Do more and type less on the command line

Helen Cook

February 2018

~$

We are not here
because we like typing

Rather we use the command line for its flexibility and power

We are not here because I want to teach you about the shell

(although, that will happen too)

Rather because I want to inspire you to customize and improve your tools

New stuff first!

Ask questions about anything!

Shout if you have a better way!

3 different ways to extend your command toolbox

sam (alias/function)
newest (function)
tab (script)

A new, perhaps better, way to do all this stuff

How to always invoke ls with -l argument?

alias ll='ls -l'

```
alias ll='ls -l -color=auto'
```

How to randomly sample lines from a file?

Usage: cat file | sam

create alias for this part

There is a better way to do this,
we'll do that next

sam — outline

1.
2. Randomize the lines in the file
3. Select the top 10 lines
4.
5.

sam — outline

1. Number the lines in the file
2. Randomize them
3. Select the top 10 lines
4. Sort them back into their original order
5. Remove the line numbers

How to number the lines in a file?

nl

```
(helen@helens-mbp[ttys000])-(~/projects/type-less/examples)
(0)-(o)% head -5 chemicals
CID100000085    carnitine
CID100000119    gamma-aminobutyric acid (GABA
CID100000137    5-aminolevulinic acid
CID100000143    leucovorin
CID100000158    prostaglandin E2

(helen@helens-mbp[ttys000])-(~/projects/type-less/examples)
(0)-(o)% head -5 chemicals | nl
     1  CID100000085    carnitine
     2  CID100000119    gamma-aminobutyric acid (GABA
     3  CID100000137    5-aminolevulinic acid
     4  CID100000143    leucovorin
     5  CID100000158    prostaglandin E2
```

How to randomize the lines in a file?

shuf

```
(helen@helens-mbp[ttys000])-(~/projects/type-less/examples)
(0)-(o)% head -n 5 chemicals|shuf
CID100000158    prostaglandin E2
CID100000137    5-aminolevulinic acid
CID100000143    leucovorin
CID100000085    carnitine
CID100000119    gamma-aminobutyric acid (GABA
```

How to get the first N lines in a file?

head

```
(helen@helens-mbp[ttys000])-(~/projects/type-less/examples)
(0)-(o)% cat chemicals|nl|shuf|head -5
    49  CID100000853    thyroxine
  1065  CID100119830    tenofovir disoproxil fumarate
   782  CID100006058    cysteamine
   173  CID100002554    carbamazepine
  1125  CID100159247    sevelamer hydrochloride
```

How to sort the lines in a file?

sort

```
(helen@helens-mbp[ttys000])-(~/projects/type-less/examples)
(0)-(o)% cat chemicals|nl|shuf|head -5|sort -n
    19  CID100000297    methane
    94  CID100002099    alosetron
   211  CID100002733    stealth
   408  CID100003706    indinavir
  1194  CID103081276    fluticasone/salmeterol
```

How to remove columns from a file?

cut (or awk)

```
(helen@helens-mbp[ttys000])-(~/projects/type-less/examples)
(0)-(o)% cat chemicals|nl|shuf|head -5|sort -n|cut -f2-
CID100000923    sodium
CID100001134    thymidine
CID100002250    atorvastatin
CID100002487    butorphanol
CID100004163    methysergide
```

sam — outline

1. (nl) Number the lines in the file
2. (shuf) Randomize them
3. (head) Select the top 10 lines
4. (sort) Sort them back into their original order
5. (cut) Remove the line numbers

```
alias sam='nl | shuf | head | sort -n | cut -f2-'
```

As an alias, we can't do sam file

How to define a function?

```
sam () {

}
```

How to get the arguments to a function?

First argument is in $1

Second is in $2

etc…

How to test if a file exists?

```
[ -f $file ]
```

[-f $file]

What is the syntax for if?

```
sam () {
    if [                ]; then

    else

    fi

}
```

```
sam () {
    if [[ -f "$1" ]]; then

    else

    fi

}
```

```
sam () {
    if [[ -f "$1" ]]; then
        input="$1"
    else
        fi

}
```

How do you read from STDIN?

cat -

```
sam () {
    if [[ -f "$1" ]]; then
        input="$1"
    else
        input="-"
    fi

}
```

```
sam () {
    if [[ -f "$1" ]]; then
        input="$1"
    else
        input="-"
    fi
    cat $input | nl | shuf | head | sort -n | cut -f2-
}
```

```
sam () {
    if [[ -f "$1" ]]; then
        input="$1"
    else
        input="-"
    fi
    cat $input | nl | shuf | head | sort -n | cut -f2-
}
```

```
(helen@sund29736[ttys009])-(~/projects/type-less/examples)
(0)-(o)% sam chemicals
CID100001546    cladribine
CID100002162    amlodipine
CID100002914    cyproterone acetate
CID100003100    diphenhydramine
CID100003292    ethotoin
CID100003462    gemeprost
CID100003763    isoflurane
CID100003958    lorazepam
CID106918430    ceftobiprole
CID124812758    canagliflozin
```

```
(helen@sund29736[ttys009])-(~/projects/type-less/examples)
(0)-(o)% cat chemicals|sam
CID100003063    dihydrocodeine
CID100003372    fluphenazine
CID100003696    imipramine
CID100004499    nisoldipine
CID100004536    Milli
CID100083606    strontium
CID100092725    pyocin S3
CID100114709    eslicarbazepine
CID116132438    Humalog
CID118609542    Cephalosporins
```

For fun: adapt sam so it will work with very large files

For fun: adapt sam so it will work with very large files

Hint: seek

How to list the N newest files in a directory tree?

Usage: newest N

newest — outline

1. Find all the files in the directory tree
2. For each file …
3. Get the last modified date
4. Sort files into the right order
5. Print only the top 10 lines

Use a shell function to define “newest” that takes an optional argument

```
newest () {

}
```

How to specify an optional argument?

```
newest () {
    lines=10
    if [[ $1 != "" ]]; then
        lines=$1
    fi

}
```

How to list all files in a directory tree?

ls -R

Will list files recursively, but will order them by directory

find

```
find . -type f
```

From here

Only files

```
find . -type f
```

From here

Only files

find . -type f

```
(helen@sund29736[ttys009])-(~/projects/type-less/examples)
(0)-(o)% find . -type f
./chemicals
./solutions/sam.py
```

How to get the last modified date for a file?

find . -type f -printf '%T@ %p \n'

```
(helen@Helens-MacBook-Pro[ttys007])-(~/Google Drive/Human-Virus-PPI/data)
(0)-(o)% find . -type f -printf '%T@ %p \n'|head
1520072281.0000000000 ./.DS_Store
1519736844.0000000000 ./.STRING-VIRUSES-INSTALL.swp
1515412339.0000000000 ./.swp
1464360694.0000000000 ./A
1504086376.0000000000 ./actions_inconsistencies
1418833586.0000000000 ./alberto-payload/create_payload.py
 496841761.0000000000 ./alberto-payload/Icon
1418833739.0000000000 ./alberto-payload/stringrnautils.py
1418833742.0000000000 ./alberto-payload/stringrnautils.pyc
1464360700.0000000000 ./B
```

find . -type f -printf '%T@ %p \n'

UNIX time: Seconds since Jan 1 1970

```
(helen@Helens-MacBook-Pro[ttys007])-(~/Google Drive/Human-Virus-PPI/data)
(0)-(o)% find . -type f -printf '%T@ %p \n'|head
1520072281.0000000000 ./.DS_Store
1519736844.0000000000 ./.STRING-VIRUSES-INSTALL.swp
1515412339.0000000000 ./.swp
1464360694.0000000000 ./A
1504086376.0000000000 ./actions_inconsistencies
1418833586.0000000000 ./alberto-payload/create_payload.py
 496841761.0000000000 ./alberto-payload/Icon
1418833739.0000000000 ./alberto-payload/stringrnautils.py
1418833742.0000000000 ./alberto-payload/stringrnautils.pyc
1464360700.0000000000 ./B
```

```
newest () {
    lines=10
    if [[ $1 != "" ]]; then
        lines=$1
    fi
    find . -type f -printf '%T@ %p \n' |

}
```

How to sort on the dates?

sort -n

```
newest () {
    lines=10
    if [[ $1 != "" ]]; then
        lines=$1
    fi
    find . -type f -printf '%T@ %p \n' |
    sort -n |


}
```

How to show the last $lines lines?

```
newest () {
    lines=10
    if [[ $1 != "" ]]; then
        lines=$1
    fi
    find . -type f -printf '%T@ %p \n' |
    sort -n |
    tail -n $lines |

}
```

How to show only the filenames?

cut

Field delimiter is a space

```
cut -f2- -d " "
```

Print field 2 and onwards

```
newest () {
    lines=10
    if [[ $1 != "" ]]; then
        lines=$1
    fi
    find . -type f -printf '%T@ %p \n' |
    sort -n |
    tail -n $lines |
    cut -f2- -d " "
}
```

```
newest () {
    local lines=10
    if [[ $1 != "" ]]; then
        lines=$1
    fi
    find . -type f -printf '%T@ %p \n' |
    sort -n |
    tail -n $lines |
    cut -f2- -d " "
}
```

But it doesn’t work on my Mac

Gnu find has -printf

BSD find doesn’t

http://brew.sh

brew install coreutils

How to get the last modified date for a file?

stat -f "%m %N"

How to run a command on each file found?

xargs

find … | xargs <cmd>

```
(helen@sund29736[ttys009])-(~/projects/type-less/examples)
(0)-(o)% find . -type f -print0 | xargs -0 ls -lt
-rwxr-xr-x  1 helen  staff   1723 Mar  8 11:10 ./solutions/sam.py
-rw-r--r--  1 helen  staff  35785 Feb  1 12:05 ./chemicals
```

```
newest () {
    lines=10
    if [[ $1 != "" ]]; then
        lines=$1
    fi
    find . -type f -print0 |
    xargs -0 stat -f "%m %N" |
    sort -n |
    tail -n $lines |
    cut -f2 -d " "
}
```

```
newest () {
    local lines=10
    if [[ $1 != "" ]]; then
        lines=$1
    fi
    find . -type f -print0 |
    xargs -0 stat -f "%m %N" |
    sort -n |
    tail -n $lines |
    cut -f2 -d " "
}
```

How to avoid awk for tab delimited files?

```
awk -F"\t" '{print $3 "\t" $5 "\t" $3 "\t"}'
```

```
awk -F"\t" '{print $3 "\t" $5 "\t" $3 "\t"}'
```

This is terrible

Let’s replace:

```
awk -F"\t" '{print $3 "\t" $5 "\t" $3 "\t"}'
```

With:

```
tab 3 5 2
```

tab — outline

1. Get columns to print from cmd line
2. Build up awk print string
3. Run awk

Use a script to define “tab”

Pass the column numbers as arguments

```
#!/bin/sh
```

How to get the arguments to a script?

First argument is in $1

Second is in $2

etc…

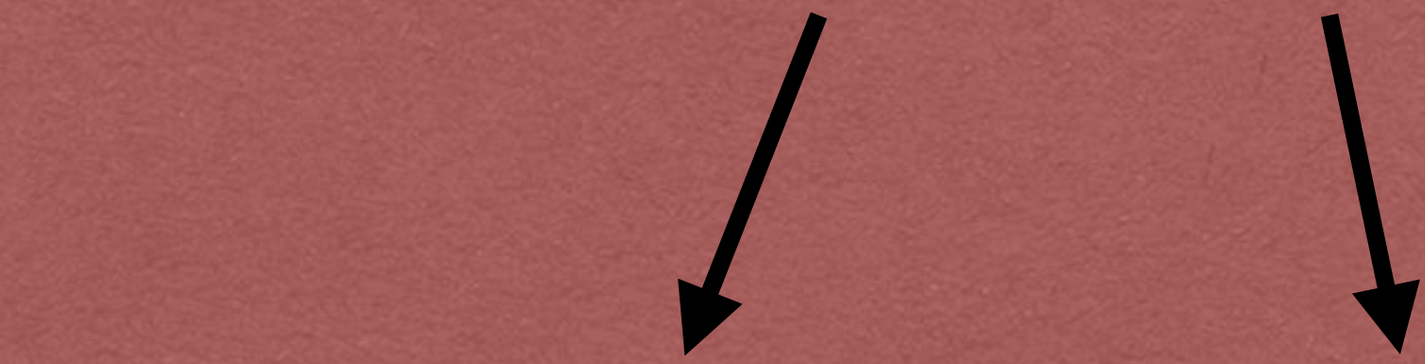
How to iterate through all arguments?

while $1 is not empty:
shift the arg array to move
the next arg into $1

What is the syntax for while condition in sh?

```
while [      ...      ]; do
...
done
```

MANDATORY SPACES!!



```
while [ ... ]; do
...
done
```

```
#!/bin/sh

while [                    ]; do

done;
```

```
#!/bin/sh

while [ "$1" != "" ]; do
    shift;
done;
```

How to construct the awk string?

1

1 \t 2

1 \t 2 \t 3

… and get the quoting correct

```
#!/bin/sh

first=1
while [ "$1" != "" ]; do
    if [ $first == 1 ]; then
        first=0
    else

    fi
    shift;
done;
```

How to concatenate strings in sh?

```
str = "some"
```

```
str = "$str"more
```

```
#!/bin/sh
printstring=""
first=1
while [ "$1" != "" ]; do
    if [ $first == 1 ]; then
        first=0
    else

    fi
    shift;
done;
```

```
#!/bin/sh
printstring=""
first=1
while [ "$1" != "" ]; do
    if [ $first == 1 ]; then
        printstring=
        first=0
    else
        printstring=
    fi
    shift;
done;
```

```
#!/bin/sh
printstring=""
first=1
while [ "$1" != "" ]; do
    if [ $first == 1 ]; then
        printstring="$"$1
        first=0
    else
        printstring=
    fi
    shift;
done;
```

```
#!/bin/sh
printstring=""
first=1
while [ "$1" != "" ]; do
    if [ $first == 1 ]; then
        printstring="$"$1
        first=0
    else
        printstring=$printstring'"\t"'
    fi
    shift;
done;
```

```
#!/bin/sh
printstring=""
first=1
while [ "$1" != "" ]; do
    if [ $first == 1 ]; then
        printstring="$"$1
        first=0
    else
        printstring=$printstring'"\t"'"$"$1
    fi
    shift;
done;
```

```
#!/bin/sh
printstring=""
first=1
while [ "$1" != "" ]; do
    if [ $first == 1 ]; then
        printstring="$"$1
        first=0
    else
        printstring=$printstring'"\t"'"$"$1
    fi
    shift;
done;
awk -F"\t" '{print '$printstring'}'
```

```
#!/bin/sh
printstring=""
first=1
while [ "$1" != "" ]; do
    if [ $first == 1 ]; then
        printstring="$"$1
        first=0
    else
        printstring=$printstring'"\t"'"$"$1
    fi
    shift;
done;
awk -F"\t" '{print '$printstring'}'
```

Alias, function or script?

Alias

Just replaces text
Best for changing defaults of commands

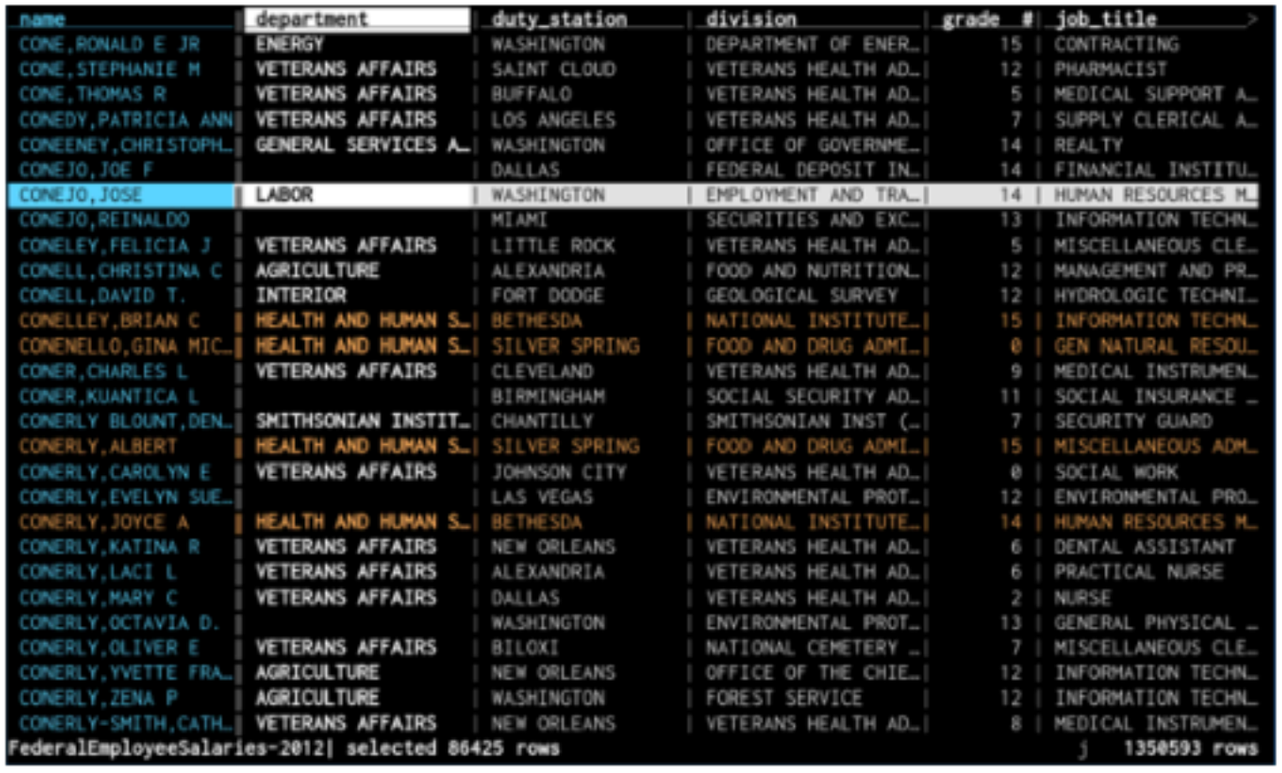
Function

Takes arguments
Is in memory
Executes quickly
(autoload in some shells)

Script

Spawns new process
Can write in any language

~$



VisiData (`vd` on the command-line) is an open-source multitool for interacting with tabular data from within the terminal. It's great for investigative journalists, data scientists, unix command-line users, and anyone else who wants to quickly look at or manipulate data.

http://visidata.org/

(vi for spreadsheets)

Visidata tutorial:

https://jsvine.github.io/intro-to-visidata/

Exercises

Implement an alias, function or script
for something that you do frequently
(and share it!)

A short explanation of the conventions used for the following exercises

Questions

are on blue slides

Answers

are on yellow slides

Hints and explanations

are on green slides

Pitfalls/traps/avoid are on red slides

Exercises

are on purple slides

Code
is on white slides

How to not print output on STDERR?

2> /dev/null