Intro to R and RStudio

Jason Willwerscheid

9/6/2022

What is R?

- Created in 1991 as a “dialect” of S, a language that had been commercially available since 1976.

What is R?

- Created in 1991 as a "dialect" of S, a language that had been commercially available since 1976.
- Both R and S are *statistical* programming languages; that is, they were developed specifically with data analysis tasks in mind.

According to John Chambers, one of the creators of S:

"[W]e wanted users to be able to begin in an interactive environment, where they did not consciously think of themselves as programming. Then as their needs became clearer and their sophistication increased, they should be able to slide gradually into programming, when the language and system aspects would become more important."

Why R?

- Free and open source (unlike S!).

Why R?

- Free and open source (unlike S!).
- A large community of active users, which means. . .

Why R?

- Free and open source (unlike S!).
- A large community of active users, which means...
- A wide variety of online resources for learning R.

Why R?

- Free and open source (unlike S!).
- A large community of active users, which means. . .
- A wide variety of online resources for learning R.
- A large number of (open source!) **packages** for both simple and cutting-edge data analysis tasks.

Why R?

- Free and open source (unlike S!).
- A large community of active users, which means...
- A wide variety of online resources for learning R.
- A large number of (open source!) **packages** for both simple and cutting-edge data analysis tasks.
- For example, the `ggplot2` package makes it easy to create beautiful plots.

Why R?

- Free and open source (unlike S!).
- A large community of active users, which means...
- A wide variety of online resources for learning R.
- A large number of (open source!) **packages** for both simple and cutting-edge data analysis tasks.
- For example, the `ggplot2` package makes it easy to create beautiful plots.
- R is now the dominant language in industry "data science" positions.

What is RStudio?

- An **IDE** (integrated development environment) for using R.

What is RStudio?

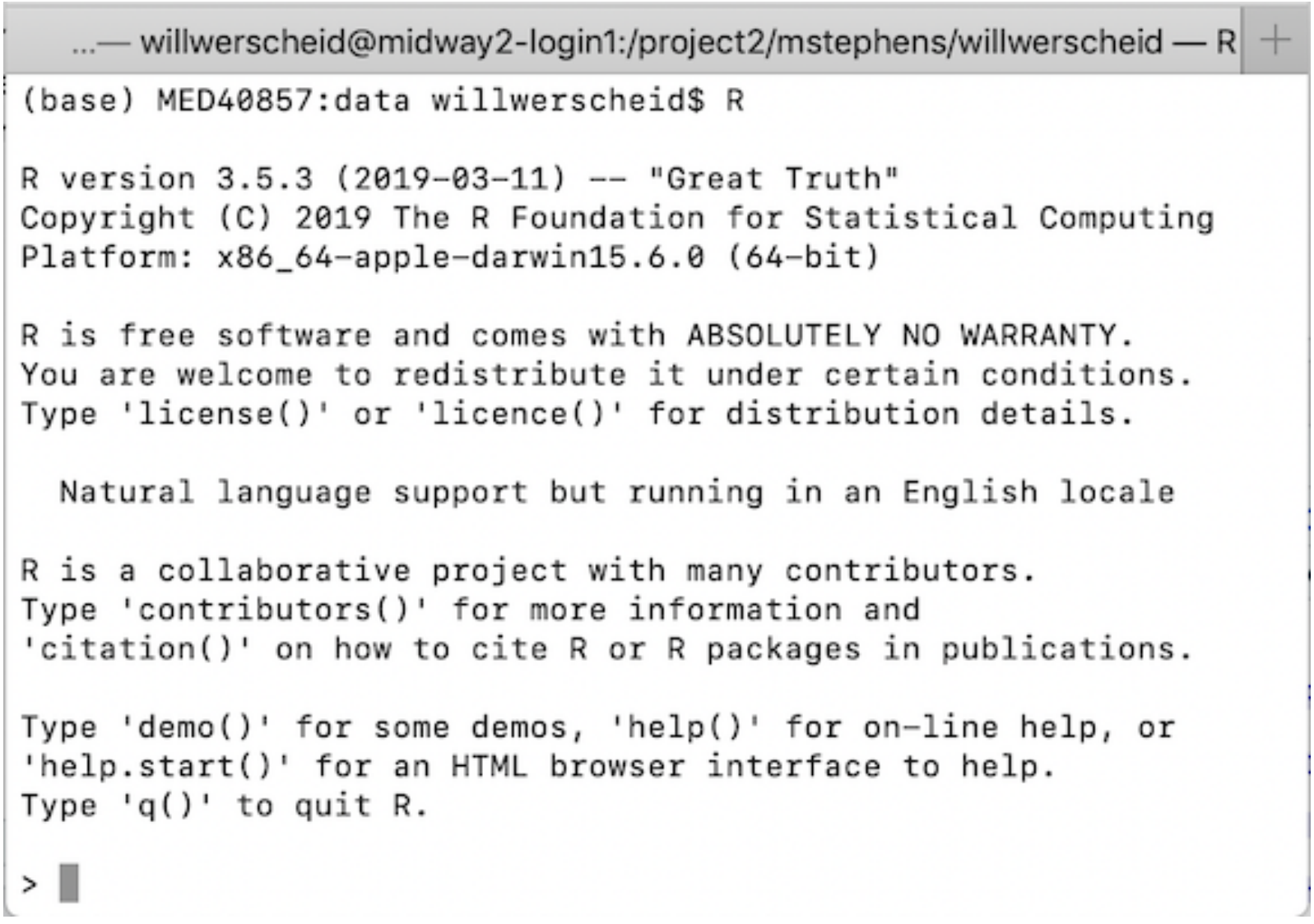
- An **IDE** (integrated development environment) for using R.
- Makes it much easier to do code-related things via a **GUI** (graphical user interface).

What is RStudio?

- An **IDE** (integrated development environment) for using R.
- Makes it much easier to do code-related things via a **GUI** (graphical user interface).
- Other IDEs exist, but RStudio is by far the most widely used.

Why RStudio?

Without RStudio, working in R looks something like this:



Why RStudio?

RStudio gives you the **console** for individual commands, but it also makes it easy to:

- Create, edit, and run R **scripts** instead of entering commands one by one.

Why RStudio?

RStudio gives you the **console** for individual commands, but it also makes it easy to:

- Create, edit, and run R **scripts** instead of entering commands one by one.
- Get help for a **function** or package.

Why RStudio?

RStudio gives you the **console** for individual commands, but it also makes it easy to:

- Create, edit, and run R **scripts** instead of entering commands one by one.
- Get help for a **function** or package.
- View and inspect **variables**.

Why RStudio?

RStudio gives you the **console** for individual commands, but it also makes it easy to:

- Create, edit, and run R **scripts** instead of entering commands one by one.
- Get help for a **function** or package.
- View and inspect **variables**.
- Debug code.

Why RStudio?

RStudio gives you the **console** for individual commands, but it also makes it easy to:

- Create, edit, and run R **scripts** instead of entering commands one by one.
- Get help for a **function** or package.
- View and inspect **variables**.
- Debug code.
- Create and tinker with visualizations.

Why RStudio?

RStudio gives you the **console** for individual commands, but it also makes it easy to:

- Create, edit, and run R **scripts** instead of entering commands one by one.
- Get help for a **function** or package.
- View and inspect **variables**.
- Debug code.
- Create and tinker with visualizations.
- Create reports via R Markdown.

Why RStudio?

RStudio gives you the **console** for individual commands, but it also makes it easy to:

- Create, edit, and run R **scripts** instead of entering commands one by one.
- Get help for a **function** or package.
- View and inspect **variables**.
- Debug code.
- Create and tinker with visualizations.
- Create reports via R Markdown.
- Manage projects via git.

Why RStudio?

You can certainly do each of these things outside of RStudio if you *really* want to; but RStudio allows you to do it all in one place. Think of it as a one-stop shop for managing R projects.

What is R Markdown?

- An R package like any other (install using the function `install.packages()`).

What is R Markdown?

- An R package like any other (install using the function `install.packages()`).
- Converts .Rmd files to readable reports (.md for GitHub, .pdf for printing, .html for webpages, etc.).

What is R Markdown?

- An R package like any other (install using the function `install.packages()`).
- Converts .Rmd files to readable reports (.md for GitHub, .pdf for printing, .html for webpages, etc.).
- An .Rmd file is a mixture of code, text, and markup (i.e., marks that tell R Markdown how to format the document).

What is R Markdown?

- An R package like any other (install using the function `install.packages()`).
- Converts .Rmd files to readable reports (.md for GitHub, .pdf for printing, .html for webpages, etc.).
- An .Rmd file is a mixture of code, text, and markup (i.e., marks that tell R Markdown how to format the document).
- Can easily add mathematical formulas like $\int_0^\infty \frac{1}{2\pi} e^{-x^2/2}\, dx$ via LaTeX.

What is R Markdown?

- An R package like any other (install using the function `install.packages()`).
- Converts .Rmd files to readable reports (.md for GitHub, .pdf for printing, .html for webpages, etc.).
- An .Rmd file is a mixture of code, text, and markup (i.e., marks that tell R Markdown how to format the document).
- Can easily add mathematical formulas like $\int_0^\infty \frac{1}{2\pi} e^{-x^2/2}\, dx$ via LaTeX.
- Can also be used to create presentations (like this one!).

Why R Markdown?

- Easy to create reports that include both code and commentary.

Why R Markdown?

- Easy to create reports that include both code and commentary.
- Others can download and run themselves, so the analysis is **reproducible.**

Summary

- R is a statistical programming *language*. It provides syntax for doing things via code.

Summary

- R is a statistical programming *language*. It provides syntax for doing things via code.
- RStudio is an *IDE*. It provides tools for writing R code.

Summary

- R is a statistical programming *language*. It provides syntax for doing things via code.
- RStudio is an *IDE*. It provides tools for writing R code.
- R Markdown is an R *package*. It provides functions for generating reports that include R code. (But it is so well integrated into RStudio that you will probably never type out any R Markdown functions).

Summary

- R is a statistical programming *language*. It provides syntax for doing things via code.
- RStudio is an *IDE*. It provides tools for writing R code.
- R Markdown is an R *package*. It provides functions for generating reports that include R code. (But it is so well integrated into RStudio that you will probably never type out any R Markdown functions).
- git is a *version control system*. It allows users to collaborate on projects and keeps track of changes.

Summary

- R is a statistical programming *language*. It provides syntax for doing things via code.
- RStudio is an *IDE*. It provides tools for writing R code.
- R Markdown is an R *package*. It provides functions for generating reports that include R code. (But it is so well integrated into RStudio that you will probably never type out any R Markdown functions).
- git is a *version control system*. It allows users to collaborate on projects and keeps track of changes.
- GitHub is a *hosting service* that allows users to manage online repositories via git.

Summary

- R is a statistical programming *language*. It provides syntax for doing things via code.
- RStudio is an *IDE*. It provides tools for writing R code.
- R Markdown is an R *package*. It provides functions for generating reports that include R code. (But it is so well integrated into RStudio that you will probably never type out any R Markdown functions).
- git is a *version control system*. It allows users to collaborate on projects and keeps track of changes.
- GitHub is a *hosting service* that allows users to manage online repositories via git.
- GitHub Desktop provides a *GUI* (i.e., a point-and-click interface) for interacting with GitHub.