

Chapter 1: Introduction

What is tidyverse?

Claire Liow

University of Tokyo

5/24/2022

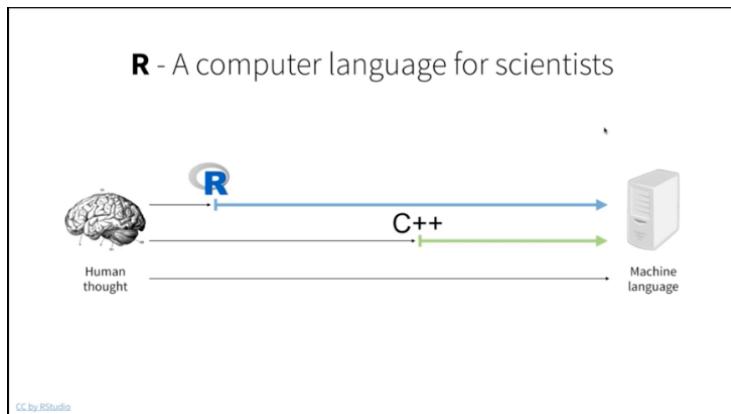
- 1 What is R and tidyverse?
- 2 Differences between Base R and tidyverse
- 3 Summary

Section 1

What is R and tidyverse?

What is R?

A programming language!



Source: *RStudio*

Then, what is tidyverse?



- A collection of R packages

Then, what is tidyverse?



- A collection of R packages
- A dialect of the Base R programming language

Then, what is tidyverse?



- A collection of R packages
- A dialect of the Base R programming language
- Support natural workflow of data analysis

Then, what is tidyverse?



- A collection of R packages
- A dialect of the Base R programming language
- Support natural workflow of data analysis
- Data import, tidying, manipulation, visualization, programming

Then, what is tidyverse?



Source: *Slyvia Canelon*

Section 2

Differences between Base R and tidyverse

Load tidyverse

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

Example

```
UNpop <- read.csv("data/UNpop.csv")
```

```
UNpop
```

```
##   year world.pop
## 1 1950  2525779
## 2 1960  3026003
## 3 1970  3691173
## 4 1980  4449049
## 5 1990  5320817
## 6 2000  6127700
## 7 2010  6916183
```

Let's calculate the % of population increase from 1950!

Base R syntax

```
## calculate the ratio compared to 1950  
UNpop$ratio <- UNpop$world.pop / UNpop$world.pop[1]  
  
## convert to percentage increase and round  
UNpop$percent <- round((UNpop$ratio - 1) * 100, 1)
```

tidyverse syntax

```
UNpop %>%  
  # calculate the ratio compared to 1950  
  mutate(ratio = world.pop / first(world.pop),  
         # convert to percentage increase and round  
         percent = round((ratio - 1) * 100, 1))
```

```
##   year world.pop    ratio percent  
## 1 1950  2525779 1.000000     0.0  
## 2 1960  3026003 1.198047    19.8  
## 3 1970  3691173 1.461400    46.1  
## 4 1980  4449049 1.761456    76.1  
## 5 1990  5320817 2.106604   110.7  
## 6 2000  6127700 2.426063   142.6  
## 7 2010  6916183 2.738238   173.8
```

Same task, with different syntax

Base-R

```
UNpop$ratio <- UNpop$world.pop / UNpop$world.pop[1]  
UNpop$percent <- round((UNpop$ratio - 1) * 100, 1)
```

tidyverse

```
UNpop %>%  
  mutate(ratio = world.pop / first(world.pop),  
         percent = round((ratio - 1) * 100, 1))
```

Section 3

Summary

Summary

How to distinct between base R/tidyverse?

- Base R: Lots of **\$** and **[[]]**
- Tidyverse: **%>%** (Forward pipe operator)

Why tidyverse?

- Resemble human language

Shortcut to type **%>%**

Cmd + **Shift** + **M** (Mac)

Ctrl + **Shift** + **M** (Windows)

Source: *RStudio*

Summary

How to distinct between base R/tidyverse?

- Base R: Lots of **\$** and **[[]]**
- Tidyverse: **%>%** (Forward pipe operator)

Why tidyverse?

- Resemble human language
- Intuitive and logical

Shortcut to type %>%

Cmd + Shift + M (Mac)

Ctrl + Shift + M (Windows)

Source: RStudio

Summary

How to distinct between base R/tidyverse?

- Base R: Lots of **\$** and **[[]]**
- Tidyverse: **%>%** (Forward pipe operator)

Why tidyverse?

- Resemble human language
- Intuitive and logical
- Combination of both base R and tidyverse

Shortcut to type **%>%**

Cmd + **Shift** + **M** (Mac)

Ctrl + **Shift** + **M** (Windows)

Source: *RStudio*