Write-up

Introduction

This project is about wrangling the data of trends in international migrant stock revised in 2015 based on the tidy data principles. The excel data file includes 8 sheets in total: tables 1 to 6 are sheets containing the data related to the migrant stock, the annex sheet is a table for classification of countries and areas by major area and region, and the notes sheet is a footnote table that explains some information in table 1 to 6. The target goal of wrangling this data in tidyverse is to have a more readable and clear dataset to be analyzed. My desired result is a table that contains its key values in the exact year which means the value of migrant information collected in each table, and also includes a country's identification information such as a country's major area, region, and development situation.

Method & Result

<u>Steps of tidying table 1 to 3, and 5 data</u>

Step 1: Load the data from excel file

Step 2: Drop the meaningless columns -- sort order, notes

Step 3: Drop the region and major area rows (eg. World, Asia, Africa) use drop function with the country code that is greater than 900. (The reason of dropping those rows is I am going to append the major area, region, developed, least developed, and Sub-Saharan Africa columns on the next step)

Step 4: Use a for-loop assign the major area, region, developed, least developed, and Sub-Saharan Africa columns from annex table to the current table regarding the equal country name.

Step 5: Use melt function let gender-year and target values to be two column names in order to satisfy the tidy data principle 1 which is letting column names to be informative, variable names and not values.

Step 6: Use assign function and lambda to separate year and gender to two variables based on principle 2(each column needs to consist of one and only one variable). The following figure is the result without a split.

Out[404]:

Country Code	Country	Data type	Gender	Year	international migrant	Major_area	Region	Developed_region	Least_developed_country	Sub_Saharan_Africa
108	Burundi	B R	Both sexes	1990	333110	Asia	Southern Asia	No	Yes	No
174	Comoros	B	Both sexes	1990	14079	Europe	Southern Europe	Yes	No	No
262	Djibouti	B R	Both sexes	1990	122221	Africa	Northern Africa	No	No	No
232	Eritrea	I	Both sexes	1990	11848	Oceania	Polynesia	No	No	No
231	Ethiopia	B R	Both sexes	1990	1155390	Europe	Southern Europe	Yes	No	No
404	Kenya	B R	Both sexes	1990	297292	Africa	Middle Africa	No	Yes	Yes
450	Madagascar	C	Both sexes	1990	23917	Latin America and the Caribbean	Caribbean	No	No	No
454	Malawi	B R	Both sexes	1990	1127724	Latin America and the Caribbean	Caribbean	No	No	No
480	Mauritius	C	Both sexes	1990	3613	Latin America and the Caribbean	South America	No	No	No
175	Mayotte	B	Both sexes	1990	15229	Asia	Western Asia	No	No	No

Step 7: Split the dataset by unique year, since I want to analyze the data regarding the migrant

stock in different years. The following image is the example after the splitting by year.

Table1.1 - international migrant stock in 1990

```
In [442]: # Split year 1990 international migrant stock from the table 1 data
          table1_1990 = table1[table1['Year'].astype(int) == 1990]
          table1_1990.head(5)
```

Out[442]:

Country Code	Country	Data type	Gender	Year	international migrant	Major_area	Region	Developed_region	Least_developed_country	Sub_Saharan_Africa
108	Burundi	B R	Both sexes	1990	333110	Asia	Southern Asia	No	Yes	No
174	Comoros	B	Both sexes	1990	14079	Europe	Southern Europe	Yes	No	No
262	Djibouti	B R	Both sexes	1990	122221	Africa	Northern Africa	No	No	No
232	Eritrea	I	Both sexes	1990	11848	Oceania	Polynesia	No	No	No
231	Ethiopia	B R	Both sexes	1990	1155390	Europe	Southern Europe	Yes	No	No

Table1.2 - international migrant stock in 1995

```
In [443]: # Split year 1995 international migrant stock data from table 1 data
          table1_1995 = table1[table1['Year'].astype(int) == 1995]
          table1_1995.head(5)
```

Out[443]:

Country Code	Country	Data type	Gender	Year	international migrant	Major_area	Region	Developed_region	Least_developed_country	Sub_Saharan_Africa
108	Burundi	B R	Both sexes	1995	254853	Asia	Southern Asia	No	Yes	No
174	Comoros	B	Both sexes	1995	13939	Europe	Southern Europe	Yes	No	No
262	Djibouti	B R	Both sexes	1995	99774	Africa	Northern Africa	No	No	No
232	Eritrea	I	Both sexes	1995	12400	Oceania	Polynesia	No	No	No
231	Ethiopia	B R	Both sexes	1995	806904	Europe	Southern Europe	Yes	No	No

<u>Steps of tidying table 4 data</u>

Step 1: Load the data from excel file

Step 2: Drop the meaningless columns -- sort order, notes

Step 3: Drop the region and major area rows (eg. World, Asia, Africa) use drop function with the country code that is greater than 900. (The reason of dropping those rows is I am going to append the major area, region, developed, least developed, and Sub-Saharan Africa columns on the next step)

Step 4: Use a for-loop assign the major area, region, developed, least developed, and

Sub-Saharan Africa columns from annex table to the current table regarding the equal country name.

Step 5: Use melt function let year and target values to be two column names in order to satisfy the tidy data principle 1 which is letting column names to be informative, variable names and not values. The following figure is the result without a split.

Out[425]:

Country Code	Country	Data type	Year	Female migrants proportion	Major_area	Region	Developed_region	Least_developed_country	Sub_Saharan_Africa
108	Burundi	B R	1990	50.987061	Asia	Southern Asia	No	Yes	No
174	Comoros	B	1990	52.290646	Europe	Southern Europe	Yes	No	No
262	Djibouti	B R	1990	47.437838	Africa	Northern Africa	No	No	No
232	Eritrea	I	1990	47.434166	Oceania	Polynesia	No	No	No
231	Ethiopia	B R	1990	47.439047	Europe	Southern Europe	Yes	No	No
...	...	...	...	...	...	...	...	...	...
882	Samoa	B	2015	49.908704	Oceania	Polynesia	No	No	No
772	Tokelau	B	2015	52.156057	Africa	Northern Africa	No	No	No
776	Tonga	B	2015	45.437096	Asia	Western Asia	No	Yes	No
798	Tuvalu	C	2015	44.680851	Africa	Eastern Africa	No	Yes	Yes
876	Wallis and Futuna Islands	B	2015	49.52615	Africa	Eastern Africa	No	No	Yes

1392 rows × 9 columns

Step 6: Split the dataset by unique year, since I want to analyze the data regarding the migrant

stock in different years. The following image is the example after the splitting by year.

Table 4.1 - Female migrants proportion 1990

```
In [426]: table4_1990 = table4[table4['Year'] == '1990']
          table4_1990.head()
```

Out[426]:

Country Code	Country	Data type	Year	Female migrants proportion	Major_area	Region	Developed_region	Least_developed_country	Sub_Saharan_Africa
108	Burundi	B R	1990	50.987061	Asia	Southern Asia	No	Yes	No
174	Comoros	B	1990	52.290646	Europe	Southern Europe	Yes	No	No
262	Djibouti	B R	1990	47.437838	Africa	Northern Africa	No	No	No
232	Eritrea	I	1990	47.434166	Oceania	Polynesia	No	No	No
231	Ethiopia	B R	1990	47.439047	Europe	Southern Europe	Yes	No	No

Table 4.2 - Female migrants proportion 1995

```
In [427]: table4_1995 = table4[table4['Year'] == '1995']
          table4_1995.head()
```

Out[427]:

Country Code	Country	Data type	Year	Female migrants proportion	Major_area	Region	Developed_region	Least_developed_country	Sub_Saharan_Africa
108	Burundi	B R	1995	51.279757	Asia	Southern Asia	No	Yes	No
174	Comoros	B	1995	52.550398	Europe	Southern Europe	Yes	No	No
262	Djibouti	B R	1995	47.405136	Africa	Northern Africa	No	No	No
232	Eritrea	I	1995	47.241935	Oceania	Polynesia	No	No	No
231	Ethiopia	B R	1995	47.438977	Europe	Southern Europe	Yes	No	No

Steps of tidying table 6 data

Step 1: Load the data from excel file

Step 2: Drop the meaningless columns -- sort order, notes

Step 3: Drop the region and major area rows (eg. World, Asia, Africa) use drop function with the country code that is greater than 900. (The reason of dropping those rows is I am going to append the major area, region, developed, least developed, and Sub-Saharan Africa columns on the next step)

Step 4: Use a for-loop assign the major area, region, developed, least developed, and Sub-Saharan Africa columns from annex table to the current table regarding the equal country name.

Step 5: Split the dataset into three tables regarding the different target variables.

Step 6: Use three melt functions let year and target values of those three tables to be two column

names in order to satisfy the tidy data principle 1 which is letting column names to be informative, variable names and not values. The following figures are tables after the split.

Table 6.1 - Estimated refugee stock at mid-year (both sexes)

In [439]:
```
# Seperate the estimated refugee stock at mid-year from the table 6
table6_e = table6.iloc[:,[0,1,2,3,4,5,6,7,8,20,21,22,23,24]]

# Assign Estimated refugee stock as a new column
# in order to let column names to be informative, variable names and not values
table6_estimate = table6_e.melt(id_vars = ['Country', 'Country Code','Data type','Major_area','Region', 'Developed_regi
                        var_name = 'Year', value_name = 'Estimated refugee stock')
table6_estimate.head()
```

Out[439]:

	Country	Country Code	Data type	Major_area	Region	Developed_region	Least_developed_country	Sub_Saharan_Africa	Year	Estimated refugee stock
0	Burundi	108	B R	Asia	Southern Asia	No	Yes	No	1990	267929
1	Comoros	174	B	Europe	Southern Europe	Yes	No	No	1990	0
2	Djibouti	262	B R	Africa	Northern Africa	No	No	No	1990	54508
3	Eritrea	232	I	Oceania	Polynesia	No	No	No	1990	0
4	Ethiopia	231	B R	Europe	Southern Europe	Yes	No	No	1990	741965

Table 6.2 - Refugees as a percentage of the international migrant stock

In [440]:
```
# Seperate Refugees as a percentage of the international migrant stock from the table 6
table6_r = table6.iloc[:,[0,1,2,9,10,11,12,13,14,20,21,22,23,24]]

# Assign Refugees proportion as a new column
# in order to let column names to be informative, variable names and not values
table6_proportion = table6_r.melt(id_vars = ['Country', 'Country Code','Data type','Major_area','Region', 'Developed_re
                        var_name = 'Year', value_name = 'Refugees proportion')

table6_proportion.head()
```

Out[440]:

	Country	Country Code	Data type	Major_area	Region	Developed_region	Least_developed_country	Sub_Saharan_Africa	Year	Refugees proportion
0	Burundi	108	B R	Asia	Southern Asia	No	Yes	No	1990	80.43259
1	Comoros	174	B	Europe	Southern Europe	Yes	No	No	1990	0
2	Djibouti	262	B R	Africa	Northern Africa	No	No	No	1990	44.597901
3	Eritrea	232	I	Oceania	Polynesia	No	No	No	1990	0
4	Ethiopia	231	B R	Europe	Southern Europe	Yes	No	No	1990	64.21771

Table 6.3 - Annual rate of change of the refugee stock

```
In [441]: # Seperate Annual rate of change of the refugee stock from the table 6
          table6_a = table6.iloc[:,[0,1,2,15,16,17,18,19,20,21,22,23,24]]

          # Assign Annual rate of change as a new column
          # in order to let column names to be informative, variable names and not values
          table6_annual = table6_a.melt(id_vars = ['Country', 'Country Code','Data type','Major_area','Region', 'Developed_region
                                  var_name = 'Year', value_name = 'Annual rate of change')
          table6_annual.head()
```

Out[441]:

	Country	Country Code	Data type	Major_area	Region	Developed_region	Least_developed_country	Sub_Saharan_Africa	Year	Annual rate of change
0	Burundi	108	B R	Asia	Southern Asia	No	Yes	No	1990-1995	-3.390926
1	Comoros	174	B	Europe	Southern Europe	Yes	No	No	1990-1995	..
2	Djibouti	262	B R	Africa	Northern Africa	No	No	No	1990-1995	-9.763426
3	Eritrea	232	I	Oceania	Polynesia	No	No	No	1990-1995	..
4	Ethiopia	231	B R	Europe	Southern Europe	Yes	No	No	1990-1995	-5.505717

Discussion

According to this project I learned the three tidy data principles which are that each variable must have its own column, each observation must have its own row and each value must have its own cell. In order to achieve those rules, I gain knowledge about having to use the melt function, for-loop, assign function, etcetera.

For all tables, I drop the major area, region, or area of destination and append the major area, region, developed, least developed, and Sub-Saharan Africa columns from the annex table based on the same country code, since I want to have a clear table to be easier to find the countries that belong to a specific area. Also, I split table 1 to 5 by distinguishing years or range of years for better analysis of the migrant stock of different time zones. Table 6 has three target variables, and they have different time ranges, so I separate the table into three regarding those target columns.

Overall, I achieved my target goal which is to have clear identification information for each country and the migrant stock variables. However, I did not work on the missing and zero values since they are not part of tidy data principles. Moreover, I append the classification of countries and areas by major area and region from the annex sheet instead of building functions

and classifying the areas and regions based on the data from the original tables, but I do not know if this follows the tidy data rules.

Conclusion

Finally, using the functions I learned in the course, I completed the data wrangling for all 6 tables in accordance with the tidy data principles and obtained my desired results, which included major area, region, developed, least developed, and Sub-Saharan Africa columns for each table of migrant stock and split them by year or target variables.