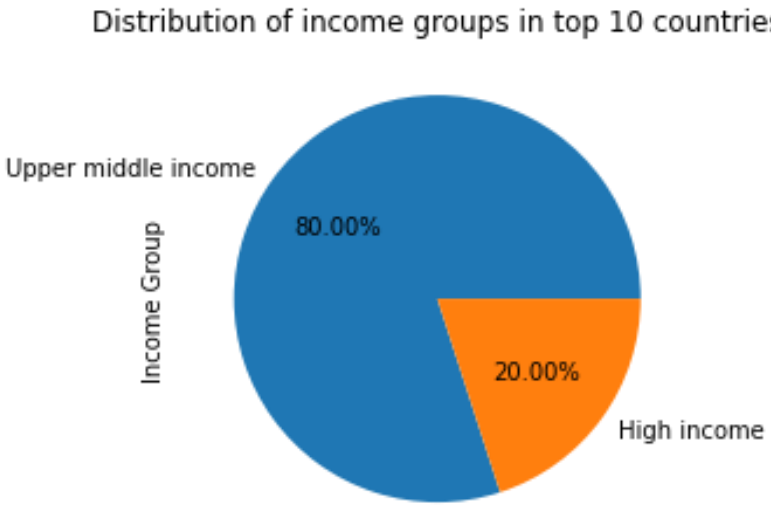

Fig 8 – Income groups pie chart in top 10 countries.

Task 2.3

The box plot can be used to compare primary and secondary school children. The resulting box plot shows that a higher percentage of secondary children have home internet access

than elementary students.

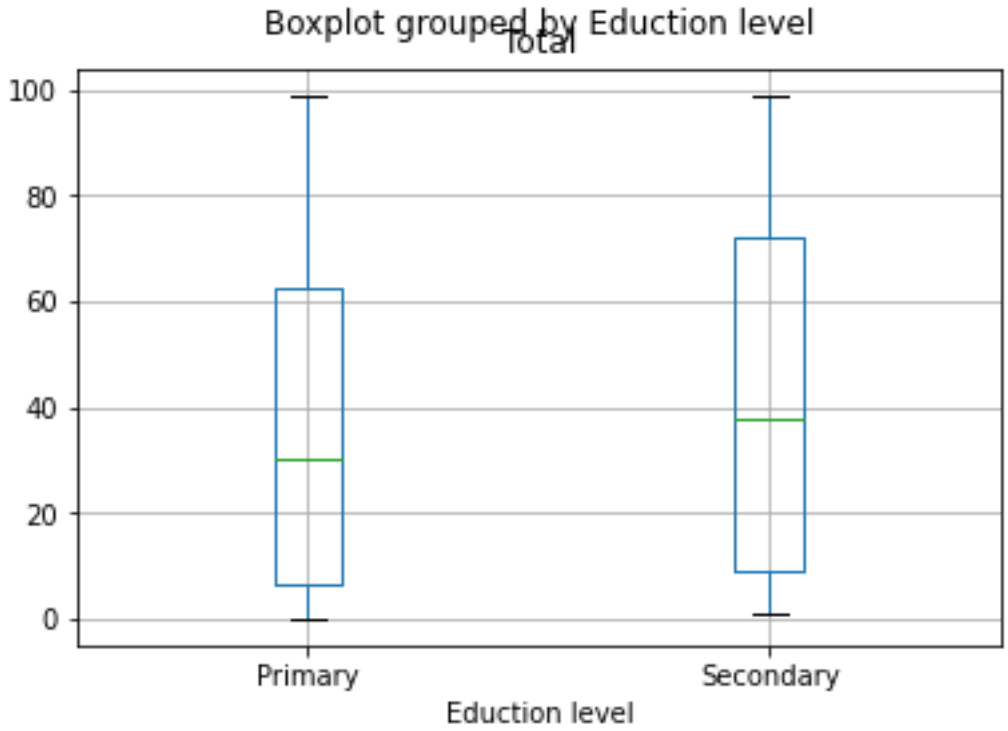

Fig 9 – Primary vs Secondary children with internet connection at home.

Reference list

B. Chen (2020). *Pandas concat() tricks you should know to speed up your data analysis*. [online] Medium. Available at: https://towardsdatascience.com/pandas-concat-tricks-you-should-know-to-speed-up-your-data-analysis-cd3d4fdfe6dd.

pandas.pydata.org. (n.d.). *pandas.to_numeric — pandas 1.4.2 documentation*. [online] Available at: https://pandas.pydata.org/docs/reference/api/pandas.to_numeric.html [Accessed 16 Apr. 2023].

Pryke, B. (2017). *SettingwithCopyWarning: How to Fix This Warning in Pandas*. [online] Dataquest. Available at: https://www.dataquest.io/blog/settingwithcopywarning/.

Stack Overflow. (2012). *python - How to deal with SettingWithCopyWarning in Pandas*. [online] Available at: https://stackoverflow.com/questions/20625582/how-to-deal-with-settingwithcopywarning-in-pandas [Accessed 15 Apr. 2023].