Best practices for R

1. Use descriptive variable names (e.g. good - sample.size; bad- ss)
2. Break large projects into multiple scripts, modularize code with functions
   - Functions can be stored in a separate R file for better usability
3. File names: use 'underscore' when saving R files (example: calculate_sum.R)
4. Use 'dots' or 'caps' when creating a variable (example: test.data or TestData)
5. Assignment: use '<-' for assigning values to variable. Use '=' to set values inside a function
6. Comments: always use '#' to comment code and explain your logic
7. Spacing: use spaces between two code symbols (example: a <- 5 is better than a<-5)

Common mistakes in R

1. Using '&' instead of '&&'
   - A single '&' is used in logical expression (TRUE/FALSE) while double '&&' is used in for/while loops
2. Using '=' instead of '=='
   - A single '=' equates a variable, a double '==' tests if something is identical
3. Naming variables with a keyword or function (e.g. 'pi', which equals to ~3.14)
4. Not closing brackets () especially in large expressions
5. Using [n] instead of [[n]]
   - [n] represents the complete list whereas [[n]] represents the n^{th} element in the list
6. Using incorrect spacing (example: a<-5 is the assignment but a < -5 is a comparison between a and -5)
7. Not loading a package: always remember to load package after installing it (example: install.packages('MASS') should followed by library(MASS))
8. Using the wrong quotation marks (curved instead of straight)
   - " instead of "
   - R automatically conforms to straight quotation marks but you often need to re-type them in order for R to make the adjustment, instead of just pasting in code (i.e. from slides)