What We Expect You to Know

Structured Programming I

Know the material from the last lecture! In particular

- What the different types of vectors are, and what differentiates them from lists
- That matrices and data.frames are internally vectors and lists with special attributes
- Different ways of accessing and setting elements in vectors, lists, matrices, and data.frames
- Vectorization and recycling by operators
- Loops: for, while, repeat, next, break
- Conditionals: if, else, switch(), ifelse()
- Logical operators, differences between &,| and &&,|| and when each is needed

What We Expect You to Know

Structured Programming I

Learn about these useful functions included in R that you may not have known before. `help()` is always a good start for this!

- Sequences (similar to colon operator): `seq_len, seq_along, seq`
- Sets: `setdiff, union, unique, setequal, duplicated, anyDuplicated, table`
- Indexing & finding things: `which, which.max, which.min, max, min, pmax, pmin, match, row, col`
- Logic operations: `all, any, identical, all.equal, isTRUE, isFALSE, xor`
- Value transformations: `cut, hist, diff, floor, ceiling, round, trunc`
- Vector reordering: `head, tail, append, rep, rep_len, rev, sort, order, sample`
- Matrix creation and manipulation: `rbind, cbind, diag, expand.grid`
- Type and type conversion: `is.<TYPE>, as.<TYPE>, anyNA, is.na, finite, mode, type, typeof, ordered, factor, unlist, class`
- Matrix / Dim info: `dim, colnames, rownames, dimnames, nrow, ncol, NROW, NCOL, length, names`
- String operations: `regexpr, gregexpr, grep, grepl, sub, gsub, regexec, nchar, substr, strsplit, sprintf, toupper, tolower, paste, paste0`
- Functional operations (more on these in the next unit) `do.call, replicate, lapply, sapply, tapply, vapply, mapply, rapply, Map, Filter, apply`

What We Expect You to Know

Programming Style

Make your program easy to read by:

1. Avoiding unnecessary noise and keeping the appearance of your code uniform.
2. Using all communication channels available to you, like variable / function names, comments, and idioms that your reader is familiar with.
3. Assuming your audience has tunnel vision and should be able to understand small blocks of your code at a time

(And remember, the code you hand in is checked for style automatically)