INF1340 Midterm Writeup

This project in this document is mainly cleaning up the datasets from the United Nations that talks about trends in international migrant stock based on tidy data principles by using Python. In the following, I'll discuss the logic and methods have been applied to take actions and present the results.

```
[ ]  1  import numpy as np
     2  import pandas as pd
```

```
[ ]  1  filename = 'UN_MigrantStockTotal_2015.xlsx'
     2  #read all the data into a single dataframe
     3  dfs = pd.read_excel(filename, sheet_name=None, header=[15])
```

```
[ ]  1  dfs.keys()
```

```
dict_keys(['CONTENTS', 'Table 1', 'Table 2', 'Table 3', 'Table 4', 'Table 5', 'Table 6', 'ANNEX', 'NOTES'])
```

The very beginning of this project is to ask Python to read the excel sheets by using a *"read_excel"* function, come out data frames. In order to make the data frames easier to understand, I ignored row 1 to 14, made the row 15 to be the header, and made the rest of data to be the values.

I used the *"keys"* function to see the tables we have in the excel sheet and was going to clean each table one by one.

```
1  table1 = dfs['Table 1']
2  table1.head(5)
```

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	1990	1995	2000	2005	2010	...	2000.1	2005.1	2010.1	2015.1	1990.2
0	1	WORLD	NaN	900	NaN	152563212	160801752	172703309	191269100	221714243	...	87884839	97866674	114613714	126115435	74815702
1	2	Developed regions	(b)	901	NaN	82378628	92306854	103375363	117181109	132560325	...	50536796	57217777	64081077	67618619	42115231
2	3	Developing regions	(c)	902	NaN	70184584	68494898	69327946	74087991	89153918	...	37348043	40648897	50532637	58496816	32700471
3	4	Least developed countries	(d)	941	NaN	11075966	11711703	10077824	9809634	10018128	...	5361902	5383009	5462714	6463217	5236216
4	5	Less developed regions excluding least develop...	NaN	934	NaN	59105261	56778501	59244124	64272611	79130668	...	31986141	35265888	45069923	52033599	27464255

5 rows × 23 columns

I named data frame "Table 1" to be table1 and used the *"head"* function to see the first 5 rows.

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	1990	1995	2000	2005	2010	...	2000.1	
0	1	WORLD	NaN	900	NaN	152563212.0	160801752.0	172703309.0	191269100.0	221714243	...	87884839.0	97
1	2	Developed regions	(b)	901	NaN	82378628.0	92306854.0	103375363.0	117181109.0	132560325	...	50536796.0	57
2	3	Developing regions	(c)	902	NaN	70184584.0	68494898.0	69327946.0	74087991.0	89153918	...	37348043.0	40
3	4	Least developed countries	(d)	941	NaN	11075966.0	11711703.0	10077824.0	9809634.0	10018128	...	5361902.0	5
4	5	Less developed regions excluding least develop...	NaN	934	NaN	59105261.0	56778501.0	59244124.0	64272611.0	79130668	...	31986141.0	35

5 rows × 23 columns

Then, I noticed that there were some ".." values in the data frame, it looked like missing values due to the excel format. In order to make the data frames more organized, I replaced these missing values to "NaN" by using the *"replace"* function.

```
[ ]  1  #dropping columns with no meanings#
     2  table1.drop(columns=['Unnamed: 0','Unnamed: 2'], inplace=True)
```

```
     1  #rename columns#
     2  table1.columns=['Destination', 'CountryCode', 'Data Type', '1990B',
     3                  '1995B', '2000B',
     4                  '2005B', '2010B', '2015B', '1990M',
     5                  '1995M', '2000M',
     6                  '2005M', '2010M', '2015M', '1990F',
     7                  '1995F', '2000F',
     8                  '2005F', '2010F', '2015F']
```

By noticing that columns "Unnamed: 0" and "Unnamed: 2" did not have specific meanings, and were missing significant values, I dropped these two columns.

On the next step, according to *tidy data principle #1: column names need to be informative, variable names and not values*, I renamed columns to better explain the data with "Destination", "CountryCode", and "Data type". I combined the gender with the year and will split them later. Here, "B" stands for "both sexes", "M" stands for "male" and "F" stands for "female".

	Destination	CountryCode	Data Type	1990B	1995B	2000B	2005B	2010B	2015B	1990M	...	2005M	
0	WORLD	900	NaN	152563212.0	160801752.0	172703309.0	191269100.0	221714243	243700236	77747510.0	...	97866674.0	11
1	Developed regions	901	NaN	82378628.0	92306854.0	103375363.0	117181109.0	132560325	140481955	40263397.0	...	57217777.0	6
2	Developing regions	902	NaN	70184584.0	68494898.0	69327946.0	74087991.0	89153918	103218281	37484113.0	...	40648897.0	5
3	Least developed countries	941	NaN	11075966.0	11711703.0	10077824.0	9809634.0	10018128	11951316	5843107.0	...	5383009.0	
4	Less developed regions excluding least develop...	934	NaN	59105261.0	56778501.0	59244124.0	64272611.0	79130668	91262036	31641006.0	...	35265888.0	4

By noticing that the Destination column had both continents and countries, and I wanted it to be clearer, I added one "Continent" column by using the *lambda* function. In order to get all the continents, I filtered all the continents "Africa", "Asia", "Europe", "Latin America and the Caribbean", "Northern America", "Oceania" by using a *loop*.

Then, refer to *tidy data principle #1*, I used "melt" function to make the row of all "Year + Gender" to be the column "Year", and put all values into the "Count" column.

	Destination	CountryCode	Data Type	Continent	Year	Count
0	WORLD	900	NaN		1990B	152563212.0
1	Developed regions	901	NaN		1990B	82378628.0
2	Developing regions	902	NaN		1990B	70184584.0
3	Least developed countries	941	NaN		1990B	11075966.0
4	Less developed regions excluding least develop...	934	NaN		1990B	59105261.0
5	Sub-Saharan Africa	947	NaN		1990B	14690319.0
6	Africa	903	NaN	Africa	1990B	15690623.0
7	Eastern Africa	910	NaN	Africa	1990B	5964031.0
8	Burundi	108	B R	Africa	1990B	333110.0
9	Comoros	174	B	Africa	1990B	14079.0

Next, according to *tidy data principle #2: each column needs to consist of one and only one variable,* I used the *lambda* function to split the "Year" column into two columns "Year" and "Gender".

Then, I replaced the values in the "Gender" column from "B" to "both sex", "M" to "male, and "F" to "female".

	Destination	CountryCode	Data Type	Continent	Year	Count	Gender
0	WORLD	900	NaN		1990	152563212.0	both sex
1	Developed regions	901	NaN		1990	82378628.0	both sex
2	Developing regions	902	NaN		1990	70184584.0	both sex
3	Least developed countries	941	NaN		1990	11075966.0	both sex
4	Less developed regions excluding least develop...	934	NaN		1990	59105261.0	both sex
...	...	...	...	...	...	...	...
4765	Samoa	882	B	Oceania	2015	2460.0	female
4766	Tokelau	772	B	Oceania	2015	254.0	female
4767	Tonga	776	B	Oceania	2015	2604.0	female
4768	Tuvalu	798	C	Oceania	2015	63.0	female
4769	Wallis and Futuna Islands	876	B	Oceania	2015	1411.0	female

4770 rows × 7 columns

Finally, according *to tidy data principle #3: variables need to be in cells, not rows and columns*, I moved "both sex", "female" and "male to be new columns by using *pivot table* function.

	Gender	Destination	CountryCode	Data Type	Continent	Year	both sex	female	male
0		Afghanistan	4	B	Asia	1990	57686.0	25128.0	32558.0
1		Afghanistan	4	B	Asia	1995	71522.0	32417.0	39105.0
2		Afghanistan	4	B	Asia	2000	75917.0	33069.0	42848.0
3		Afghanistan	4	B	Asia	2005	87300.0	38026.0	49274.0
4		Afghanistan	4	B	Asia	2010	102246.0	44537.0	57709.0
...		...	...	...	...	...	...	...	...
1372		Zimbabwe	716	B R	Africa	1995	431226.0	185214.0	246012.0
1373		Zimbabwe	716	B R	Africa	2000	410041.0	176198.0	233843.0
1374		Zimbabwe	716	B R	Africa	2005	392693.0	168723.0	223970.0
1375		Zimbabwe	716	B R	Africa	2010	397891.0	170924.0	226967.0
1376		Zimbabwe	716	B R	Africa	2015	398866.0	171487.0	227379.0

1377 rows × 8 columns

For Table 2, Table 3, Table 4 and Table 5, I used the same logic and methods to clean the table, and there are the results following.

Table 2

```
1  tidy_table2
```

	Gender	Destination	CountryCode	Year	both sex	female	male
0		Afghanistan	4	1990	12067.570	5887.736	6179.834
1		Afghanistan	4	1995	16772.522	8090.080	8682.442
2		Afghanistan	4	2000	19701.940	9555.403	10146.537
3		Afghanistan	4	2005	24399.948	11783.622	12616.326
4		Afghanistan	4	2010	27962.207	13594.574	14367.633
...		...	...	...	...	...	...
1585		Zimbabwe	716	1995	11683.136	5877.504	5805.632
1586		Zimbabwe	716	2000	12499.981	6280.133	6219.848
1587		Zimbabwe	716	2005	12984.418	6548.180	6436.238
1588		Zimbabwe	716	2010	13973.897	7068.861	6905.036
1589		Zimbabwe	716	2015	15602.751	7915.194	7687.557

1590 rows × 6 columns

Table 3

	Gender	Destination	CountryCode	Type of data	Year	both sex	female	male
0	Afghanistan	4	B	1990	0.478025	0.426785	0.526843
1	Afghanistan	4	B	1995	0.426424	0.400701	0.450392
2	Afghanistan	4	B	2000	0.385328	0.346076	0.422292
3	Afghanistan	4	B	2005	0.357788	0.322702	0.390557
4	Afghanistan	4	B	2010	0.365658	0.327609	0.401660
...	...	...	...	...	...	...	...
1368	Zimbabwe	716	B R	1995	3.691012	3.151236	4.237471
1369	Zimbabwe	716	B R	2000	3.280333	2.805641	3.759626
1370	Zimbabwe	716	B R	2005	3.024340	2.576640	3.479828
1371	Zimbabwe	716	B R	2010	2.847388	2.417985	3.286978
1372	Zimbabwe	716	B R	2015	2.556383	2.166555	2.957754

1373 rows × 7 columns

Table 4

	Gender	Destination	CountryCode	Type of data	Year	female
0	Afghanistan	4	B	1990	43.559963
1	Afghanistan	4	B	1995	45.324516
2	Afghanistan	4	B	2000	43.559414
3	Afghanistan	4	B	2005	43.557847
4	Afghanistan	4	B	2010	43.558672
...	...	...	...	...	...
1372	Zimbabwe	716	B R	1995	42.950564
1373	Zimbabwe	716	B R	2000	42.970825
1374	Zimbabwe	716	B R	2005	42.965625
1375	Zimbabwe	716	B R	2010	42.957493
1376	Zimbabwe	716	B R	2015	42.993637

1377 rows × 5 columns

Table 5

	Gender	Destination	CountryCode	Type of data	Year	both sex	female	male
0		Afghanistan	4	B	1990-1995	4.299812	5.094004	3.664544
1		Afghanistan	4	B	1995-2000	1.192711	0.398266	1.828173
2		Afghanistan	4	B	2000-2005	2.794196	2.793477	2.794752
3		Afghanistan	4	B	2005-2010	3.160624	3.161003	3.160332
4		Afghanistan	4	B	2010-2015	26.379880	28.900067	24.191602
...		...	...	...	...	...	...	...
1140		Zimbabwe	716	B R	1990-1995	-7.480574	-7.890123	-7.166608
1141		Zimbabwe	716	B R	1995-2000	-1.007503	-0.998071	-1.014607
1142		Zimbabwe	716	B R	2000-2005	-0.864580	-0.867001	-0.862757
1143		Zimbabwe	716	B R	2005-2010	0.262999	0.259214	0.265850
1144		Zimbabwe	716	B R	2010-2015	0.048948	0.065769	0.036272

1145 rows × 7 columns

In table 6, there are three different types of datasets in the same observational unit, according to *tidy data principle #4: each table column needs to have a singular data type*, so I split table 6 into three sub-tables. They are table6_E, table6_R and table6_A.

Table6_E

	Destination	CountryCode	Type of data	Year	Estimated	Gender
0	WORLD	900	NaN	1990	18836571.0	both sexes
1	Developed regions	901	NaN	1990	2014564.0	both sexes
2	Developing regions	902	NaN	1990	16822007.0	both sexes
3	Least developed countries	941	NaN	1990	5048391.0	both sexes
4	Less developed regions excluding least develop...	934	NaN	1990	11773616.0	both sexes
...	...	...	...	...	...	...
1585	Samoa	882	B	2015	0.0	both sexes
1586	Tokelau	772	B	2015	0.0	both sexes
1587	Tonga	776	B	2015	0.0	both sexes
1588	Tuvalu	798	C	2015	0.0	both sexes
1589	Wallis and Futuna Islands	876	B	2015	0.0	both sexes

1590 rows × 6 columns

Table6_R

	Destination	CountryCode	Type of data	Year	Percentage
0	WORLD	900	NaN	1990	12.346732
1	Developed regions	901	NaN	1990	2.445494
2	Developing regions	902	NaN	1990	23.968236
3	Least developed countries	941	NaN	1990	45.565880
4	Less developed regions excluding least develop...	934	NaN	1990	19.919743
...	...	...	...	...	...
1585	Samoa	882	B	2015	0.000000
1586	Tokelau	772	B	2015	0.000000
1587	Tonga	776	B	2015	0.000000
1588	Tuvalu	798	C	2015	0.000000
1589	Wallis and Futuna Islands	876	B	2015	0.000000

1590 rows × 5 columns

Table6_A

	Destination	CountryCode	Type of data	Year	Annual rate
0	WORLD	900	NaN	1990-1995	-2.123497
1	Developed regions	901	NaN	1990-1995	9.388424
2	Developing regions	902	NaN	1990-1995	-2.839417
3	Least developed countries	941	NaN	1990-1995	-0.680327
4	Less developed regions excluding least develop...	934	NaN	1990-1995	-4.383600
...	...	...	...	...	...
1320	Samoa	882	B	2010-2015	NaN
1321	Tokelau	772	B	2010-2015	NaN
1322	Tonga	776	B	2010-2015	NaN
1323	Tuvalu	798	C	2010-2015	NaN
1324	Wallis and Futuna Islands	876	B	2010-2015	NaN

1325 rows × 5 columns

In this project, for the first time I have worked on a real-life dataset with massive data. By using tidy data principles, I am able to clean the dataset in a standard way and make the analyst's work much easier to observe and extract needed variables. However, there are five principles in total, but I did not apply the principle #5 because there are different observational units in table 6, so it's not necessary to apply it into this dataset and joint them in one table. After discussion my classmates, I realized everyone has their own logic to deal with the data, and analysts have their own freedom to decide on how to clean the data.

Also, during the data cleaning process, I truly thought "melt" and "pivot table" are two super useful functions for data scientists to use. They can basically be applied to all datasets that need to be cleaned up. In addition, it would be more difficult for data scientist to deal with context data such as Destination. In this project, I extracted "Continent" from "Destination". If I want to organize the destination column in a more accurate way, I'll probably need more steps to deal with that.