Packages and Functions in R

ECON 490 (*Spring 2024*)

Taylor Mackay || Email: tmackay@fullerton.edu

Slides Overview

In these slides, we'll talk about packages and functions in R

Key package for this week is `tidyverse`

- Provides basic data cleaning manipulation tools
- Used in all future coding activities + capstone projects

Packages in R

Last week, we defined ***packages*** – collections of functions

- "Base R" is the set of functions included when you download R
- Packages expands the range of functions we can use

Two steps to using any package:

1) Install it using `install.packages()` – only have to do this once
2) Load it using `library()` – do this every time you open R

R Syntax

Last week, we defined ***objects*** – things like vectors, data frames, etc.

- We use ***functions*** to perform calculations on objects, transform them, etc.
- From the first coding activity, used functions like `mean()`, `min()`, `max()`, etc.

In general, functions take on the following form:

```
verb(arguments)
```

With `tidyverse`, we'll often be working with a data set, so we'd have:

```
verb(data.frame, argument.1, …)
```

R Output

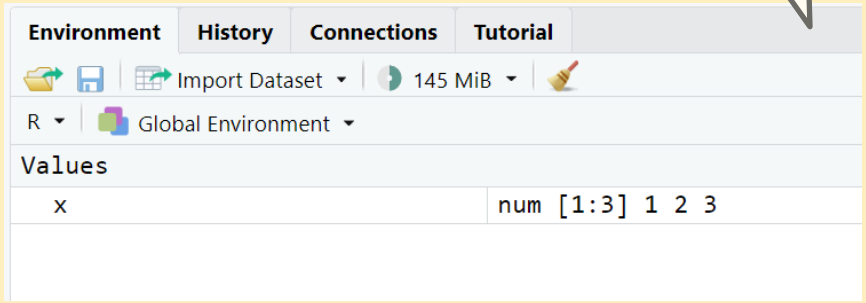
Whenever you run a function in R, you'll want to keep think about where the output of that function is going

Two general options here:

1. Let the output print in the console
2. Save the output as a new object using the assignment operator <-

With option (2), output is visible in "Environments" window

```
> # Option 1) print output in console
>
> c(1, 2, 3)
[1] 1 2 3
>
> # Option 2) save output as new object
>
> x <- c(1, 2, 3)
```



Different Tools for the Same Task

There's lots of R packages available for all kinds of tasks

- This means there's multiple ways to accomplish a single task
- It's important to think about ***how*** you're doing something

In practice, this means asking, "Am I using the appropriate package and function?"

For the coding activities and capstone projects, default choice is `tidyverse`

- Use whenever you're doing data cleaning or manipulation
- When getting help online or using AI, keep this in mind!

```r
# Create a sample dataframe
data <- data.frame(
  STATE = c("CA", "NY", "CA", "NY", "TX", "TX"),
  INCOME = c(50000, 60000, 55000, 70000, 48000, 52000)
)

# Calculate the average income grouped by state
average_income_by_state <- tapply(data$INCOME, data$STATE, mean)

# Print the result
print(average_income_by_state)
```

Prompt: *"Write r code to calculate the average value of INCOME grouped by STATE"*

Deceptively short – only makes sense if you're very comfortable R with tapply.

```r
# Install and load the tidyverse package if not already installed
if (!require(tidyverse)) {
  install.packages("tidyverse")
  library(tidyverse)
}

# Create a sample dataframe
data <- data.frame(
  STATE = c("CA", "NY", "CA", "NY", "TX", "TX"),
  INCOME = c(50000, 60000, 55000, 70000, 48000, 52000)
)

# Calculate the average income grouped by state
average_income_by_state <- data %>%
  group_by(STATE) %>%
  summarise(AVG_INCOME = mean(INCOME, na.rm = TRUE))

# Print the result
print(average_income_by_state)
```

Prompt: *"Write r code **using the tidyverse package** to calculate the average value of INCOME grouped by STATE"*

Nice and easy to read!

The Tidyverse Package

Here, we'll review the "Introduction to Tidyverse" slides

- Handout and slides are available on Canvas Week 3 Overview page
- Use these to help second coding activity!

As noted before, `tidyverse` should be your default choice for data cleaning

- In coding activities, questions will specifically reference `tidyverse` functions
- Use these functions (and not Base R or other packages) for full credit