R Reference

Basic Operations

- Use `variable <- value` to assign a value to a variable.
- `#` starts a comment.
- Statements in a block must be surrounded by curly braces `{ }`.
- `?thing` displays the help for `thing`.
- `length(thing)` produces the length of a collection.
- `c(value1, value2, value3, ...)` creates a vector.
- `vector_name[i]` selects the i'th value from a vector.
- `mean`, `max`, and `min()` calculate simple statistics.
- `plot` creates simple visualizations.
- `list.files(pattern = "txt")` returns the names of all files that contain "txt" in their name.

Data frames

- `dim(dat)` gives the dimensions of a data frame.
- `dat[x, y]` selects a single element from a data frame.
- `dat[i, ]` selects the i'th row; `dat[, i]` selects the i'th column.
- `low:high` specifies a slice including elements from `low` to `high`.
- `apply(dat, 1, mean)` calculates the mean of each row.
- `apply(dat, 2, mean)` calculates the mean of each column.

Functions

- `name <- function(...args...)` defines a new function.
- `name <- function(arg = default)` specifies a default value for a parameter.
- Call a function using `name(...values...)`.

Control Flow

- Create a `for` loop to process elements in a collection one at a time:

  ```
  for (variable in collection) {
      ...body...
  }
  ```

- Create a conditional using `if` and `else`:

  ```
  if (condition_1) {
      ...body...
  } else if (condition_2) {
      ...body...
  } else {
      ...body...
  }
  ```

- Use `==` to test for equality.

- `X & Y` is only true if both X **and** Y are true.
- `X | Y` is true if either X **or** Y, or both, are true.
- Use `stopifnot(condition)` to check that something is true when the program is running.

Using R from the command-line

- `commandArgs(trailingOnly = TRUE)` returns the command-line arguments.
- `file("stdin")` reads from standard input.
- `cat(vec, sep = "\n")` writes to standard ouput each element of `vec` on its own line.