Keyboard shortcuts

Run source line(s) in console: Command+Enter (Ctrl+Enter on Windows)

Insert assignment operator, `<-`: Alt+- (dash)

Insert pipe operator, `%>%`: Cmd+Shift+M (Ctrl+Shift+M on Windows)

Cancel / get unstuck: Escape key. If you forget to close a paranthesis, quote, etc., and your R prompt shows `+` instead of `>`, focus cursor on console, and hit the Escape key.

General setup / options

- Tools, Global Options:
  - Uncheck "Restore .RData into workspace at startup"
  - Set to *Never*: "Save workspace to .RData on exit: Never"
- Use Projects! Create a project with *File – New Project*.
  - This creates a `.Rproj` file. This is just a metadata file that opens R *running in that folder* so R is running where your data lives.
  - *File – New File – New R Script* creates a new R script in your project folder. Save this script and write commands in a script. Use new scripts for new tasks/classes.

Packages

If you don't have a particular package installed already: `install.packages(packagename)`. Only **dplyr**, **readr**, and **tidyr** are strictly required. Load packages in each script each time you start R/RStudio.

```
library(dplyr)
library(readr)
library(tidyr)
```

dplyr verbs

- `filter()`: Limits *rows* – returns only rows matching conditions
- `select()`: Limits *columns* based on name, position, etc.
- `mutate()`: Adds new variables, modifies existing variables
- `arrange()`: Arranges data in ascending order (use `desc(var)` to arrange descending)
- `summarize()`: Reduces multiple values down to a single value
- `group_by()`: Groups a data.frame/tibble by one or more variables. Most useful with `summarize()`.

The pipe: %>%

When you load the **dplyr** library you can use `%>%`, the *pipe*. Running `x %>% f(args)` is the same as `f(x, args)`. If you wanted to run function `f()` on data `x`, then run function `g()` on that, then run function `h()` on that result: instead of nesting multiple functions, `h(g(f(x)))`, it's preferable and more readable to create a chain or pipeline of functions: `x %>% f() %>% g() %>% h()`. Pipelines can be spread across multiple lines, with each line ending in `%>%` until the pipeline terminates. The keyboard shortcut for inserting `%>%` is Cmd+Shift+M on Mac, Ctrl+Shift+M on Windows.