

sed/awk

sed - stream editor - easier to perform operations on files you're passing through pipes. use any delimiter in sed commands

awk - useful for data extraction. condition followed by an action in curly braces.

sed remove blank lines

```
sed -i '/^$/d' file
```

add test 'new line' to end of file

-e allow multiple sed commands

eureka@ubuntu:~\$ cat test_file day,city,host 1,CHI,b 2,LON,c 3,TOK,a 2,CHI,c 3,CHI,a 2,LON,c 4,SAO,a	eureka@ubuntu:~\$ sed -e '\$a new line' test_file day,city,host 1,CHI,b 2,LON,c 3,TOK,a 2,CHI,c 3,CHI,a 2,LON,c 4,SAO,a new line
---	--

print line after pattern and print line that holds pattern

eureka@ubuntu:~\$ cat test_file day,city,host 1,CHI,b 2,LON,c 3,TOK,a 2,CHI,c 3,CHI,a 2,LON,c 4,SAO,a	eureka@ubuntu:~\$ sed -n '/3,CHI,a/,+1 p' test_file 3,CHI,a 2,LON,c
---	---

remove all lines that have more than 4 characters

eureka@ubuntu:~\$ cat test_file day,city,host 123 1,CHI,b 2,LON,c 3,TOK,a 2,CHI,c 3,CHI,a 2,LON,c 4,SAO,a	eureka@ubuntu:~\$ sed -i '/^.\{4\}/d' test_file	eureka@ubuntu:~\$ cat test_file 123
--	---	--

swap columns 2 and 3 and skip header

eureka@ubuntu:~\$ cat test_file day,city,host 1,CHI,b 2,LON,c 3,TOK,a 2,CHI,c 3,CHI,a 2,LON,c 4,SAO,a	eureka@ubuntu:~\$ awk '{FS=","; if (NR == 1) ; else print \$1, \$3, \$2}' < test_file 1 b CHI 2 c LON 3 a TOK 2 c CHI 3 a CHI 2 c LON 4 a SAO
---	--

add 200 to column 2

eureka@ubuntu:~\$ cat file.txt CHI,200,c
eureka@ubuntu:~\$ awk -F "," '{print \$1,\$2 + 200,\$3}' file.txt CHI 400 c