

Reflection

Data analysis sometimes becomes frustrating, independently of the software you are using. This happens when you have either error messages or warning messages that do not allow you to complete your analysis. Fortunately, the more experience you get using software and the many available resources in a simple google search, coding and analyzing data has become easier. In this reflection, I will address all the problems I had using R software for my final project.

R has the advantage of being an open source software, my experience in a research institution in a developing country allows me to appreciate the availability of free software. However, as it was evident during the past semester, the fancier a model gets, the harder it is to find packages in R that allow us to run the code in a couple of lines. I would not consider myself an expert in the R language, but I feel the experience I have so far had made me quite fluent in specific packages.

Thankfully, in my final project, I barely bump into problems in the analysis of the data. However, I spend more time than expected cleaning the data and merging the data. Here is a summary of my experiences.

Merging data

There were several issues when merging data in R. I had to merge different years and different variables all spread in distinct excel files. The common variable was the country name. Common to a certain extent, some names included Republic in some data files, this was changed manually in Excel. The way I have learned to merge a file is using the *merge* function, which is used for tables or data frames. The code for this function is *merge(x, y, by= ” ”)*, my problem was that when two datasets merged they would not match the country observations, even though it was the same. I spend hours and hours trying to figure out what was wrong with it. Finally, I discovered that the observations of the variable country in some files had an amount of space, for example: “Argentina” and “Argentina ”. I looked in google for trimming white space in R [<https://stat.ethz.ch/R-manual/R-devel/library/base/html/trimws.html>]. Next are the steps I took to solve my problem.

This is how I discovered the problem:

```
R> lev12 <- levels(sci12$Country)
```

This is the solution to the problem:

```
R> level12 <- trimws(lev12, "r")
```

This is how I added the trimmed version back to my dataset:

```
R> levels(sci12$Country) <- level12
```

The second problem with merging data is that I did not want to eliminate rows that had missing observations, I wanted to keep all information available, so instead of using *merge* I used a function called *full_join* from the *dplyr* package:

```
PISA<- full_join(sci69, math69, by='Country')%>%  
  full_join(., read69, by='Country')%>%  
  full_join(., sci12, by='Country')%>%  
  full_join(., math12, by='Country')%>%  
  full_join(., read12, by='Country')%>%  
  full_join(., sci15, by='Country')%>%  
  full_join(., math15, by='Country')%>%  
  full_join(., read15, by='Country')
```

However, when merging the economic variables, where I had about 214 countries, and the education quality variables when I had at most 77, and since I just wanted to keep data for the countries with education variables available, I used the function:

```
EcoEdu<- left_join(PISA, Eco, by='Country')
```

From correlation to covariance matrix

I had some minor problems getting the full correlation matrix and then converting it into a covariance matrix. The *getCov* function in *lavaan* did not work for making my lower diagonal matrix into a symmetric matrix. So again, I look help in the web and found [<https://groups.google.com/forum/#!topic/lavaan/WF8gxtEKuE4>] that changing the code a little bit for obtaining the symmetric matrix.

```
ee.corr06 <- matrix(lowcor06, nrow = 6, ncol = 6,  
  dimnames = list(c("lgdp", "edq06S", "edq06M", "edq06R",  
    "growth06", "pol06"),  
  c("lgdp", "edq06S", "edq06M", "edq06R",  
    "growth06", "pol06")))
```

Overall, I did not have major problems when using R, and I was able to solve them relatively fast. Keeping in mind that we had to write this reflection was helpful because I made notes in my code of the problems I run into and how to solve them. Hence, writing this reflection became relatively easier. I hope this is not only informative but useful for other people who run into merging data frames or symmetric matrices problems in the future.