Sed Cheat Sheet

Sed is a stream editor. If that sounds strange, picture a stream of characters and file lines flowing through a pipe, and we can send a sed command to alter, delete, insert, match a regex pattern, and several other options.

Sed has several commands, but by far the most popular one is the substitute command: *s*. The substitute command changes occurrences of a regular expression to a new value. One simple example was given in one of our lectures, where we changed *all* the occurrences of "UNIX" by "Linux" in our input file.

```
sed "s/UNIX/Linux/g" file
```

Below you will find a quick table of popular options used with *sed.*

Usage: `sed [command-line-options] ['sed-command'] file`

Basic Examples:

sed -n 12,18p file	-*n* suppresses printing, except lines 12 to 18 (*p* forces print)
sed 12-18d	Show all lines, except lines 12 to 18 (*d* deletes the match)
sed G file	Double-space a file
sed 'G;G' file	Triple-space a file
sed -f script.sed file	Loads the commands inside *script.sed* and executes them
sed '$d' file.txt	Delete the last line
sed 's/foo/bar/' file	Replace foo with bar only for the first instance in a line
sed 's/foo/bar/4' file	Replace foo with bar only for the 4th instance in a line
sed 's/foo/bar/g' file	Replace foo with bar for all instances in a line
sed '/baz/s/foo/bar/g' file	Only if line contains baz, then substitute foo with bar

Command Line Options:

-n	No-printing (unless a specific print-p instruction is found)
-f scriptname	If you have several sed commands you can put them in a script file
-h	Prints a summary of sed commands
-e	Combine multiple sed commands (-e [first] -e [second] -e [third])

Pattern Flags:

/g	Global replacement (replaces all occurrences)
/1, /2, /3, etc.	Specifies which occurrence
/p	By default sed prints every line. /p forces print even when using -n
/w filename	Write the modified data to a file
/I	Ignore case (uppercase “i”)
FILENAME	References the current input file