R Lesson 1: R and RStudio

vanderbi.lt/r

Steve Baskauf

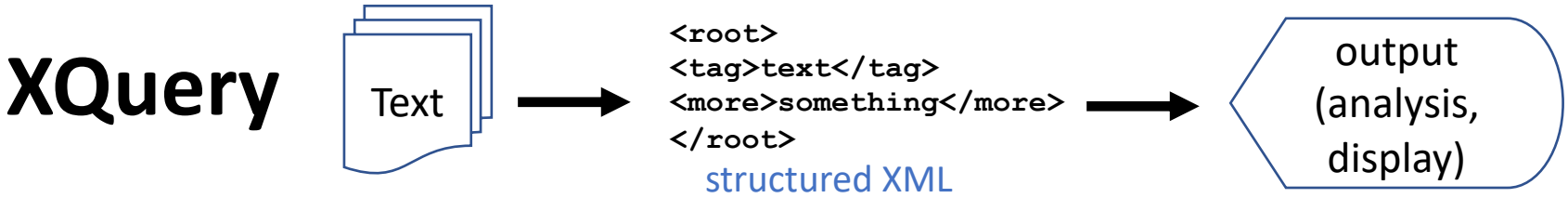
vanderbi.lt/r

Digital Scholarship and Communications Office (DiSC)

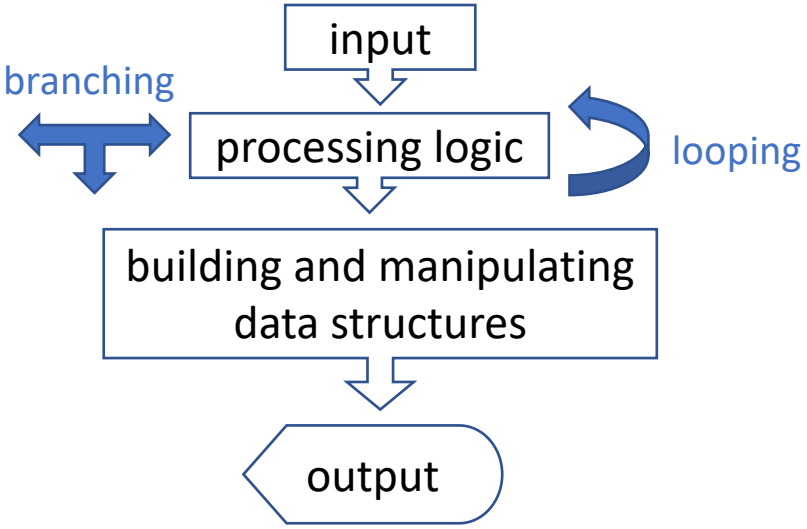
- Unit of the Vanderbilt Libraries
- Support for data best practices (DMP tool, repositories), GIS, copyright, Linked Data (including Wikidata), tools (GitHub, ORCID, Open Science Framework, etc.), and Open Access publishing.
- Offers on-demand educational programming, consultations, web resources
- Currently offering lessons on Python, R, and GIS
- More online at: **vanderbi.lt/disc**
- Email: disc@vanderbilt.edu

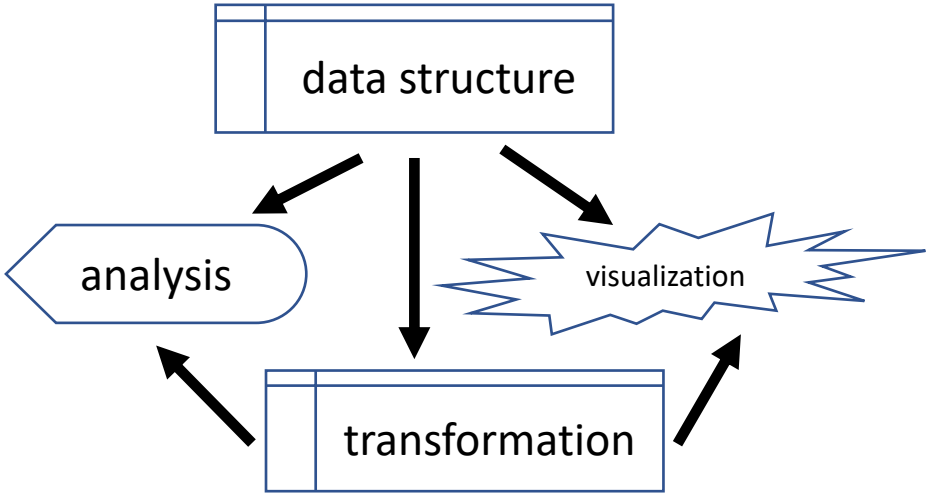
vanderbi.lt/r

Is R for you?



beginner **Python** advanced





advanced **R** beginner

Uses for R

- Statistical analysis
- Data wrangling and visualization
- Literate programming (R Markdown)
- Modeling
- Web development (Shiny)

This series will serve as a basic introduction to enable you to explore any of these topics

R basics

- Free, open source, multiplatform
- Package development
  - makes R extensible, huge, and powerful.
  - more centrally managed than Python (CRAN=Comprehensive R Archive Network)
- R the language vs. RStudio the IDE
  - R can be run by itself interactively in the console
  - R code can also be run as a script from the command line
  - RStudio is an Integrated Development Environment that makes it easier to run R interactively or as an entire script
  - RStudio is the ONLY common IDE for R (vs. many for Python)

The Anaconda option

- Includes Python, R, IDEs (Spyder and RStudio), Jupyter notebooks, and the VS Code editor as options.
- **Includes most common add-on packages**
- Includes a package manager called Conda.
- Considerations: size, conflicts with HomeBrew, differences in virtual environments
- Link to installation page on the "Installing R and RStudio" page.

Decide on your distribution

- Either:
  - do the big Anaconda installation
  - install R, then RStudio
- Can you install Anaconda if you already have RStudio? (yes)
- See the Installing R and RStudio page for instructions for installing them separately from Anaconda.

Ways to run R

- See web page "Navigating around in RStudio"
- Interactively in the console
  - Enter `r`
  - Prompt will change to ">"
  - Type commands. Results will be displayed immediately below.
  - Session "remembers" what's in your workspace between commands.
  - `q()` to exit. Workspace can be saved.
- Hard core uses do this, but not typically newbies.

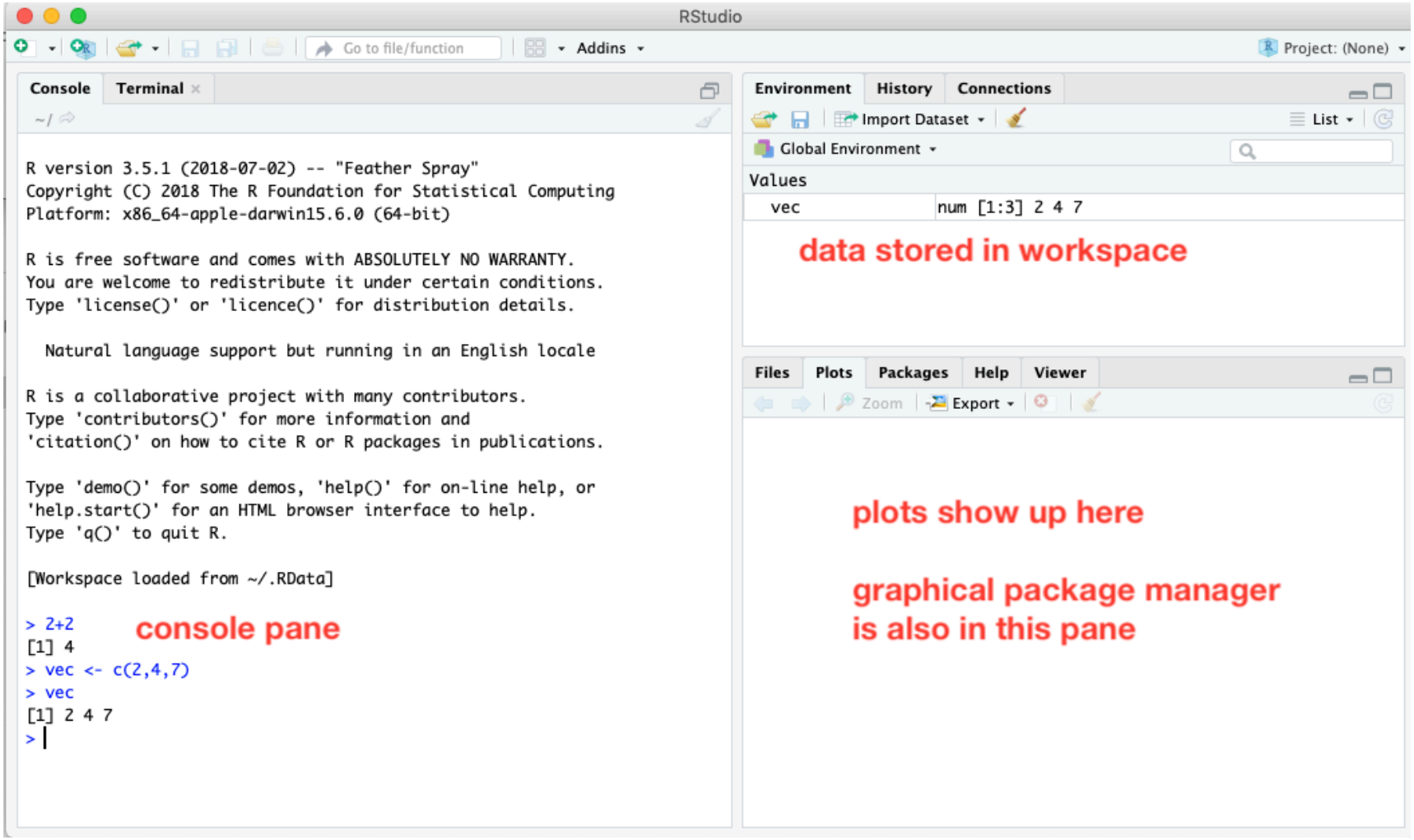
Ways to run R (cont.)

- By script using a code editor followed by **`Rscript`**
  - Series of commands saved in a file.
  - Script run in the console
  - Can be automated as part of Windows batch file or Linux shell script (on Mac)
- In a Jupyter notebook
  - Must set up an R environment in order to see R as an option for a new notebook (see installation page for link to instructions for setting up an R environment).
- These are both fine options in some circumstances, but more typically people use RStudio.

Ways to run R (cont.)

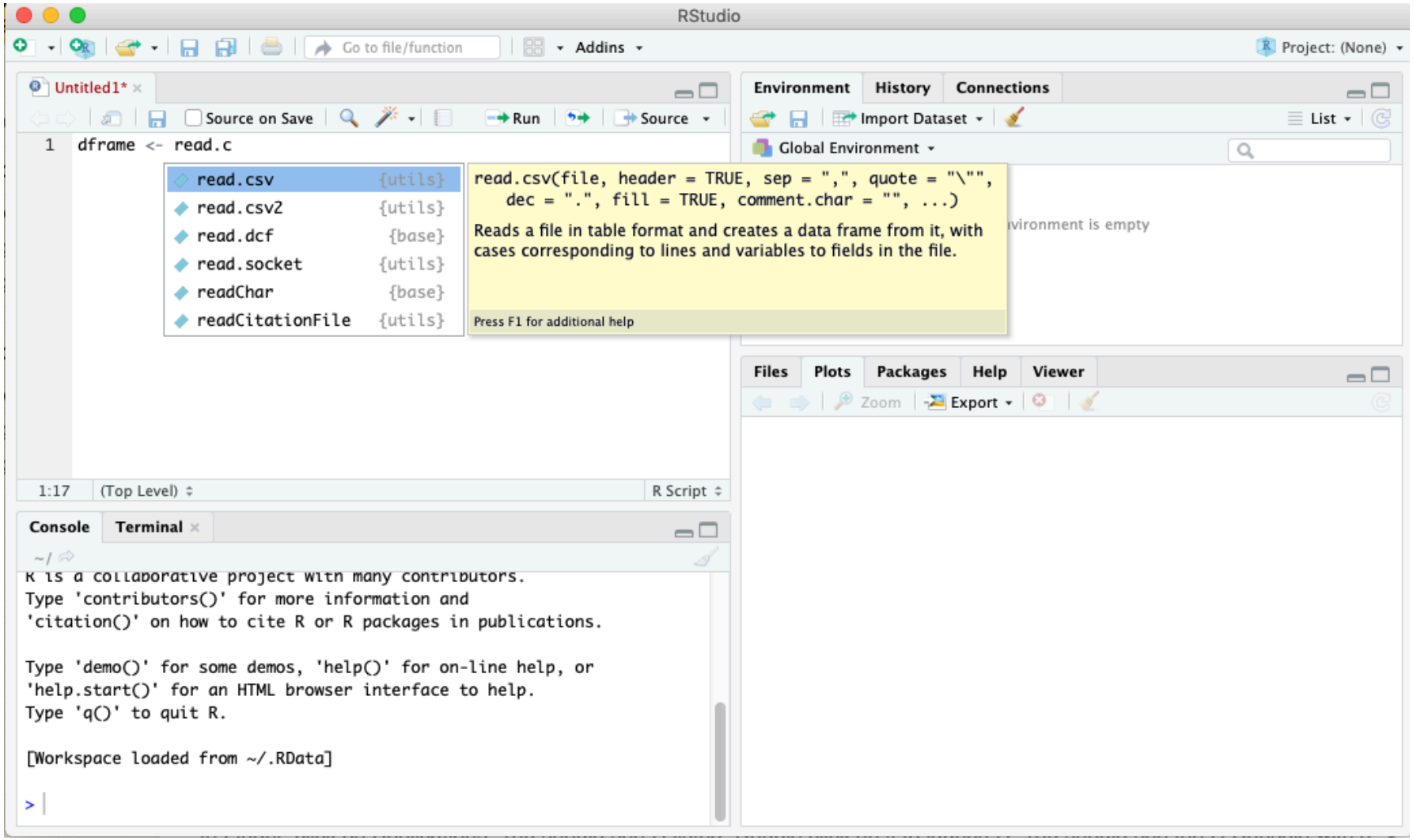
- Using the RStudio Integrated Development Environment (IDE)
  - This is what pretty much everybody does.
  - Code can be run interactively in the console pane
  - Part or all of a script can be run from the editor pane
  - RStudio also manages packages and allows you to visualize what's in your workspace
- The following instructions are detailed on a web page, so don't worry about taking notes. Follow along and try to do the examples.

Introduction to the RStudio graphical user interface (GUI)



Open an editor pane

File > New File > R Script



Running code

- By default only the current editor line is executed by the Run button (the line the cursor is on).
- Results show in console (lower left)
- MUST highlight the entire script to run all of it.
- Workspace environment details in upper right pane.

- Example on web page

Installing vs. loading packages

- Installing causes download from CRAN
- If using Anaconda, installing probably not necessary
- GUI vs. command line

```
install.packages("ggplot2")
```

- Loading makes the code in the package available to your script
- GUI vs. command line

```
library(ggplot2)
```

DIY example from web page

- Note that installing the ggplot2 and Hmisc packages may be required to generate the plot as shown on the page.
- Installing the entire tidyverse package is probably a good idea since it includes ggplot2 and other commonly used packages. It's big, so it takes a while…

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