

CS 1132 lecture 11

- I. Vectorized selection and filtering
 - a. Enables much more concise code
 - b. Goal: minimize mental energy required to understand *what* some code is trying to accomplish
 - i. Defer *how* the operation is accomplished to function calls or vectorized expressions
- II. Example: Hamming distance
 - a. Sum each occurrence of a character in one string differing from the corresponding character in another
- III. Example: Wheel-of-Fortune
 - a. Entire game can be implemented with two one-line functions
 - b. Use logical indexing to select which characters should be replaced without changing others
- IV. Selection in 2D arrays
 - a. Vectorized relations yield 2D array of same shape
 - i. Reminder: use single `&`, `|` for AND and OR, since short-circuiting does not apply to vectorized code
 - b. Selection via logical indexing yields a column vector, with elements selected in column-major order
 - i. Can multiply logical matrix by original to zero out non-selected values while retaining shape