

Think of sed as a sort of "find and replace" function;
You can find a good list of common things you'd want to do at
<http://www.cornerstonemag.com/sed/sedlline.txt>

Some quickies about regular expressions in sed:

A by itself just matches the letter A, like you'd expect.
[ABC] by itself just matches the letter A, B, or C.
A* tells it to match 0 or more A characters.
[ABC]* tells it to match 0 or more characters among A, B, C.
[A-Z]* tells it to match 0 or more characters among A, B, C, ..., Z.
[^A]* tells it to match 0 or more characters that aren't A.
(ABC)* tells it to match 0 or more repetitions of "ABC".

* is not the only modifier:

A+ tells it to match 1 or more A characters.
A? tells it to match 0 or 1 A characters.
A{4} tells it to match precisely 4 A characters.

They can get very complex very quickly, so sometimes it is easier to string a few sed calls through pipes instead of trying to brute force everything in one regexp; the performance penalty should hardly be noticable for small datasets or if you have patience!

sed can also delete certian lines or numbers of lines, but I haven't found this as useful as the substitution powers. Like I said before, good usage of sed forms part of the backbone of any complex scripting task.