R & RStudio

Introduction

R Basics

Why use R?

- R is free (Open Source)
- Extensive statistical modeling capabilities (through ∼14,000 packages[1])
- Flexible programming of user functions
- Working with multiple objects, i.e. various datasets & regression results
- Active user community providing up-to-date functions & support

[1] https://cran.r-project.org/web/packages/

R Basics

R Overview

- Download R & RStudio
  - `http://www.r-project.org/`
  - `https://www.rstudio.com/`
- Version & citation
  - `sessionInfo()`
  - `citation()`
- File formats
  - .Rdata: Workspace
  - .R: Code
  - .Rhistory: (saved) code
  - .Rmd: R Markdown file

R Basics

Setup

- List & change options: `options()`
- Set global options through .Rprofile
- Show working directory: `getwd()`
- Set working directory: `setwd("path(/)")`
- List files: `dir()`

Input

- Execute .R file: `source("file.R", echo = TRUE)`

Output

- Save code: `savehistory("file.Rhistory")`
- Save output: `sink("file.txt", split = TRUE)`

Functions & packages

```
function(object[...], option = ...)
```

- Functions
  - `function` without `()`: Show code of function
  - `function()`: ∼List corresponding elements
  - `function(...)`: Execute function as specified with `(arguments)`
- Syntax
  - R is case sensitive
  - Abbreviations of functions are not allowed
  - Comments begin with #
- Help
  - Function unknown: `help.search("topic")`
  - Function known: `help(function)`
  - Show similar functions: `fun...` + tab-key
  - Show available options: `function(` + tab-key

Functions & packages

Style guide

- `http://r-pkgs.had.co.nz/style.html`
  - File names: If files need to be run in sequence, prefix them with numbers
  - Object names: Use an underscore (_) to separate words
  - Spacing: Place spaces around all infix operators (=, +, -, <-, etc.) and after commas
  - Use <-, not =, for assignment
  - Comment your code (chunks)
  - ...

Functions & packages

Packages

- Functions are included in (numerous) packages
- List installed packages: `library()`
- List loaded packages: `search()`
- Update packages: `update.packages()`
- Searching for packages
  - `available.packages()`
  - `http://cran.r-project.org/web/views/`
- Installing & loading packages
  1. `install.packages("package")`
  2. `library(package)`

Objects & workspace

Everything in R is an object...

- Data structures: `str(...)`, `class(...)`
  - vector: A set of values / elements
  - factor: A set of named elements
  - data.frame: A two-dimensional set of vectors and/or factors
  - matrix: A two-dimensional set of objects with the same mode
  - array: A multidimensional matrix
  - list: A combinations of various objects
- Mode of an object: `mode(...)`
  - numeric
  - character
  - logical
  - ...
- Type of an object: `typeof(...)`
  - integer
  - double
  - character
  - logical
  - ...

Objects & workspace

The workspace

- Contains all available objects...
  - Datasets
  - Subsets of data
  - Model results
  - ...
- List all objects: `ls()`
- Print content of an object: `object`
- **Create object**: `object <- ...`
- Structure of an object: `str(object)`
- Remove object: `rm(object)`

Accessing data

Load and save .Rdata files

- Load .Rdata: `load("file.Rdata")`
- Save object(s): `save(object, file = ...)`
- Save workspace: `save.image(file = ...)`

Import and export e.g. Stata files: package `foreign`

- Import .dta file: `read.dta(...)`
- Import .sav file: `read.spss(...)`
- Export to Stata: `write.foreign(...)`

Accessing data

Selecting variables and observations

- Working with indexes
  - Basic structure: `data[...,...]`
  - Selecting variables: `data[ ,1:3]`
  - Selecting variables: `data[ ,c(1,3,4)]`
  - Selecting observations: `data[1:10, ]`
- Selecting variables using $-notation
  - Basic structure: `data$var1`
  - Combine multiple variables with `data.frame()`
- ~~Selecting variables using `attach`~~

Exploring data

Description I

- Dataviewer and -editor
  - View dataset: `View(...)`
  - Edit dataset: `fix(...)`
- Data overview
  - Data structure/ dimension: `str(...)`
  - Attributes (e.g. labels): `attributes(...)`
  - Variable names: `names(data)`
  - List first, last observations: `head(...)`, `tail(...)`
  - Number of observations, variables: `nrow(...)`, `ncol(...)`

Exploring data

Description II

- Central tendency
  - Arithmetic mean: `mean(...)`
  - Median: `median(...)`
  - Quantiles: `quantile(...)`
- Dispersion
  - Variance: `var(...)`
  - Standard deviation: `sd(...)`
  - Interquartile range: `IQR(...)`
- Summary & table
  - mean, median, 25th & 75th quartiles, min, max: `summary(...)`
  - Tukey's five number summary: `fivenum(...)`
  - Frequencies: `table(...)`
  - Cross tabulation: `table(...,...)`
  - Proportions: `prop.table(...)`

Modeling

```
y ~ x1 + x2, data = df
```

function	package	method
`lm()`	stats	linear regression
`glm()`	stats	generalized linear models
`lmer()`	lme4	mixed effects models
...	...	...

Programming

For loops

- Automate iterations
  - Output: Initialize empty object
  - Sequence: Determine what to loop over
  - Body: Code that is run in each iteration
- (Many) alternatives available
  - e.g., apply-family

```
> # Compute Pearson's second skewness coefficients in a for loop
> output <- vector(''double'', length(df))
> for (i in seq_along(df)) {
>   output[i] <- 3*(mean(df[,i]) - median(df[,i])) / sqrt(var(df[,i]))
> }
```

Tidyverse

"A collection of modern R packages that share common philosophies, embed best practices, and are designed to work together"

https://www.tidyverse.org/

Tidyverse

`dplyr`

- Package for transforming data the tidyverse-way
  - Extract variables (`select()`), rows (`filter()`), reordering (`arrange()`), grouping (`group_by()`)
  - Create new variables (`mutate()`), summarise variables (`summarise()`)

`magrittr`

- Provides the pipe operator **%>%** to pass results to next function
  - Run a sequence of transformations in one code block

```
> data %>%
>   select(var1, var7, var8, var9) %>%
>   filter(var1 >= x) %>%
>   mutate(var_new = (var7 + var8) / 2) %>%
>   group_by(var9) %>%
>   summarise(m = mean(var_new), v = var(var_new))
```

Tidyverse

`ggplot`

- Powerful plotting package based on the *grammar of graphics*
  - Build any graph using a set of structured components
  - Data + geoms + coodinate system
- Code components connected with +
  - Create complex plots in one code block
  - e.g. multiple layers, grouping, faceting

```
> ggplot(data) +
>   geom_point(aes(x = var1, y = var2)) +
>   geom_smooth(aes(x = var1, y = var2)) +
>   facet_wrap(~ var3) +
>   labs(x = 'new x lab') +
>   xlim(0, 100)
```

R Markdown

`rmarkdown`

- An authoring framework for data science
  1. Format text using Markdown
  2. Include and execute code with R code chunks
  3. Knit together text and results and output to various formats
- Heavily integrated into the RStudio IDE
- Bundles other packages for extensive functionality
  - knitr, e.g. .Rmd → .md
  - pandoc, e.g. .md → .html

`https://rmarkdown.rstudio.com/`

Resources

- `help.start()`
- Books
  - Adler, J. (2012). *R in a Nutshell*. Sebastopol, CA: O'Reilly.
  - Crawley, M. J. (2007). *The R Book*. Chichester: Wiley.
  - Wickham, H. and Grolemund, G. (2017). *R for Data Science*. Sebastopol, CA: O'Reilly.
- Reference Manuals
  - `https://cran.r-project.org/manuals.html`
- Online Learning
  - `https://www.rstudio.com/online-learning/#R`
  - `http://www.statmethods.net/`
- R vocabulary
  - `http://adv-r.had.co.nz/Vocabulary.html`
- Awesome R packages
  - `https://awesome-r.com/`