Tourism Case Study Hints

Create the Cleaned_Tourism Table

Part 1

Tasks

1. If necessary, set you working directory. Read the **tourism** table and create the **cleaned_tourism** table.
2. Remove the columns **_1995** through **_2013**.
3. Create the **Country_Name** and **Tourism_Type** columns from values in the **Country** column. Valid values for **Tourism_Type** are *Inbound tourism* and *Outbound tourism*.
4. Run your code and view the **cleaned_tourism** table so far. Remove rows that contain only the country name and tourism type information.

Hints

1. Use `set.wd()` to redefine your working directory and use this directory to create your output files for this case study.
2. Consider using the `dplyr::select` function to keep only the columns you want.
3. Look at the **tourism** table and notice that there is a value in column **A** when there is a country name in the **Country** column. Use a `case_when()` statement to test for this and then assign the value in **Country** to **Country_Name**. In the output table, the value of **Country_Name** should be included in every row. You can also use IF-THEN logic to create the values for **Tourism_Type**. You should explicitly define the column and write the value to every row in the output data.
4. Run your code. In the output table, notice that there are rows that have the same values in the **Country/Country_Name** columns and in the **Country/Tourism_Type** columns. These are rows that do not contain any other data. Use a `filter()` statement to eliminate these rows.

Part 2

Tasks

1. In the Series column, convert values to uppercase and convert ".." and " " to a missing value.
2. Determine the conversion type (Mn or Thousands) that will be used to calculate values for the new **Y2014** column. **Hint**: You might find it easiest to create a new column with this information.
3. In the **_2014** column, change values of ".." and " " to a missing value.

Hints

1. Use a `stringr` function to convert the values in **Series** to uppercase and use a `case_when()` statement to change ".." and " " to a missing value.
2. Look at the data and notice that the **Country** column has some labels that indicate whether the value in **_2014** is in thousands (Arrivals and Departures), or millions (Mn) for various expenditures. This is important information for creating the **Y2014** column with actual numeric values. The conversion type is the last word of the value, so consider using functions to extract that value. You might want to assign that value to a column that you can drop later.
3. Use IF-THEN logic to change ".." and " " to a missing value in the **_2014** column.

Part 3

Tasks

1. Create the **Y2014** column by explicitly converting character values in **_2014** to numeric and multiplying by the conversion type (millions or thousands) that is found in the **Country** column or new column, if you created one.
2. Create the new **Category** column from values in the **Country** column and change the original values to the following valid values: Arrivals, Departures, Passenger Transport - US$, Tourism expenditure in other countries - US$, Tourism expenditure in the country - US$, Travel - US$
3. Include only **Country_Name**, **Tourism_Type**, **Category**, **Series**, and **Y2014** in the output table.

Hints

1. Use `case_when()` logic to test the conversion type (Thousands or Mn), and then convert character values to numeric values and multiply by either 1000 or 1000000. For those rows that do not have values, make sure a missing value is assigned.
2. If you created a column to hold the conversion type, you can use `case_when()` logic to test for the *Mn* or *Thousands* and then create **Category** by using a function to extract the values from **Country**.
3. Use `select` to specify the columns you want in the **Final_Tourism** table.

Create the Final_Tourism Table

Tasks

1. Create a factor variable for the **Continent** column that labels continent IDs with the corresponding continent names found in the table below.
2. Merge the **cleaned_tourism** table with **country_info** to create the **final_tourism** table. Include only matches in the output table. Use the new format to format **Continent**.

Continent ID	Continent Name
1	North America
2	South America
3	Europe
4	Africa
5	Asia
6	Oceana
7	Antarctica

Hints

1. Use functions from the `forcats` package to create the format.
2. Use a join to combine **cleaned_tourism** and **country_info**. Also, be sure to check that the names of the BY columns in both tables are the same before you merge the tables.

Create the NoCountryFound Table

Tasks

1. Create the **NoCountryFound** table that has a list of countries from **Cleaned_Tourism** that are not found in the **country_info** table. This table should have only one row for each country.

Hints

1. Use a join to output rows where there is a row from the **cleaned_tourism** table but not a match in the **country_sorted** table. Make sure you output only unique countries from the **cleaned_tourism** table that do not have a match in the **country_sorted** table. `anti_join` might help.