Introduction

Prior to commencing the midterm assignment for the data file "UN_MigrantStockTotal_2015", it was important for me to understand the information being presented in each of the spreadsheets and understand the difference in the information being reported in each of the tables.

It is evident that the spreadsheets share commonalities in structure and header/row information. However, the information differs in terms of the data being reported. It was also important for me to gain an understanding of the trends related to International migration. It is noted from the results that International migration increased dramatically from 1990 (152 million) to 2015 (244 million), an increase of 92 million. In addition, in researching this particular study, I found a few summaries that provided further highlights and insight regarding which countries the majority of migrants came from and the amount of female versus male migrants.

Having knowledge and information regarding the spreadsheets certainly helped with comprehension of the data and also eased my stress level as the tables visually appear intimidating. It is also important to note that there are differences in the values being used. Some tables are reporting unit value amounts while others are reporting based on percentages.

Methods/Results

I started the project by saving the data file "UN_MigrantStockTotal_2015" in my personal Google Drive and set up the access directly to this folder. In addition, I set-up the following libraries:

"Import numpy as np" and "Import pandas as pd".

In order to access the "UN_MigrantStockTotal_2015.xlsx" file, the following code was used:

df=pd. ExcelFile('/content/drive/MyDrive/UN_MigrantStockTotal_2015.xlsx')

Initially, I was using code related to a csv file and after a few failed attempts realized that the code should be updated to access an Excel File. The failed attempts allow you to gain knowledge of the different types of error messages and become familiar with how to resolve them.

In addition, I called up all the spreadsheets contained in the data file using the following code:

df.sheet_names

Having a list of the worksheets contained in the data frame assisted me in organizing my work flow.

Output is as follows:

['CONTENTS'

'TABLE 1',

'TABLE 2',

'TABLE 3',

'TABLE 4',

'TABLE 5',

'TABLE 6',

'ANNEX',

'NOTES']

My approach for this assignment was to tackle each worksheet contained in the data frame individually as this assisted me in focusing on the individual tables and allowed me to compartmentalize the individual data cleanup tasks needed for each of the spreadsheets, which ended up being quite similar.

Data Cleaning Steps:

Following Principle 1 of Tidy Data "Column headers should be variable names, not values". The first task was to review the column headings and confirm adherence to this principle. The spreadsheets showed that the column headings in the spreadsheets contain values. Using Table 1 as an example, sex/gender and the year are all variables contained in the column header which is against the Tidy Data rules. This variable information should be contained inside the dataset and not reside in the header.

International migrant stock at mid-year (both sexes)						International migrant stock at mid-year (male)						International migrant stock at mid-year (female)					
1990	1995	2000	2005	2010	2015	1990	1995	2000	2005	2010	2015	1990	1995	2000	2005	2010	2015

Once each of the tables was called up individually the data cleaning process commenced. (Please note that a similar process was used for each of the Tables and my write-up will use Table 1 as an example).

Table1=pd.read_excel(df, sheet_name="Table 1")

table1

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	...	Unnamed: 13	Unnamed: 14	Unnamed: 15
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	United Nations	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	Population Division	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
275	261	Samoa	NaN	882	B	3357	4694	5998	5746	5122.0	...	3101	2940	2594.0
276	262	Tokelau	NaN	772	B	270	266	262	258	429.0	...	144	133	206.0
277	263	Tonga	NaN	776	B	2911	3274	3684	4301	5022.0	...	1981	2328	2727.0
278	264	Tuvalu	NaN	798	C	318	263	217	183	154.0	...	121	101	85.0
279	265	Wallis and Futuna Islands	NaN	876	B	1402	1680	2015	2365	2776.0	...	1018	1194	1401.0

In order to better visualize the data in the table, I set the visual to Line 13 which displayed the information near the top of the spreadsheet where the real data resides. This made it easier working with the data.

#Removing the “Header”

table1=table1[13:]

table1

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	...	Unnamed: 13	Unnamed: 14
13	Sort\norder	Major area, region, country or area of destina...	Notes	Country code	Type of data (a)	International migrant stock at mid-year (both ...	NaN	NaN	NaN	NaN	...	NaN	NaN
14	NaN	NaN	NaN	NaN	NaN	1990	1995	2000	2005	2010.0	...	2000	2005
15	1	WORLD	NaN	900	NaN	152563212	160801752	172703309	191269100	221714243.0	...	87884839	97866674
16	2	Developed regions	(b)	901	NaN	82378628	92306854	103375363	117181109	132560325.0	...	50536796	57217777
17	3	Developing regions	(c)	902	NaN	70184584	68494898	69327946	74087991	89153918.0	...	37348043	40648897
...	...	...	...	...	...	...	...	...	...	...	...	...	...
275	261	Samoa	NaN	882	B	3357	4694	5998	5746	5122.0	...	3101	2940
276	262	Tokelau	NaN	772	B	270	266	262	258	429.0	...	144	133
277	263	Tonga	NaN	776	B	2911	3274	3684	4301	5022.0	...	1981	2328
278	264	Tuvalu	NaN	798	C	318	263	217	183	154.0	...	121	101
279	265	Wallis and Futuna Islands	NaN	876	B	1402	1680	2015	2365	2776.0	...	1018	1194

In order to move the values into the data set, I determined that the first step would be to create New Column Names. Creating new column headings would assist in the later steps of bringing the vales currently stored in the column headers into the data rows and columns and would also assist in the separating of values contained in the columns.

“my_columns=” was used to begin the process of defining my New Column names and “sort\norder” to order the new column headings as listed below and in accordance to how the data in the table is set.

```
#New Column Names

my_columns=["sort\norder","Major area, region, country or area of destination", "Notes", "Country Code", "Type of data (a)",
            "Both sexes, 1990", "Both sexes, 1995", "Both sexes, 2000", "Both sexes, 2005", "Both sexes, 2010", "Both sexes, 2015",
            "Male, 1990", "Male, 1995", "Male, 2000", "Male, 2005", "Male, 2010", "Male, 2015", "Female, 1990", "Female, 1995",
            "Female, 2000", "Female, 2005", "Female, 2010", "Female, 2015"]
my_columns
```

```
['sort\norder',
 'Major area, region, country or area of destination',
 'Notes',
 'Country Code',
 'Type of data (a)',
 'Both sexes, 1990',
 'Both sexes, 1995',
 'Both sexes, 2000',
 'Both sexes, 2005',
 'Both sexes, 2010',
 'Both sexes, 2015',
 'Male, 1990',
 'Male, 1995',
 'Male, 2000',
 'Male, 2005',
 'Male, 2010',
 'Male, 2015',
 'Female, 1990',
 'Female, 1995',
 'Female, 2000',
 'Female, 2005',
 'Female, 2010',
 'Female, 2015']
```

table2.columns=my_columns

table2

The output is as follows:

	sort\norder	region, country or area of destination	Notes	Country Code	Type of data (a)	Both sexes, 1990	Both sexes, 1995	Both sexes, 2000	Both sexes, 2005	Both sexes, 2010	...	Male, 2000	Male, 2005	Male
13	Sort\norder	Major area, region, country or area of destina...	Notes	Country code	Type of data (a)	International migrant stock at mid-year (both ...	NaN	NaN	NaN	NaN	...	NaN	NaN	
14	NaN	NaN	NaN	NaN	NaN	1990	1995	2000	2005	2010.0	...	2000	2005	
15	1	WORLD	NaN	900	NaN	152563212	160801752	172703309	191269100	221714243.0	...	87884839	97866674	1146
16	2	Developed regions	(b)	901	NaN	82378628	92306854	103375363	117181109	132560325.0	...	50536796	57217777	640
17	3	Developing regions	(c)	902	NaN	70184584	68494898	69327946	74087991	89153918.0	...	37348043	40648897	505
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
275	261	Samoa	NaN	882	B	3357	4694	5998	5746	5122.0	...	3101	2940	
276	262	Tokelau	NaN	772	B	270	266	262	258	429.0	...	144	133	
277	263	Tonga	NaN	776	B	2911	3274	3684	4301	5022.0	...	1981	2328	
278	264	Tuvalu	NaN	798	C	318	263	217	183	154.0	...	121	101	
279	265	Wallis and Futuna Islands	NaN	876	B	1402	1680	2015	2365	2776.0	...	1018	1194	

Male, 2010	Male, 2015	Female, 1990	Female, 1995	Female, 2000	Female, 2005	Female, 2010	Female, 2015
NaN	NaN	International migrant stock at mid-year (female)	NaN	NaN	NaN	NaN	NaN
2010.0	2015.0	1990	1995	2000	2005	2010.0	2015.0
114613714.0	126115435.0	74815702	79064275	84818470	93402426	107100529.0	117584801.0
64081077.0	67618619.0	42115231	47214055	52838567	59963332	68479248.0	72863336.0
50532637.0	58496816.0	32700471	31850220	31979903	33439094	38621281.0	44721465.0
...	...	...	...	...	...	...	...
2594.0	2469.0	1586	2243	2897	2806	2528.0	2460.0
206.0	233.0	120	119	118	125	223.0	254.0
2727.0	3127.0	1423	1556	1703	1973	2295.0	2604.0
85.0	78.0	138	115	96	82	69.0	63.0
1401.0	1438.0	676	821	997	1171	1375.0	1411.0

The next step was the process of removing Columns for fields that contain information related to Notes and Type of data. Removing these columns assisted in the visualization of the table to be cleaner without compromising the substance of the data contained in the table.

#Removing the Columns

table1=table1.iloc[2:, 1:]

table1=table1.drop([“Notes”, “Type of data (a)”], axis=1)

table1

The output, once this code is run, is as follows: Both the “Notes” and “Type of data (a)” columns are removed and no longer displayed in the table.

	Major area, region, country or area of destination	Country Code	Both sexes, 1990	Both sexes, 1995	Both sexes, 2000	Both sexes, 2005	Both sexes, 2010	Both sexes, 2015	Male, 1990	Male, 1995	Male, 2000	
15	WORLD	900	152563212	160801752	172703309	191269100	221714243.0	243700236.0	77747510	81737477	87884839	9
16	Developed regions	901	82378628	92306854	103375363	117181109	132560325.0	140481955.0	40263397	45092799	50536796	5
17	Developing regions	902	70184584	68494898	69327946	74087991	89153918.0	103218281.0	37484113	36644678	37348043	4
18	Least developed countries	941	11075966	11711703	10077824	9809634	10018128.0	11951316.0	5843107	6142712	5361902	
19	Less developed regions excluding least develop...	934	59105261	56778501	59244124	64272611	79130668.0	91262036.0	31641006	30501966	31986141	3

In following the next principles of Tidy Data, the next principle indicates that “Multiple variables should not be stored in one column”. As noted above, multiple variables are stored in one column such as year and gender and total of both male and female “International migrant Stock at Mid-Year” information. In addition, another principle indicates that variables should not be stored in both rows and columns. As seen in this Table, we have variables that appear in the columns as well as the rows.

In order to satisfy these principles, the “Melt” function was used to bring the variable information about gender and year into the data.

The following code was used:

```
table1_tidy=table1.melt(id_vars=['Major area, region, country or area of destination', 'Country Code'],
  value_vars=["Both sexes, 1990", "Both sexes, 1995", "Both sexes, 2000", "Both sexes, 2005",
  "Both sexes, 2010", "Both sexes, 2015", "Male, 1990", "Male, 1995", "Male, 2000", "Male, 2005", "Male, 2010",
  "Male, 2015", "Female, 1990",
  "Female, 1995", "Female, 2000", "Female, 2005", "Female, 2010", "Female, 2015"],
value_name="International migrant stock at mid-year",
var_name="Sex, Year")
table1_tidy
```

The “Melt” function allows you to change the format of the DataFrame format from wide to long and therefore allows the values to display in the data cells.

You must specify the id_vars, value_vars, value_name, var_name. This step satisfies the principle that the columns should not contain values. The column header is now just column names and does not include pertinent data.

The output is as follows:

	Major area, region, country or area of destination	Country Code	Sex, Year	International migrant stock at mid-year
0	WORLD	900	Both sexes, 1990	152563212
1	Developed regions	901	Both sexes, 1990	82378628
2	Developing regions	902	Both sexes, 1990	70184584
3	Least developed countries	941	Both sexes, 1990	11075966
4	Less developed regions excluding least develop...	934	Both sexes, 1990	59105261
...	...	...	...	...
4765	Samoa	882	Female, 2015	2460.0
4766	Tokelau	772	Female, 2015	254.0
4767	Tonga	776	Female, 2015	2604.0
4768	Tuvalu	798	Female, 2015	63.0
4769	Wallis and Futuna Islands	876	Female, 2015	1411.0

In order to ensure that we do not have more than one variable stored in one column, the next step was to create a separate the cell which contains “Sex” and another cell which contains the “Year”. In order to split the variables “Sex” and “Year”, the function “str.split” function was used. This function splits the two strings into separate columns.

The code is as follows:

```
#Seperating Columns into 2
table1_tidy[["Sex", "Year"]]=table1_tidy['Sex, Year'].str.split(',',expand=True)
table1_tidy=table1_tidy.drop("Sex, Year", axis=1)
table1_tidy
```

The output is as follows:

	Major area, region, country or area of destination	Country Code	International migrant stock at mid-year	Sex	Year
0	WORLD	900	152563212	Both sexes	1990
1	Developed regions	901	82378628	Both sexes	1990
2	Developing regions	902	70184584	Both sexes	1990
3	Least developed countries	941	11075966	Both sexes	1990
4	Less developed regions excluding least develop...	934	59105261	Both sexes	1990
...	...	...	...	...	...
4765	Samoa	882	2460.0	Female	2015
4766	Tokelau	772	254.0	Female	2015
4767	Tonga	776	2604.0	Female	2015
4768	Tuvalu	798	63.0	Female	2015
4769	Wallis and Futuna Islands	876	1411.0	Female	2015

4770 rows × 5 columns

This satisfies the principle that multiple variables should not be stored in one column.

This exercise has reinforced the Tidy Data principles. In addition, learning how to work with the code was challenging but yet rewarding once it finally worked. I encountered a great deal of syntax issues and realized how important naming conventions are in prior steps and ensuring your names are consistent throughout. I spent all of the time focusing on the data contained on the left side of the spreadsheets and should have also considered the data cleaning in the “Major area, region, country or area of destination”column.

In conclusion, as experience and knowledge is gained in both the use of Python code and gaining more exposure to data cleaning, there is definitely opportunity of improving the code by reducing and simplifying the steps taken. Overall, I found the Melt () function beneficial and a valuable tool.