Introduction to tidyverse (focusing on dplyr)

Julen Astigarraga

19 noviembre, 2020



I’m far from being an expert but I have quite a bit of experience in car crashes.

In this R Markdown file you will find the basic concepts to understand the core tidyverse collection of R packages. It is based on [tidyverse](https://www.tidyverse.org/) and [Jesse Mostipak’s dplyr tutorial](https://www.kaggle.com/jessemostipak/dive-into-dplyr-tutorial-1). I rearranged the information of these two files in a manner that I find the most interesting to understand tidyverse based on my own experience. Also, I add some comments and examples to understand tidyverse better. All credits deserves to the authors of the original documents.

# Tidyverse

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures. — [tidyverse](https://www.tidyverse.org/)

* [R for Data Science](https://r4ds.had.co.nz/) by Hadley Wickham & Garrett Grolemund
* Core tidyverse packages:
  + [readr](https://readr.tidyverse.org/): read rectangular data
  + [dplyr](https://dplyr.tidyverse.org/): data manipulation
  + [tibble](https://tibble.tidyverse.org/): modern reimagining of the data.frame
  + [tidyr](https://tidyr.tidyverse.org/): tidy data
  + [purrr](https://purrr.tidyverse.org/): functional programming
  + [stringr](https://stringr.tidyverse.org/): string manipulation
  + [forcats](https://forcats.tidyverse.org/): solve common problems with factors
  + [ggplot2](https://ggplot2.tidyverse.org/): graphics

## 1| readr (read rectangular data)

* *few advantages compared to base R*
  + Use a consistent naming scheme for the parameters (e.g. col\_names and col\_types not header and colClasses).
  + Are much faster (up to 10x).
  + Leave strings as is by default, and automatically parse common date/time formats.
  + Have a helpful progress bar if loading is going to take a while.

## 2| dplyr (data manipulation)

[A note by Roland Krasser](https://cran.r-project.org/web/packages/explore/vignettes/explore_mtcars.html): the data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

* Objective: select the best car brand (this example isn’t real)

1. First I’ll select those variable that I know the meaning
2. Then I would filter those cars whose number of cylinders is greater than 4.
3. Then arrange it in descending order
4. Finally show me the results
5. Create a new variable (column) with mutate relating the number of cylinders & miles/(US) gallon
6. But wait! I’m just interested in the brand (Mercedes, Ferrari etc.) not in the model
7. But I didn’t want that!
8. Clearly Pontiac is the best brand!

## 3| tibble (modern reimagining of the data.frame)

* They do less than data.frames
* They complain more than data.frames

## 4| tidyr (tidy data)

* tidy data 3 characteristics:

1. Every column is variable
2. Every row is an observation
3. Every cell is a single value

## 5| purrr (functional programming)

Laura will explain next week ;)

## 6| stringr (string manipulation)

A cohesive set of functions designed to make working with strings as easy as possible

## 7| forcats (solve common problems with factors)

## 8| ggplot

ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

1. I create a tibble that contains productivity data in 3 consecutive years and the measurement site.
2. Tidy data
3. Visualise putting site 1 at the end

