Introduction

For the purpose of this assignment, I have cleaned a data set from UN's statistics. The process includes firstly identify the problematic sectors for structural improvements, then splitting the data frame to edit following tidy data principles. There are total six tables (excluding contents, annex, and notes) in the excel file. In order to clean the tables, I have created six Jupiter notebooks for each table. The key highlights of the results including sorting the column "Major area, region, country or area of destination" into " regions, countries, and area of destination." On the other hand, I've also melted the year column and separated year data from gender. Overall, the outcome of the dataset closely followed principle one and two from tidy data principles.

Methods and Tools

It is important to initially identify the problem with the current data frame to seek to improve. For this purpose, three major issues with the existing file were noticed. For one, the column titled "Major area, region, country or area of destination" contains too many different values. For instance, "Developed regions," Burundi," and "Sub-Saharan Africa" are three different concepts and should not be grouped as a whole. Secondly, the year column does not exist, instead, the years are columns which is a violation of tidy data principle one "Column names need to be informative, variable names and not values." Thirdly, the blocks of years are jammed under the column that contains gender information, which is a violation to tidy principle two "each column needs to consist of one and only one variable." It is important to keep these recognitions in mind while starting to clean table 1.

Table 1

Firstly, I removed the dropped 14 rows of unnecessary information from the data frame (titles and descriptions.) Then, I renamed each column and dropped three axis "Sort,""Notes," and "Data Type" because I think they are not relevant to the task and has many empty entries for most variables (however, they can be easily add back if needed.) A key insight to note at this stage is that I have combined year and gender in the columns, however, they will be separated in the next step. The outcome shows as follows:

[19]:

	Country/Area	Country Code	1990b	1995b	2000b	2005b	2010b	2015b	1990m	1995m	2000m	2005m	2010m	
15	WORLD	900	152563212	160801752	172703309	191269100	221714243.0	243700236.0	77747510	81737477	87884839	97866674	114613714.0	12
16	Developed regions	901	82378628	92306854	103375363	117181109	132560325.0	140481955.0	40263397	45092799	50536796	57217777	64081077.0	6
17	Developing regions	902	70184584	68494898	69327946	74087991	89153918.0	103218281.0	37484113	36644678	37348043	40648897	50532637.0	5
18	Least developed countries	941	11075966	11711703	10077824	9809634	10018128.0	11951316.0	5843107	6142712	5361902	5383009	5462714.0	
19	Less developed regions excluding least develop...	934	59105261	56778501	59244124	64272611	79130668.0	91262036.0	31641006	30501966	31986141	35265888	45069923.0	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
110	Timor-Leste	626	8954	9743	10602	11286	10983.0	10834.0	4450	4763	5095	5582	6441.0	
111	Viet Nam	704	28118	51262	56754	51768	61756.0	72793.0	15230	28187	32825	31106	35971.0	
112	Southern Asia	5501	19436343	15343019	15278020	13722011	14326591.0	14103682.0	10601061	8382298	8414610	7537840	7953646.0	
113	Afghanistan	4	57686	71522	75917	87300	102246.0	382365.0	32558	39105	42848	49274	57709.0	
114	Bangladesh	50	881617	934735	987853	1166700	1345546.0	1422805.0	759452	805196	850941	1007698	1164455.0	

100 rows × 20 columns

Then, I started to identify how to separate the column that says “country/area.” I noticed there are mainly three categories — regions, countries, and areas of destination. Therefore, I decided to split the data frame into three data frames according to the above categorization.

Table 1: #1 Splitting Data Frame for Regions

To split the data frame into regions, I noticed a country code was given to each area and decided to locate the regions I want to include according to their country codes. From observing the data frame, I identified the four regions I want to include in the column “regions” according to their country code which is 901,902,934,941. Next, I created a new data frame that only includes these four country codes. To do so, I wrote codes that suggest the values I want to include must be mathematically and logically equal to the given country codes. In addition, I changed the name of the column from “Country/Area” to “Regions” following pandas documentation. After completing the process, I decided to melt the year columns because tidy data principle one suggests “Column names need to be informative, variable names and not values.” To do so, I looked up pandas documentation and used the melt function it was given, I provided the columns I needed to keep and suggest to melt the year column. After completing the task, I decided to separate the gender from the year column because tidy principle two suggests “each column needs to consist of one and only one variable.” To do so, I used a lambda function that separated them according to their index position. For example, if I wanted to separate “1990b” to years and gender, I would use the lambda function to set the fourth position of this string as the gender and everything before that year. The outcome of splitting the data frame for regions shows as follows.

Out[20]:

	Regions	Country Code	International migrant stock at mid-year	Gender	Year
0	Developed regions	901	82378628	b	1990
1	Developing regions	902	70184584	b	1990
2	Least developed countries	941	11075966	b	1990
3	Less developed regions excluding least develop...	934	59105261	b	1990
4	Developed regions	901	92306854	b	1995
...	...	...	...	...	...
67	Less developed regions excluding least develop...	934	34060745.0	f	2010
68	Developed regions	901	72863336.0	f	2015
69	Developing regions	902	44721465.0	f	2015
70	Least developed countries	941	5493028.0	f	2015
71	Less developed regions excluding least develop...	934	39228437.0	f	2015

72 rows × 5 columns

Table 1: #2 Splitting Data Frame for Countries

The next step is to create a new data frame that only shows the data from the countries. Similar to the previous step, I initially identified the countries I want to include basing off their country codes, which I find out that all of them are smaller than 900. Next, I created the data frame titled as CountryDf. The following steps are the same as I did for the previous data frame. The outcome shows:

Out[21]:

	Countries	Country Code	International migrant stock at mid-year	Gender	Year
0	Burundi	108	333110	b	1990
1	Comoros	174	14079	b	1990
2	Djibouti	262	122221	b	1990
3	Eritrea	232	11848	b	1990
4	Ethiopia	231	1155390	b	1990
...	...	...	...	...	...
95	Iraq	368	83638	b	1990
96	Israel	376	1632704	b	1990
97	Jordan	400	1146349	b	1990
98	Kuwait	414	1074391	b	1990
99	Lebanon	422	523693	b	1990

100 rows × 5 columns

Table 1: #3 Splitting Data Frame for Area of Destination

The final step for this data frame is to split a data frame that only shows the area of destination. Same as the previous two frames, I first identified the country codes of these areas. However, this step is more complex than the previous two because they are not as simple to locate. To elaborate, most areas are in between code 903 to 954, however, some codes of them are not areas I want to include. To exclude the codes and keep the ones I want, I used a "not equal to" statement to expel the ones I don't want. Once I figured out the task, the rest of the process is the same as I did with the previous two frames. The outcome shows as follow and thus ends my cleaning for the first table from the excel file.

Out[23]:

	Area of Destination	Country Code	International migrant stock at mid-year	Gender	Year
0	Sub-Saharan Africa	947	14690319	b	1990
1	Eastern Africa	910	5964031	b	1990
2	Middle Africa	911	1460530	b	1990
3	Northern Africa	912	2403200	b	1990
4	Southern Africa	913	1392359	b	1990
5	Western Africa	914	4470503	b	1990
6	Asia	935	48142261	b	1990
7	Central Asia	5500	6630683	b	1990
8	Eastern Asia	906	3959345	b	1990
9	South-Eastern Asia	920	2876616	b	1990

Table 2, 3 &4

It is not as challenging to clean table 2, 3, and 4 once I understood the methods I used for cleaning table 1, and the majority of the work I did for these three tables is repetitive as to what I did for table 1. To elaborate, the alternations I did for those three tables is to change the statistic names for each one (for example, the name for table 2 is "Total population at mid-year,") and add or deduct values in the column section. For example, table 4 only has statistics for females compares to other tables that has both sex, male, and female statistics. Therefore, when I complete the first process on importing data frame and renaming the columns, I need to be

mindful of this change. The outcome when I firstly imported table 4 into the notebook looks like the follow image, which one can see the columns are a lot less than the previous tables.

Out[18]:

	Country/Area	Country Code	1990f	1995f	2000f	2005f	2010f	2015f
15	WORLD	900	49.03915	49.16879	49.112244	48.832993	48.305660	48.249769
16	Developed regions	901	51.123977	51.149024	51.113307	51.171501	51.658932	51.866687
17	Developing regions	902	46.592099	46.500135	46.128444	45.134297	43.319780	43.327078
18	Least developed countries	941	47.261155	47.571664	46.826689	45.157406	45.499573	45.942752
19	Less developed regions excluding least develop...	934	46.466684	46.279022	46.009598	45.130768	43.043672	42.984398
...	...	...	...	...	...	...	...	...
110	Timor-Leste	626	50.301541	51.11362	51.94303	50.540493	41.354821	41.083626
111	Viet Nam	704	45.835408	45.01385	42.162667	39.912687	41.753028	42.071353
112	Southern Asia	5501	45.457533	45.367349	44.923426	45.067527	44.483332	45.408724
113	Afghanistan	4	43.559963	45.324516	43.559414	43.557847	43.558672	49.408288
114	Bangladesh	50	13.856924	13.858366	13.859552	13.628353	13.458551	13.325719

Other than that, the outcomes for table2, 3, and 4 looks similar to table 1 in respect to their structures, and all of them has three split data frame titled “Region,” “Country,” and “Areas of Destination.”

Table 5

A noticeable change in table 5 compares to other table is that instead of a specific year, the table showcases data following a period of time (eg. 1990-1995.) In order to figure out how to sort them differently, I imported the data frame the same way I did for other tables, and the outcome looks as the follow:

t[21]:

	Country/Area	Country Code	1990-1995b	1995-2000b	2000-2005b	2005-2010b	2010-2015b	1990-1995m	1995-2000m	2000-2005m	2005-2010m	2010-2015m	1990-1995f	1995-2000f
15	WORLD	900	1.051865	1.428058	2.042124	2.95416	1.890991	1.000922	1.450294	2.151575	3.159228	1.912603	1.104667	1.405044
16	Developed regions	901	2.275847	2.264965	2.50708	2.466343	1.160824	2.265595	2.279583	2.483259	2.265689	1.074685	2.285643	2.250995
17	Developing regions	902	-0.487389	0.241777	1.328107	3.702217	2.929634	-0.45298	0.380246	1.693824	4.352954	2.927058	-0.526904	0.081268
18	Least developed countries	941	1.118175	-3.001139	-0.539636	0.419137	3.526927	1.000073	-2.718952	0.078575	0.293964	3.363629	1.249146	-3.316818
19	Less developed regions excluding least develop...	934	-0.803244	0.850177	1.62934	4.159339	2.852687	-0.733256	0.950231	1.952269	4.90598	2.87349	-0.88418	0.733402
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	Timor-Leste	626	1.688974	1.689872	1.250407	-0.544288	-0.273186	1.359473	1.347639	1.825749	2.862733	-0.180912	2.00928	2.011802
111	Viet Nam	704	12.010796	2.035528	-1.839079	3.528379	3.288573	12.311874	3.046591	-1.075794	2.906247	3.178972	11.649063	0.726827
112	Southern Asia	5501	-4.729682	-0.084908	-2.14828	0.862323	-0.313628	-4.69664	0.076948	-2.200676	1.073895	-0.649813	-4.769399	-0.281573
113	Afghanistan	4	4.299812	1.192711	2.794196	3.160624	26.37988	3.664544	1.828173	2.794752	3.160332	24.191602	5.094004	0.398266
114	Bangladesh	50	1.170107	1.105417	3.328013	2.852413	1.116608	1.169772	1.105141	3.38162	2.891693	1.147282	1.172188	1.107128

Next, just like I did for other groups, I first split the data frame for region. Then, instead of using year as a column, I used period instead. The next thing I did is the same, I firstly melt the period column, then, I separated the period from the gender according to their index position. This is the code I wrote to do so:

```
ng period from gender following tidy data principle #2 each column needs to consist of one and only one variable
=(regionsDf.assign(Gender = lambda x: x.Period.str[9].astype(str), Period = lambda x: x.Period.str[:9].astype(str)))
```

I repeated the steps and competed three separate tables for table 5 — Regions, Countries, and Area of Destination. The following images are the final outcomes:

2]:

	Regions	Country Code	Period	Annual rate of change of the migrant stock	Gender
0	Developed regions	901	1990-1995	2.275847	b
1	Developing regions	902	1990-1995	-0.487389	b
2	Least developed countries	941	1990-1995	1.118175	b
3	Less developed regions excluding least develop...	934	1990-1995	-0.803244	b
4	Developed regions	901	1995-2000	2.264965	b
5	Developing regions	902	1995-2000	0.241777	b
6	Least developed countries	941	1995-2000	-3.001139	b
7	Less developed regions excluding least develop...	934	1995-2000	0.850177	b
8	Developed regions	901	2000-2005	2.50708	b
9	Developing regions	902	2000-2005	1.328107	b

4]:

	Countries	Country Code	Period	Annual rate of change of the migrant stock	Gender
0	Burundi	108	1990-1995	-5.355717	b
1	Comoros	174	1990-1995	-0.199873	b
2	Djibouti	262	1990-1995	-4.058465	b
3	Eritrea	232	1990-1995	0.910748	b
4	Ethiopia	231	1990-1995	-7.179771	b
...	...	...	...	...	...
95	Iraq	368	1990-1995	17.382315	b
96	Israel	376	1990-1995	1.86396	b
97	Jordan	400	1990-1995	5.866269	b
98	Kuwait	414	1990-1995	-3.060279	b
99	Lebanon	422	1990-1995	2.99535	b

100 rows × 5 columns

	Area of Destination	Country Code	Period	Female migrants as a percentage of the international migrant stock	Gender
0	Sub-Saharan Africa	947	1990-1995	0.845374	b
1	Eastern Africa	910	1990-1995	-3.435412	b
2	Middle Africa	911	1990-1995	11.88581	b
3	Northern Africa	912	1990-1995	-2.872903	b
4	Southern Africa	913	1990-1995	-3.114352	b
...	...	...	...	...	...
95	South America	931	2005-2010	2.588966	b
96	Northern America	905	2005-2010	2.428884	b
97	Oceania	909	2005-2010	3.360109	b
98	Australia and New Zealand	927	2005-2010	3.555413	b
99	Melanesia	928	2005-2010	0.344728	b

100 rows × 5 columns

Table 6

Different from the other five tables, for which I generated three split data frames for each table, I split 9 data frames based on table 6. The reason for it is that table 6 contains 3 different statistics, they are "Estimated refugee stock at mid-year (both sexes)," "Refugees as a percentage of the international migrant stock," and "Annual rate of change of the refugee stock." Therefore, one can really consider table 6 as three different tables. One thing to note is that when I imported the data frame, I renamed the columns according to the year/period since this table contains both.

2005	2010	2015	1990	1995	2000	2005	2010	2015	1990-1995	1995-2000	2000-2005	2005-2010	2010-2015
13276733	15370755.0	19577474.0	12.346732	11.103013	9.164736	6.941389	6.932687	8.033424	-2.123497	-3.837069	-5.557223	-0.025089	2.947267

Example from Table 6: Estimated refugee stock at mid-year (both sexes)

The first task is to clean the data under Estimated refugee stock at mid-year (both sexes,) it is not a new mission since it looks similar to the previous tables. Therefore, I followed the rule of splitting it three ways "Region," "Countries," and "Area of Destination." The method I used for this table is similar to what I did for previous tables. Firstly I locate the value with the country code, then, I melt the column of the year. However, one thing that is different is that for table 6 I have to locate the columns because not all columns belong under the same information. To do so, I locate the columns I want for each split data frame. The following code shows the process:

```
#locating the columns I want to include basing on their position
regions1Df = regions1Df.iloc[:,0:8]
```

Finally, I separate the year from gender to two columns. The three data frames for "Estimated refugee stock at mid-year (both sexes)" looks as follow:

	Region	Country Code	Years	Estimated refugee stock at mid-year (both sexes)
0	Developed regions	901	1990	2014564
1	Developing regions	902	1990	16822007
2	Least developed countries	941	1990	5048391
3	Less developed regions excluding least develop...	934	1990	11773616
4	Developed regions	901	1995	3609670
5	Developing regions	902	1995	14244170
6	Least developed countries	941	1995	5160131
7	Less developed regions excluding least develop...	934	1995	9084039
8	Developed regions	901	2000	2997256
9	Developing regions	902	2000	12830547

	Country	Country Code	Years	Estimated refugee stock at mid-year (both sexes)
0	Burundi	108	1990	267929
1	Comoros	174	1990	0
2	Djibouti	262	1990	54508
3	Eritrea	232	1990	0
4	Ethiopia	231	1990	741965
5	Kenya	404	1990	13452
6	Madagascar	450	1990	0
7	Malawi	454	1990	874614
8	Mauritius	480	1990	0
9	Mayotte	175	1990	0

	Area of Destination	Country Code	Years	Estimated refugee stock at mid-year (both sexes)
0	Sub-Saharan Africa	947	1990	5516042
1	Eastern Africa	910	1990	3168001
2	Middle Africa	911	1990	446609
3	Northern Africa	912	1990	1202360
4	Southern Africa	913	1990	135525
5	Western Africa	914	1990	734857
6	Asia	935	1990	9937007
7	Central Asia	5500	1990	3000
8	Eastern Asia	906	1990	302529
9	South-Eastern Asia	920	1990	157977

Then, I repeated this process for “Refugees as a percentage of the international migrant stock,” and “Annual rate of change of the refugee stock” and generated three tables for each of them.

Insights

The insights I gained from completing this project reflect in multiple areas. The hardest part for me when accomplishing the task is to identify the problem sectors and come up with solutions. I spent the majority of the time trying to think of the different ways I can sort the given dataset. From this step I understood the constant challenge a data science has to face on a daily basis, especially when facing large data sets. On the other hand, this process allows me to be aware of the principles when creating new data sets in the future. In this way, it does not only benefit the people who are using the data set, but also the data scientists themselves in case they need to revisit the data.

A lot of the processes remind me of the book we are reading weekly. For instance, I always keep in mind to write comments for my code according to the chapter on documentation. On the other hand, I will try to set the column names in a way that they are objective and unbiased. Overall, the process of cleaning data taught me not only the code that needs to be done, but more importantly, the logical and critical view a data scientist must acquire when trying to comprehend and make sense of a data set.

Conclusion

In the final analysis, the three split data frames were generated from table 1-5, and 9 for table 6. The dataset are organized in the order of “Region,” “Countries,” and “Area of Destination.” In this way, I am able to achieve three things : For one, values are easier to track down because each cell contains a unique value. Secondly, the columns are contain information rather than values. Lastly, the each columns contain only one value. Overall, the outcomes of the task showcases more organized and clearer data frames.