

Unix commands for data science

Manipulating input (Bash)

ctrl+f	cursor forward one character
ctrl+b	cursor backward one character
alt+f	cursor forward one word
alt+b	cursor backward one word
ctrl+a	cursor to beginning of line
ctrl+e	cursor to end of line
ctrl+p	previous input
ctrl+n	next input
ctrl+r	reverse search previous commands
ctrl+c	close running program
ctrl+d	close shell
ctrl+u	cut from cursor to beginning of line
ctrl+k	cut from cursor to end of line (kill)
ctrl+y	paste (yank)
TAB	autocomplete
!!	run previous command
alias l="ls -ltrh"	rename common commands

Unix Basics

Getting help

ls --help	prints command help
man rm	opens manual for command

Navigation

ls	list directory
pwd	print working directory
mkdir child	make directory named "child"
cd child	change directory to "child"
cd ..	change directory to parent
cd	change directory to home (~)
cp file newfile	copy file
rm file	remove (delete) "file"
rmdir child	remove empty directory "child"
find ~ -name "*.csv"	search for files ending with "*.csv"

Remote

ssh user@example.com	remote login (secure shell)
rsync user@example.com:file local/dir/	copy remote file
wget http://example.com/data.csv	copy file from web
curl -O http://example.com/data.csv	copy file from web

Managing Processes

parallel wc -w ::: *.csv	process files in parallel
top	display processes
ps -u	display user processes
kill 1234	kill process 1234
crontab -e	edit cron jobs (run a script daily/weekly/etc.)
sleep 100 &	append "&" to run in background
ctrl+z	stop foreground process
bg	resume stopped process in background
jobs	list running processes
fg 1	bring job 1 to foreground

piping (<, |, >, >>)

<<<	pass string as input to command
<	use file as input to command
	pass output as input
>	pass output to file
>>	append output to file

globbing

ls *.csv	list files ending with ".csv"
ls d*.csv	list files starting with "d" and ending with ".csv"
ls data????	list files starting with "data." followed by any 4 characters
rm [a-z]*[0-9]	remove files starting with a letter and ending in a digit

Data Manipulation

(assumes data are in comma separated fields)

Taking Subsets

cat data.csv	returns contents of "data.csv"
head data.csv	first ten lines
tail -15 data.csv	last 15 lines
tail -n +2 data.csv	everything but first line (remove header)
cut -d, -f2 data.csv	second column
awk -F, '{print \$2}' data.csv	second column
cut -d, -f2,4 data.csv	second and fourth column
cut -d, -f2 --complement data.csv	everything except second column
grep "NaN" data.csv	all lines with a "NaN"
grep -v "NaN" data.csv	all lines <i>without</i> a "NaN"
sort data.csv uniq	only unique lines
sort data.csv uniq -d	only duplicate lines
shuf data.csv	shuffle lines
shuf data.csv head -1	random line

Transforming Data

nano data.csv	minimal text editor
sort data.csv	sort lines alphabetically
sort -t, -n -k 2 data.csv	sort lines numerically by column 2
sed 's/,/ /g' data.csv	replace string "," with a space
tr 'A-Z' 'a-z' < data.csv	convert letters to lowercase
awk -F, '{print \$1/100}' data.csv	divide column 1 by 100
awk -F, '{print \$1*\$2}' data.csv	multiply columns 1 and 2
paste -s -d, data.csv	flatten data to row
tr ', ' '\n' < data.csv	flatten data to column
paste -d, data1.csv data2.csv	combines the lines of two files
join -d, data1.csv data2.csv	performs a join of two files

Summarizing Data

sed 's/,/ /g' data.csv wc -w	word count
wc -l data.csv	number of lines in "data.csv"
grep -c "NaN" data.csv	number of lines with a "NaN"
grep -o "NaN" data.csv wc -l	total number of "NaN"
awk -F, '{sum += \$1} END {print sum}' data.csv	sum of column 1
awk -F, '{sum += \$3} END {print sum / NR}' data.csv	average of column 3
awk -F, '{sum+=\$2; sumsq+=\$2*\$2}END{print sqrt(sumsq/NR-(sum/NR)**2)}' data.csv	standard deviation of column 2
cut -d, -f2 data.csv sort -n head -1	minimum of column 2
cut -d, -f2 data.csv sort -n tail -1	maximum of column 2
tr ', ' '\n' < data.csv sort -n tail -1	maximum of all columns

Generating Numbers

echo \$((123 * 456))	integer calculator
echo "12.3 * 456" bc	calculator
seq 3 11	sequence of numbers, inclusive
echo {3..11}	sequence of numbers, inclusive
shuf -r -i 0-100 -n 10	10 random numbers between 0 and 100 with replacement
shuf -i 0-100 -n 10	10 random numbers between 0 and 100 <i>without</i> replacement

Rapid Visualization with feedgnuplot

sudo apt-get install feedgnuplot	
cut -d, -f2 data.csv feedgnuplot --terminal 'dumb'	plot column 2 in terminal
cut -d, -f2 data.csv feedgnuplot --line	line plot column 2
awk -d '{print \$3, \$2}' data.csv feedgnuplot --domain	plot 2 over column 3
cut -d, -f2 data.csv feedgnuplot --histogram --with boxes	histogram of column 2

CC BY 4.0 Kyler Brown

source: <https://github.com/kylerbrown/unix-commands-for-data-science>
further reading
info [coreutils](http://coreutils.com)
<http://datascienceatthecommandline.com/>
<http://www.drbumsen.org/explorations-in-unix/>
<http://www.gregreda.com/2013/07/15/unix-commands-for-data-science/>