Unix commands for data science

Manipulating input (Bash)

`ctrl+f`	cursor **f**orward one character
`ctrl+b`	cursor **b**ackward one character
`alt+f`	cursor **f**orward one word
`alt+b`	cursor **b**ackward one word
`ctrl+a`	cursor to beginning of line
`ctrl+e`	cursor to **e**nd of line
`ctrl+p`	**p**revious input
`ctrl+n`	**n**ext input
`ctrl+r`	**r**everse search previous commands
`ctrl+c`	**c**lose running program
`ctrl+d`	close shell
`ctrl+u`	cut from cursor to beginning of line
`ctrl+k`	cut from cursor to end of line (**k**ill)
`ctrl+y`	paste (**y**ank)
`TAB`	autocomplete
`!!`	run previous command
`alias l="ls -ltrh"`	rename common commands

Unix Basics

Getting help

`ls -help`	prints command **h**elp
`man rm`	opens **man**ual for command

Navigation

`ls`	**l**i**s**t directory
`pwd`	**p**rint **w**orking **d**irectory
`mkdir child`	**m**a**k**e **dir**ectory named "child"
`cd child`	**c**hange **d**irectory to "child"
`cd ..`	**c**hange **d**irectory to parent
`cd`	**c**hange **d**irectory to home (~).
`cp file newfile`	**c**o**p**y file
`rm file`	**r**e**m**ove (delete) "file"
`rmdir child`	**r**e**m**ove empty **dir**ectory "child"
`find ~ -name "*.csv"`	search for files ending with "*.csv"

Remote

`ssh user@example.com`	remote login (**s**ecure **sh**ell)
`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 **b**ack**g**round
`jobs`	list running processes
`fg 1`	bring job 1 to **fg**oreground

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`	lists files ending with ".csv"
`ls d*.csv`	lists files starting with "d" and ending with ".csv"
`ls data????`	lists files starting with "data." followed by any 4 characters
`rm [a-z]*[0-9]`	removes 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 *without* a "NaN"
`sort data.csv | uniq`	only **uniq**ue lines
`sort data.csv | uniq -d`	only **d**uplicate lines
`shuf data.csv`	shuffle lines
`shuf data.csv | head -1`	random line
`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

Modifying Data

`nano data.csv`	minimal text editor
`sed '/s/,/ /g' data.csv`	replace string "," with a space
`tr 'A-Z''a-z' < data.csv`	convert letters to lowercase

Generating Data

`echo $((123 * 456))`	integer calculator
`bc <<< "12.3 * 456"`	calculator
`seq 3 11`	**seq**uence 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 *without* replacement

Summarizing Data

`sort data.csv`	sort lines alphabetically
`sort -t, -n -k 2 data.csv`	sort lines numerically by column 2
`sed 's/,/ /g' | wc -w data.csv`	**w**ord **c**ount
`wc -l data.csv`	number of **l**ines 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

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 '{print $3, $2}' data.csv | feedgnuplot --domain`	plot column 2 over column 3
`cut -d, -f2 data.csv | feedgnuplot --histogram --with boxes`	histogram of column 2

source: `https://github.com/kylerbrown/unix-commands-for-data-science`

further reading

`info coreutils`

`http://datascienceatthecommandline.com/`

`http://www.drbunsen.org/explorations-in-unix/`

`http://www.gregreda.com/2013/07/15/unix-commands-for-data-science/`