**Lab: find, grep, awk 10-15-14**

* Unix split into linux and freebsd
* Find
  + find . –name “ora\*”
    - Means find here (.) anything with a name that starts with “ora”
    - -name is a flag for name
    - \* means anything that fits this (wild-carding)
  + find ~ -name “\*.txt”
    - means find any txt file in my home directory (~)
* Grep
  + grep “Moby Dick” pg2701.txt
    - find instances of “Moby Dick” in the document pg2701.txt (full text search)
  + nano oranges.txt
    - add text
    - ctrl o and enter (save), ctrl x (exit)
  + grep “orange” oranges.txt
  + grep “orange” \* means search for orange in all files
  + grep “orange” oranges.txt pg2701.txt means search for orange in both files
  + grep –r “orange” means search various directories (recursively)
* wc (word count)
  + wc pg2701.txt
  + word count: lines; words (in these lines); characters
  + grep “orange” oranges.txt | wc
  + history | wc: word count of history
  + | (pipe): take whatever history (or whatever) spits out and put it into word count
  + History | grep “ls”: full text search in history for “ls”
* Sort
  + sort oranges.txt
    - sort alphanumerically the contents
* Uniq
  + sort oranges.txt | uniq –c
    - sort doc, find unique instances of each phrase (sorted alphanumerically)
    - needs to be uniq –c
* Less
  + cat pg2701.txt | less
  + less: paginate it (but the pages are arbitrary)
* sed
  + sed: streamline editor
  + s/: substitute
  + sed ‘s/day/night/’ 24hr.txt
    - just substitutes night for day in output, not in file itself
  + sed –i ‘s/day/night’ 24hr.txt
    - makes changes (substitutes night for day) in actual document
  + safer option for making/saving changes (substitutions) to file: sed ‘s/day/night’ 24hr.txt >> night.txt
    - makes the substitutions, saves them in a new file (night.txt), leaves the original file (24hr.txt) alone
  + sed “s/day/night/g’ 24hr.txt >> night.txt
    - the g means global
* Assignment: part one: analyze the treatment of gender in Moby Dick (words that have to do with male/female, etc.)
  + Create a new project directory
  + Grep the text of Moby Dick
  + Make two lists of gendered words (file “male.txt” with male words in it; file “female.txt” with female words in it)
  + Grep the version of md with male words and its contexts, the version of md with fm words and its contexts
  + Get rid of “the”, other noncommon words
  + Count instances of gendered words, words that occur near them
  + Left with list of words that are commonly male/female in Melville
* Part 2: Weasel words:
  + Take a piece of writing you wrote, check for clarity
  + Make project directory
  + Take three papers you wrote
  + Change to plaintext
  + Make file with list of three weasel words “clearly, obviously,” other unclear words
  + Make script with words you’d like to replace them with