R functions and tidy evaluation

Date: 10th February 2023

Presenter: Balogun Stephen

About me

- Public health expert and data scientist
- Certified Rstudio tidyverse and shiny instructor
- Co-organizer Abuja R user group
- Author of two R packages {tidyndr} and {checkndr}
- Social media
    - LinkedIn: Stephen Balogun
    - Twitter : @eppydoc
    - GitHub: stephenbalogun
    - Website

Outline

- Definition
- Function structure/components
- Why write functions
- Types of functions
- More on dataframe functions
- Practice
- Q & A

Assumptions

Basic understanding of R

However, anyone should be able to follow along

- Functions are reusable blocks of code that perform specific tasks and can be called upon to run those tasks multiple times.
- Functions allow you to automate common tasks in a more powerful and general way than copy-and-pasting.

Definition

Function structure

```
name <- function(arguments) {
        body
}
```

```
filter <- function (.data, ..., .by = NULL, .preserve = FALSE)
{
    check_by_typo(...)
    by <- enquo(.by)
    if (!quo_is_null(by) && !is_false(.preserve)) {
        abort("Can't supply both `.by` and `.preserve`.")
    }
    UseMethod("filter")
}
```

fx name

argument

arg name

value

Body

Why write functions?

- Functions allow you to automate common tasks. As requirements change, you only need to update code in one place, instead of many.
- Reduces code duplication and waste
- Functions can have names that makes your code easier to understand.
- Eliminate the mistakes that can happen due to copy and paste

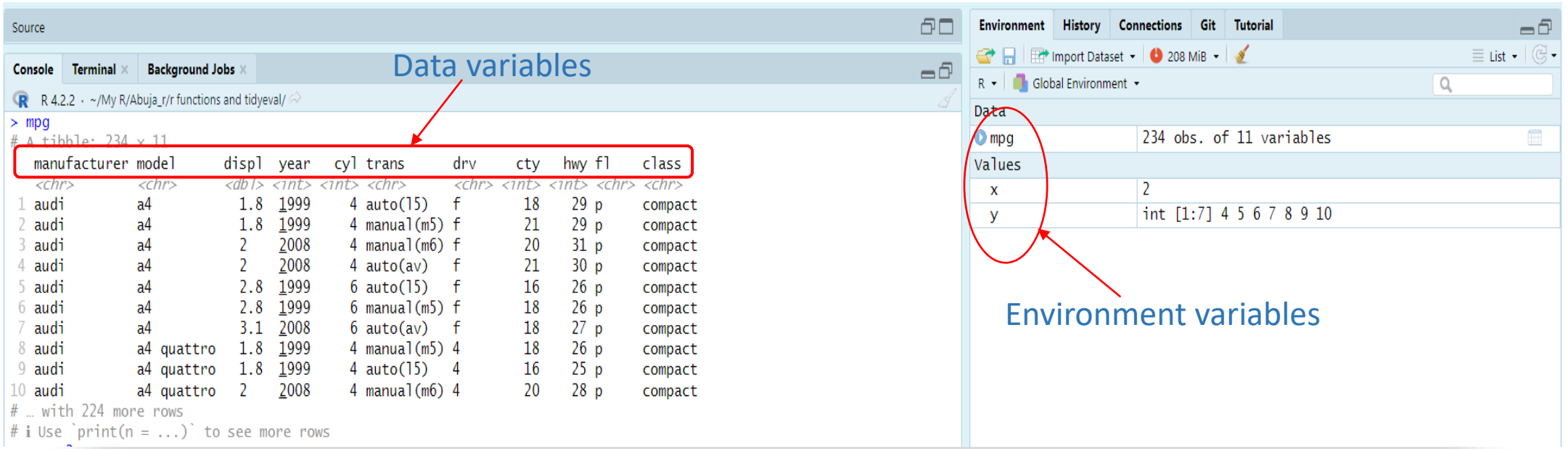
Types of functions

Vector functions

Dataframe functions

Plot functions

More on dataframe functions



Variables

- Environment variables
- Dataframe variables

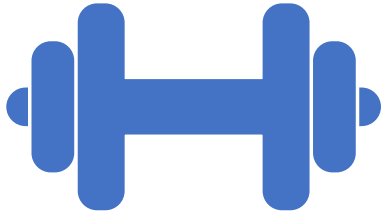
{tidyverse} and tidy evaluation

- Most base R functions are explicit in the type of variables referenced.
- {tidyverse} functions use tidy evaluation - a framework for writing R code that is both readable and flexible.
- Tidy evaluation makes code easy to read and understand by making variable referencing implicit
  - Data-masking: used by {tidyverse} functions that alter rows e.g., filter(), group_by(), arrange()
  - Tidy selection: used by {tidyverse} functions that alter columns e.g., select(), rename(), relocate(), across()
- However, this can create a problem when used in R functions - indirection

Practice

Scenario one: vector function

You are working with a data containing age of children (0-15). You would want to clean the data such that any data less than 0 or greater than 15 is recoded as missing



You are also aware that the weight should be between 0 and 45.So weights outside the range should be converted to missing

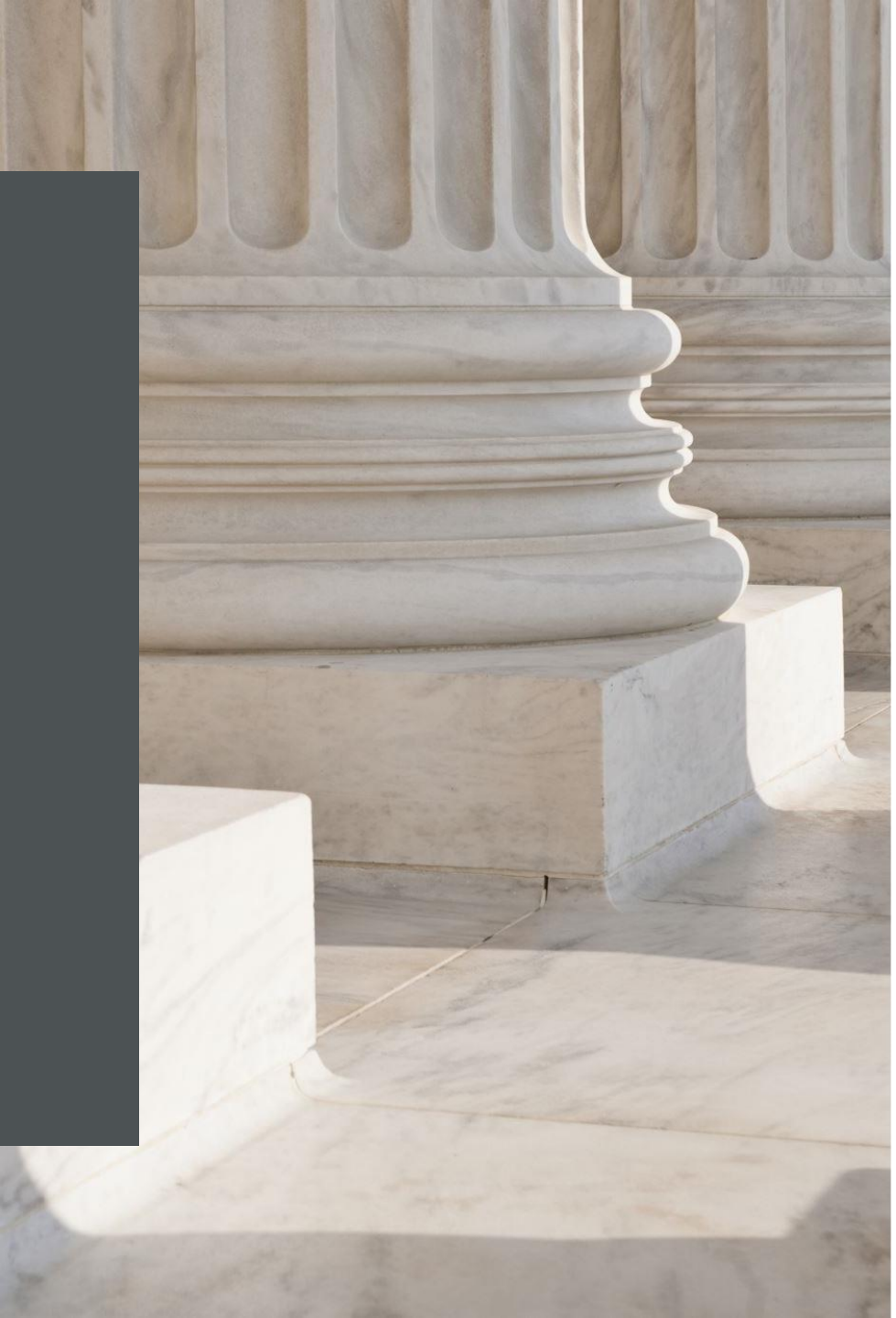
Scenario two: dataframe function for a specific task

summarise the diamonds dataset cut by price

summarise the diamonds dataset cut by depth

summarise the diamonds dataset color by price

summarise the diamonds dataset color by depth



Scenario three: make the function more generic

Reduce the four functions from previous slide to only one

Scenario four: supply more than one values as arguments

Scenario five: customize the variable names in your output

Additional resources

- https://rlang.r-lib.org/reference/topic-data-mask.html
- https://dcl-prog.stanford.edu/tidy-eval-section.html
- https://dplyr.tidyverse.org/articles/programming.html