
Tidyverse

About me (in contex of R)

Using R as a main tool while working as a quantitative analysis/data scientist in financial industry since 3.5 years.

Last time I did attend Warsaw R meetup was more than a year ago and back then I did presented how to model death probability.

Co-organizing eRka (Kraków R user meetups).

Adam Wróbel

Problem

Base R is simply not enough to cover what we want to do in R. Therefore packages.

We have a tendency to load similar set of fairly general packages for all our work: dplyr, ggplot2, readxl, etc.

Solution

```
install.packages("tidyverse")

library(tidyverse)
```

packages included:

- ggplot2, dplyr, tidyr, readr, purr, tibble,
- hms, stringr, lubridate, forcats,
- DBI, haven, httr, jsonlite, readxl, rvest, xml2,
- modelr, broom

Benefits

- We load standard set of general purpose packages
- That are consistent (more or less)
- It is easier for new-comers if for instance there is a “standard” package/function to read excel file

Let's move from lapply to map function (purrr)

Adam Wróbel

Why lapply like function in a first place?

- Whenever you need to execute a function iteratively instead of writing a loop you could do the same with lapply
- For instance:
  - You want to get 95% quantile of each variable distribution
  - You want go thought list of all possible explanatory variables and build univariate model on each of them
  - You want to extract the R^2 from a long list of models
  - You want to produce a specific plot for each variable

Why lapply like function in a first place?

- All the things mentioned in previous slide you could achieve either by loops or lapply like functions

- Why lapply like functions then?
  - ✓Code readability
  - ✓Faster in most cases
  - ✓Less prone to errors

lapply – what it does exactly

- Executes a function on each element of a list/object:

> *lapply(data.frame, mean)* – calculates mean of each column

> *lapply(list_of_data_frames, lm)* – fits default linear regression model for all data frames and returns a list of lm objects

lapply family

lapply/apply family is great and widely used by R users, but there is something new that promises to be better: **purrr package**

"The apply family of functions in base R (apply(), lapply(), tapply(), etc) solve a similar problem, but purrr is more consistent and easier to learn"

Hadley Wickham

map function (purrr)

- Does the same thing as lapply family
- Combine lapply family into clearer map function syntax
- It is type stable (like vapply)

Let’s get into R Studio!

Pipes

Pipe version:

input %>% function(parameters)

Base version:

function(input, parameters)

Example 1:

data.frame %>% summary – execute summary function on a data.frame

Example 2:

data.frame %>% select(var1, var2) %>% plot – select two variables from a data.frame and plot them against each other

walk function (purrr)

- Like map, but it produce side effect only
- Does not return anything
- Does not alter pipe processing

Pipes and purrr (1/2)

simulation %>% walk(hist) %>% map(summary)

- take simulation data.frame and
- plot histogram for each column (simulated variables) and
- compute summary statistics for each column and return them

Pipes and purrr (2/2)

```
models <- mtcars %>% split(mtcars$cyl) %>% map(~ lm(mpg ~ wt, data = .))
map(models, summary) %>% map_dbl('r.squared')
```

- take mtcars dataset and
- create a list of data.frames by splitting by the number of cylinders (cyl) and
- build linear regression model for each data.frame in a list and
- save as a list of lm objects
- extract R^2 for all models

Should we move to purrr as R community?

Thanks for the attention!

Contact data

wrobel.adam1990@gmail.com

LinkedIn

https://pl.linkedin.com/in/wrobeladam1