Day 1 Cheatsheet

Basic R

Major concepts

- **Package** - a package in R is a bundle or "package" of code (and or possibly data) that can be loaded together for easy repeated use or for sharing with others. Like an "expansion pack".
- **Function** - a function is a particular piece of code that allows you to do something in R. You can write your own, use functions that come directly from installing R, or use functions from additional packages.
- **Argument** - an option that you specify to a function.
- **Object** - an object is something that can be worked with in R - can be lots of different things!
- **Tidyverse** - This is a newer set of packages designed for data science that can make your code more intuitive as compared to the original older Base R.
- The R console is a full calculator:
  - `+`, `-`, `/`, `*` are add, subtract, divide and multiply
  - `^` or `**` is power
  - parentheses – (and) – work with order of operations
  - %% finds the remainder
- `#` is the comment symbol; nothing to its right is evaluated.

Functions

Library/Package	Piece of code	Example of usage	What it does
Base `R`	`<-`	`x <- 1`	Assigns a name `x` to something in the R environment.
Base `R`	`c()`	`x <- c(1, 3)`	Combines values into a vector.
Base `R`	`str()`	`str(x)`	Gets a summary of the object `x` structure.
Base `R`	`class()`	`class(x)`	Returns the type of the values in object `x`.
Base `R`	`length()`	`length(x)`	Returns how long the object `x` is.
Base `R`	`seq()`	`seq(from = 0, to = 100, by = 5)`	Generate regular sequences.
Base `R`	`rep()`	`rep(1, times = 10)`	Replicates the values in x. Can take `times` or `length.out` argument.

Library/Package	Piece of code	Example of usage	What it does
Base `R`	`sample()`	`sample(1:12)`	Takes a sample of the specified size from the elements of x using either with or without replacement. `replace = TRUE` samples with replacement.
Base `R`	`install.packages()`	`install.packages("tidyverse")`	Installs packages
Base `R`	`library()`	`library(tidyverse)`	Loads and attaches additional packages to the R environment. Done every time you start R.

RStudio

Major concepts

- **RStudio** - an Integrated Development Environment (IDE) for R - makes it easier to use R.
- **Source/Editor** - "Analysis" Script + Interactive Exploration - In a .R file (we call a script), code is saved on your disk
- **R Console** - Where code is executed (where things happen) - Code is not saved on your disk
- **Workspace/Environment** - Tells you what objects are in R. What exists in memory/what is loaded?/what did I read in?
- **R Markdown** - Files (.Rmd) help generate reports that include your code and output.
- **R Project** - Helps you organize your work. Helps with working directories (discussed later). Allows you to easily know which project you're on.
- RStudio Keyboard shortcuts

Functions

Library/Package	Piece of code	Example of usage	What it does
Base `R`	`View()`	`View(mtcars)`	allows you to view data in a spreadsheet format.
Base `R`	`head()`	`head(mtcars)`	Returns the top 6 rows of an object in the environment by default. You can specify how many rows you want by including the `n =`argument.

Library/Package	Piece of code	Example of usage	What it does
Base `R`	`tail()`	`tail(mtcars)`	Returns the bottom 6 rows of an object in the environment by default. You can specify how many rows you want by including the `n =` argument.

* This format was adapted from the cheatsheet format from AlexsLemonade.