An Introduction to the tidyverse

Dr Deepak Varughese, MD

2022-05-15

What are "packages" in R.

R is a programming language. While a lot of work can be done on "base R" or the original R, the way to do them can get quite complicated. A "package" is code that have been pre-written by fellow R users to try to simplify some processes. What can be 10 lines of code may be compressed to a single line using packages. Packages in general make life a lot easier. On the flipside however it makes you a bit dependent on the makers of the package keeping that package available. However , for the most part, that should not be a problem. Packages help unlock the full potential of R by having others help you.
There are perhaps purist programmers who would like to write everything themselves but most/all R users use some package or the other. All R users probably use the tidyverse.

What is the "tidyverse"

The tidyverse is a collection of packages. Tidyverse is probably used universally by all R users. It consists of several R packages including dplyr (for data "wrangling" / "analysis" / "manipulation"), ggplot2 (for data visualization), lubridate (for working with dates), tidyr (for cleaning data) and more. If these names get confusing or foreign. Once you are on your way to analysis these packages will become your best friends.

For the purpose of these tutorials we will look at analysis and visualization using the tidyverse and not necessarily using base R. This may be more useful from a practical point of view and will help be a practical solution.

Installing and loading package

The classical way of loading a library requires 2 steps

- Installing the package (Needs to be done only once on a particular computer)
- loading the package (needs to done in every script that uses that package)

```
install.packages("tidyverse")
library(tidyverse)
```

Installing and loading package (recommended way)

Even though that is the classical way of loading a library i recommend using the "pacman" package. This uses the command p_load to load the library. If the required library is not installed , it will be installed and loaded. I prefer this method in many cases as this helps reproduciblity and avoids code from breaking if the other user does not have a certain package installed.

```
#install.packages("pacman",repos="https://cran.stat.auckland.ac.nz/" )
#(remove # and run if needed)

library(pacman)
p_load(tidyverse)
```