University of Toronto
INF 1340: Programming for Data Sciences, Fall 2022

Midterm:

Tidying UN Migrant Stock Report, 2015

Audrey Medaino-Tardif

November 16th, 2022

1 Abstract

As this file will be made publicly available via GitHub (username: @odgerey), I seek to clarify the purpose and necessary accompanying files to this report:

This is the write up accompanying the midterm project. It should be read while looking at the files that are in the ZIP file. I've included in my final submission ZIP file a logbook of the work completed each day, as well as reflection points that have been summarized and more largely discussed further here. More so, I have provided a README file for identifying provenance and stacks used to avoid misuse of the files.

ZIP content:

- Jupyter File
- CSV + PICKLE files
- README
- PDF

2 Introduction

The purpose of this project was to complete two essential steps in the data wrangling process namely, data (re)structuring and data cleaning. We had to clean the data using the tidy data principles. You will see in my file that I have divided my data-cleaning file into two part. This was done more out of process than out of deliberate attempt to divide the sections. It was a three step process:

1. I began with basic principles, extracting the data, renaming the columns, etc.
2. I completely recoded and restructured my project (some structuring needed to be done before cleaning steps)
3. I tried paying attention to aesthetics. Aesthetics was definitely not my main concern, as the process and procedure were themselves a huge learning curve that necessitated a lot of time. The code would definitely need to be reworked in the real-world as it is not efficient.

3 Process

One of the elements that has been emphasized throughout the course thus far, is that thought process and critical decision making are just as important as the procedural work. There are many things that can impact the coding process, omissions or imputations. Therefore, I found it essential to keep a logbook to keep myself informed and on track each time I opened the project and worked on it. It takes a lot of concentration and this logbook helped get my juices flowing right when I opened the project, allowing me to quickly continue on a thread and sometimes, simply scrapping bad coding or incorrect analysis.

I found it useful as well to verbalize my next coding steps, jot reflection points, and force checkpoints throughout the project. The logbook has proved itself essential beyond comments in the code, as it tracked every step of the way issues I encountered and tinkering that I had to do before applying it to the project, whereas the Jupyter file is the result ”apres fait”. The code itself is a cleaned up version that has been changed multiple times before this version.

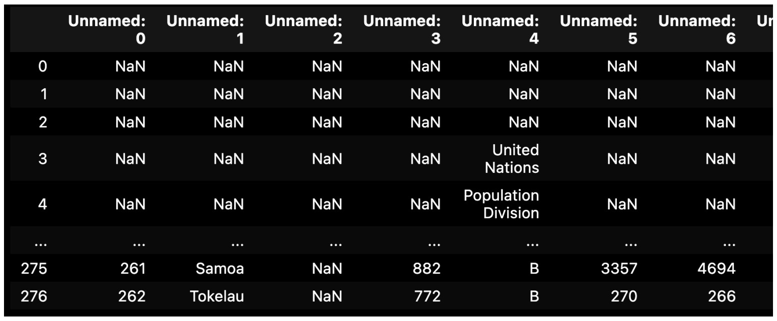
I started with good process ideas, however, if there is one thing I would do different next time is that I would do csv file ”checkpoints” at the end of each work session. For example, I had to completely recode my project because I had failed to see within the typical df.head(20) and df.tail(20) that I had unwanted replicating data that threw off my structure and threw an error each time I tried to merge columns from one dataframe to another. Thus, a saved file and quick glance checkpoint could really prevent scrapping a week or two’s worth of work.

4 Procedure & Principles of Tidy Data

My final submission restructures the data into ten different tables. This could have been four, which I will explain later in this report.

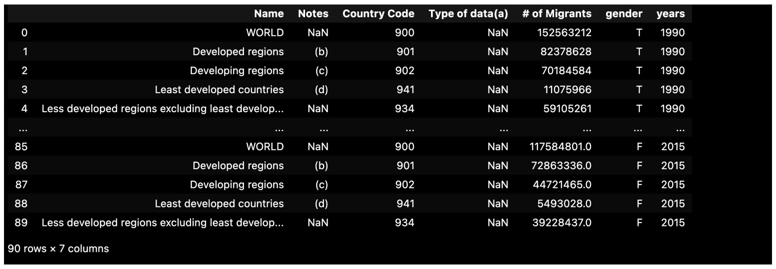
I would like to propose my procedure as a two-step process that is - first, basic tidying, and secondly, analyzing followed by more in depth tidying. Because tidying a data set is a rather new concept, I began by understanding the basics of the data before beginning with basic and very apparent issues.

Removing the titles lines 1-15 that mostly comprised of blank space (translated as a NaN value) was the first step I took, as well as basic renaming of the columns to their proper names. I did this for tables 1 through 6.



	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	United Nations	NaN	NaN
4	NaN	NaN	NaN	NaN	Population Division	NaN	NaN
...	...	...	...	...	...	...	...
275	261	Samoa	NaN	882	B	3357	4694
276	262	Tokelau	NaN	772	B	270	266

Figure 1: This is the original file's extracted display



	Name	Notes	Country Code	Type of data(a)	# of Migrants	gender	years
0	WORLD	NaN	900	NaN	152563212	T	1990
1	Developed regions	(b)	901	NaN	82378628	T	1990
2	Developing regions	(c)	902	NaN	70184584	T	1990
3	Least developed countries	(d)	941	NaN	11075966	T	1990
4	Less developed regions excluding least develop...	NaN	934	NaN	59105261	T	1990
...	...	...	...	...	...	...	...
85	WORLD	NaN	900	NaN	117584801.0	F	2015
86	Developed regions	(b)	901	NaN	72863336.0	F	2015
87	Developing regions	(c)	902	NaN	44721465.0	F	2015
88	Least developed countries	(d)	941	NaN	5493028.0	F	2015
89	Less developed regions excluding least develop...	NaN	934	NaN	39228437.0	F	2015

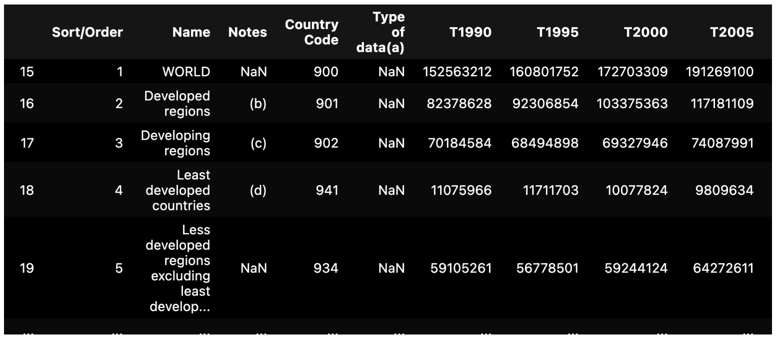
90 rows × 7 columns

Figure 2: The clean file's display

During this basic tidying of data, I decided to put aside the contents of "Notes", "Annex" and "Content" files to be handled at a later time. While I knew these contained important information, without yet knowing the structure of the project, I did not want to add an extra layer of information to my already very untidy dataframes. For the record, I also knew I would need to melt and separate my tables, which was the step I was most concerned with, as it involved a lot of coding and panda files I was unfamiliar with before the start of this project. Therefore, I handled Tidy Data Principles 1, 2, & 3 before handling any other.

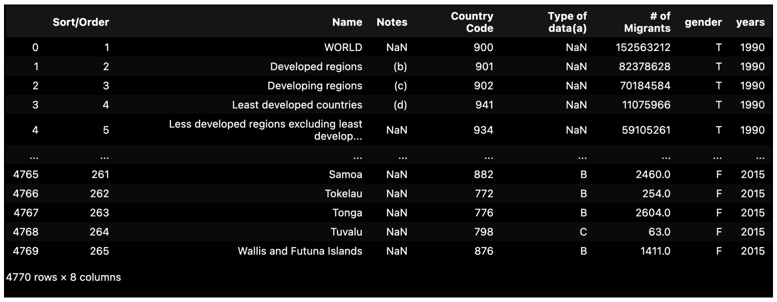
Principle 1, 2 &3 : Columns must be variable names and must not contain values. Each column needs to consist of one and only one variable. Rows must be observations and cannot contain values. Each row needs to consist of one and only one observation. There can only be one value measured and observed per cell.

I worked on Table 1 first, as my goal was to get a template of code to apply to the other tables and adjust where need be. All the tables contained multilayered or nested columns (dates) that were not variables. I used the melt and assign tools to reformat the data vertically and thereby remove non-variables from the columns and separate any double observations.



	Sort/Order	Name	Notes	Country Code	Type of data(a)	T1990	T1995	T2000	T2005
15	1	WORLD	NaN	900	NaN	152563212	160801752	172703309	191269100
16	2	Developed regions	(b)	901	NaN	82378628	92306854	103375363	117181109
17	3	Developing regions	(c)	902	NaN	70184584	68494898	69327946	74087991
18	4	Least developed countries	(d)	941	NaN	11075966	11711703	10077824	9809634
19	5	Less developed regions excluding least develop...	NaN	934	NaN	59105261	56778501	59244124	64272611
...	...	...	...	...	...	...	...	...	...

Figure 3: Before the melt & assign



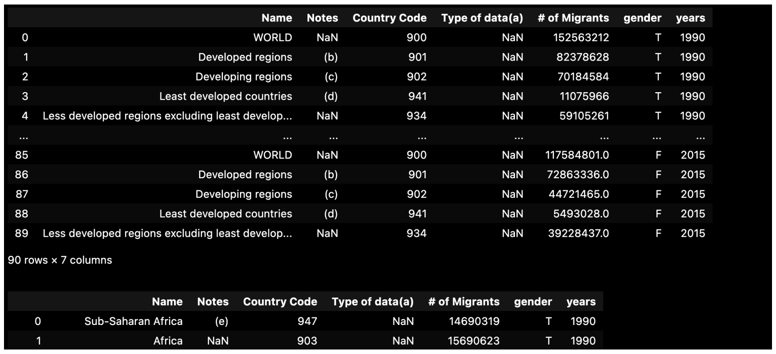
	Sort/Order	Name	Notes	Country Code	Type of data(a)	# of Migrants	gender	years
0	1	WORLD	NaN	900	NaN	152563212	T	1990
1	2	Developed regions	(b)	901	NaN	82378628	T	1990
2	3	Developing regions	(c)	902	NaN	70184584	T	1990
3	4	Least developed countries	(d)	941	NaN	11075966	T	1990
4	5	Less developed regions excluding least develop...	NaN	934	NaN	59105261	T	1990
...	...	...	...	...	...	...	...	...
4765	261	Samoa	NaN	882	B	2460.0	F	2015
4766	262	Tokelau	NaN	772	B	254.0	F	2015
4767	263	Tonga	NaN	776	B	2604.0	F	2015
4768	264	Tuvalu	NaN	798	C	63.0	F	2015
4769	265	Wallis and Futuna Islands	NaN	876	B	1411.0	F	2015

4770 rows × 8 columns

Figure 4: After the melt & assign

I applied the same code with small identification modifiers to all the other tables, except for Table 6, which required extra attention. Table 6 had the same problem, having years in the columns. It also had another problem, a year range that could not be interchanged with a single year. In my initial analysis, I thought I could transpose the date range to whatever value as a single year (i.e. rather than 1990-1995: 2, 1990:2, 1991:2, 1992:2, so forth.) However, this was not possible as there was no way of knowing what value to attribute to the year that landed in two date ranges (i.e. 1990-1995:2, 1995-2000:3, do I attribute 1995 to 2 or 3?). I also looked into merging Table 6's "Annual Rate Change" column into Table 5 that also dealt with date ranges and annual rate change. However, Table 6 is particular from the other tables as the data on refugees does not distinguish numbers by gender. For this reason, I decided to separate Table 6's annual rate change column into a separate table.

Having completed the basic column and row tidying, I began to dig a bit deeper analyzing the data itself.The "Major Area, Region" column beginning with the following measurements "WORLD", "Developed Regions", "Developing Regions"... really had me question whether this summarizing observation was necessary and logical as a value of a column "Area, Country, Region". While the "Notes" provided a few clues, as to what country was included in each, I had no real explanation as to what that data summed. As well, what defines the parameters or scale of developed and less developed, developed and developing? In the real world, I would have needed additional information from the person who gave me the data and if that was not possible, I would have left those first five rows of data out of the presented clean sheet, as it cannot offer any insight and could be used to misconstrue. I chose to separate those rows into a separate table, a summarizing table, for each table. This choice doubled my work load and forced me to recode my project entirely. I am not entirely convinced that was the best option. I could have rather added the following columns: "Status", "Area", "Region", "Country". This would have been clearer and would have enabled the one measured observation per cell rule, which was not respected in my final submission for the "Name" column. Separating WORLD into another table did not fix in retrospect.



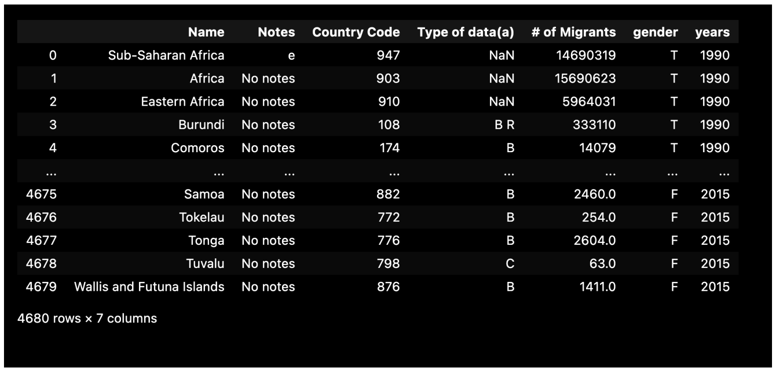
	Name	Notes	Country Code	Type of data(a)	# of Migrants	gender	years
0	WORLD	NaN	900	NaN	152563212	T	1990
1	Developed regions	(b)	901	NaN	82378628	T	1990
2	Developing regions	(c)	902	NaN	70184584	T	1990
3	Least developed countries	(d)	941	NaN	11075966	T	1990
4	Less developed regions excluding least develop...	NaN	934	NaN	59105261	T	1990
...	...	...	...	...	...	...	...
85	WORLD	NaN	900	NaN	117584801.0	F	2015
86	Developed regions	(b)	901	NaN	72863336.0	F	2015
87	Developing regions	(c)	902	NaN	44721465.0	F	2015
88	Least developed countries	(d)	941	NaN	5493028.0	F	2015
89	Less developed regions excluding least develop...	NaN	934	NaN	39228437.0	F	2015

90 rows × 7 columns

	Name	Notes	Country Code	Type of data(a)	# of Migrants	gender	years
0	Sub-Saharan Africa	(e)	947	NaN	14690319	T	1990
1	Africa	NaN	903	NaN	15690623	T	1990

Figure 5: As two separate tables

Finally, I dealt with "type of data(a)" and "Notes" – the NaN values. While I initially considered removing them both altogether, I felt this was too much of a data reduction. The nuances these notes and details provide are essential to any research or policy that can be conducted based on this data. I started by populating the NaN values in the "Notes" column. This is not problematic as the NaN values are not there for lack of data, but rather because there are no notes to be added for those measured observations. "No Notes" was then data that replaced NaN, which better represented the available data.



	Name	Notes	Country Code	Type of data(a)	# of Migrants	gender	years
0	Sub-Saharan Africa	e	947	NaN	14690319	T	1990
1	Africa	No notes	903	NaN	15690623	T	1990
2	Eastern Africa	No notes	910	NaN	5964031	T	1990
3	Burundi	No notes	108	B R	333110	T	1990
4	Comoros	No notes	174	B	14079	T	1990
...	...	...	...	...	...	...	...
4675	Samoa	No notes	882	B	2460.0	F	2015
4676	Tokelau	No notes	772	B	254.0	F	2015
4677	Tonga	No notes	776	B	2604.0	F	2015
4678	Tuvalu	No notes	798	C	63.0	F	2015
4679	Wallis and Futuna Islands	No notes	876	B	1411.0	F	2015

4680 rows × 7 columns

Figure 6: No notes

There was very important information contained in the type of data lettering that required extra effort and analysis in the appendix if left as letters in the tables. I found this inefficient and thus, attempted to turn each string into a column. I was successful, but only through a very inefficient manual coding, which returned a value of True or False if type of data b, migrants as foreign citizens, were counted in a country's data. The inconvenience of my boolean operation resulted in me being unable to remove the type of data (a) column despite having restructured its content into other columns.

Lastly, I merged Tables 2 and 3 with Table 1, as they contained information relevant to each other and had identical columns and rows. The last merge I would have liked to accomplish is Table 4 with Table 1. Here too, the rows are identical. The missing column information pertained to Male percentages, which could have been imputated based on other information found in Table1. However, this would have required extensively more time and tinkering.

I will make a brief note on aesthetics. My column names should definitely be revisited and reformatted for a more professional look. For example, "Name" is an improper column name and should have been changed to country or region for better clarity. However, as I was more concerned with process and procedure, I did not have the time to focus on renaming and because of those extra summarizing tables and my inefficient code for the "type of data" column, this would have required extensive extra time.

5 Reflections

I learned of the importance of titling and naming columns and rows to avoid getting lost mid-project when the data-cleaning sheet starts to be quite lengthy. However, I would add that it is also important to reduce defined parameters or functions to very small acronyms for code writing, as errors thrown because of spelling mistakes can use up a lot of time that could be better spent writing actual code rather than solving a silly problem. In my next data project, I will definitely keep track of my acronyms in my logbook, as from one work session to the next, exactly what you meant by "t" with the comment "Refactoring" really does not explain what the code is doing.

I might do a workbook for each table to avoid document confusion, which can get super messy when trying to get your code to do what is desired in multiple steps with several commented out unsuccessful attempts. I kept a record in case one of my previous efforts had a simple error. At the end, it might have been a bit easier to piece together separate Jupyter files into one clean one by copy and pasting. Though that could also be a cautionary tale if you wait to do that merge too close to a deadline.

Keeping your working file tidy and commenting and with clear defined code logic is essential. I created a new dataframe titled "addedColumn". This was great for the 20 minutes I spent on the piece of code, but when I was forced to restructure, I deleted the code and started from scratch as I had no idea what it did nor what it represented. For debugging, fixing, and restructuring, clear titles and comments are an absolute obligation.

As mentioned briefly earlier, I would have liked to remove the "Major Area, Country, Region" column and split it into multiple columns: "Area", "Region", "Country". These values would have been easy to populate as the original file already provides the information, simply in a different format. More so, these duplicated the data in a way, as they simply offered summations. For example, Northern Africa was simply a regional summation of the data already attributed to each country. The purpose of tidying data is to be able to make it as machine readable as possible to be able to parse and make those summations should they be needed. However, I did not know how to remove them from the rows.

My concluding reflections is that this code would definitely need to be refactored. While I was not sure how to merge python functions and loops and panda's tools,there must be a way to do most of the steps I did in a much more efficient and reusable code line. The purpose of code is to be able to repurpose functions to process large quantities of data. My results are very specific to this dataset and any changes to the source document would result in errors. In any case, it is as factored as I could make it for the experience in Python, Pandas and NumPy that I have.