Tidyverse - Script 1

Thiyanga Talagala

07/04/2020

1. Introduction to Tidyverse

1.1 What is tidyverse?

1.2 It is a collection of R packages

```
library(tidyverse)
```

will load the core tidyverse packages in R. All packages seamlessly integrate and work together harmoniously.

1.3 Tidyverse workflaw

1.4 Tidyverse workflaw with packages

In terms of R packages, the workflow is nicely depicted as in this picture,

1.4.1 Importing data

1.4.2 Tidyr

Organise rows of data into unique values.

Here, there's one row for each acoustic measurement so that each word is spread acros five rows. This dataset is considered "tall" because there are many more rows than columns, giving the impression of a tall rectangle when you view it.

Why are there different ways of storing the data? It depends on the research question you're asking. If you're interested in each word, you might want to keep the extra-wide format so that each row represents a word. If you're interested in each acoustic measurement individually, you might want to use the tall version.

It's good to know how to convert your data from wide to tall and vice versa. Not only because it lets you look at it in different ways, but because certain kinds of visualizations require one or the other.

1.4.3 Transform your data

Once you have your data loaded, yo'll probably need to do some cleaning before moving on to analysis. We call this data pre-processing.

This step is important to make sure that your dataset is nice and clean and consistently formatted and [ready for analysis.]. Doing this in base R is little bit harder.

[Now that you've got your data loaded in], you'll proably need to do some preprocessing before moving on to analysis. [This is also called cleaning your data.] Clean your data by removing columns, renaming columns, re-ordering columns, adding columns, etc.

In my experience, my R code always starts off with many lines of preprocessing to make sure everything is nice and clean and consistently formatted before moving on to the statistics and visualizations.

What do I mean by "processing"? Well, it's likely that your data isn't exactly the way it needs to be for analysis. Maybe the column names aren't right, or a number that's supposed to be a number is actually being treated like text, or you may want to add, remove, or reorder columns entirely. Even if your data is pristine, you may have to do set a reference level for categorical data or filter the data in some way. All of this can be done using functions in the dplyr package, one of the core tidyverse packages. I'll cover some basics, but you can read more about the topics in this section R for Data Science chapters 18 on Pipes (http://r4ds.had.co.nz/pipes.html), and Chapter 5 "Data Transformation" (http://r4ds.had.co.nz/transform.html)

1.4.4: Visualising

1.4.5: Modelling

1.4.6: Communicate

Data

We will be using the gapminder dataset pioneered by Hans Rolling. These data represent the health and wealth of every nation in the world.

2. Pipe function (%>%)

This little function is one of the most useful feature feature of the tidyverse. What the pipe does is it takes the output of the function on the left and feeds it to the right function as its first argument.

3. Tibble

4. Factor

Script 2: readr

We saw how to get your data into R using `read_csv` and `read_excel` and how these functions are a bit more efficient than the base R equivalents.