Technologies For Data Manipulation And Analysis

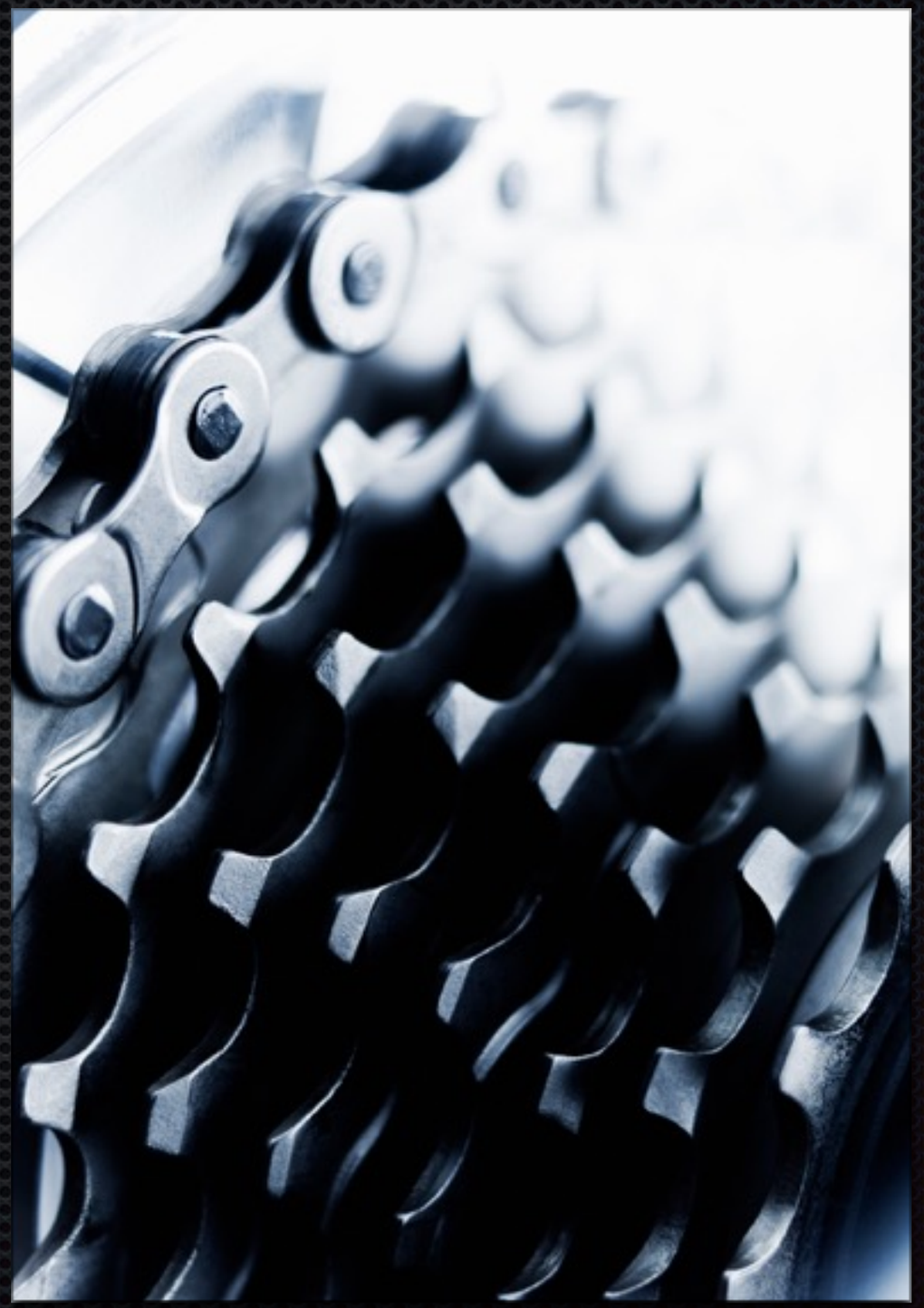
There's More than One Way to Groom a Cat(alog)

Presented by Annie Glerum and Kathryn Lybarger

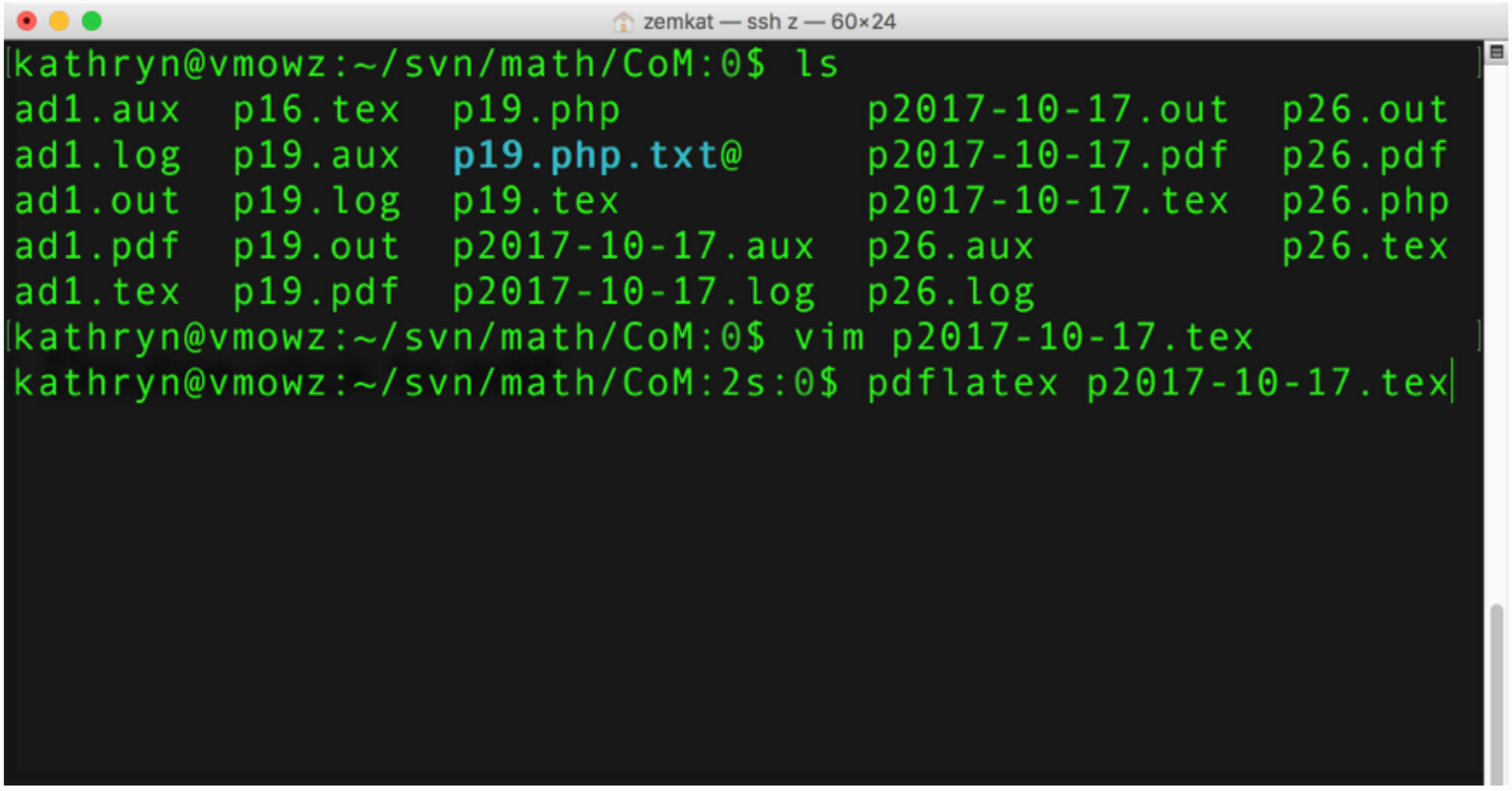
OLAC Preconference October 26, 2017

Tools

Command Line



“How retro!”


zemkat — ssh z — 60×24
kathryn@vmowz:~/svn/math/CoM:0$ ls
ad1.aux p16.tex p19.php p2017-10-17.out p26.out
ad1.log p19.aux p19.php.txt@ p2017-10-17.pdf p26.pdf
ad1.out p19.log p19.tex p2017-10-17.tex p26.php
ad1.pdf p19.out p2017-10-17.aux p26.aux p26.tex
ad1.tex p19.pdf p2017-10-17.log p26.log
kathryn@vmowz:~/svn/math/CoM:0$ vim p2017-10-17.tex
kathryn@vmowz:~/svn/math/CoM:2s:0$ pdflatex p2017-10-17.tex

Why the command line?

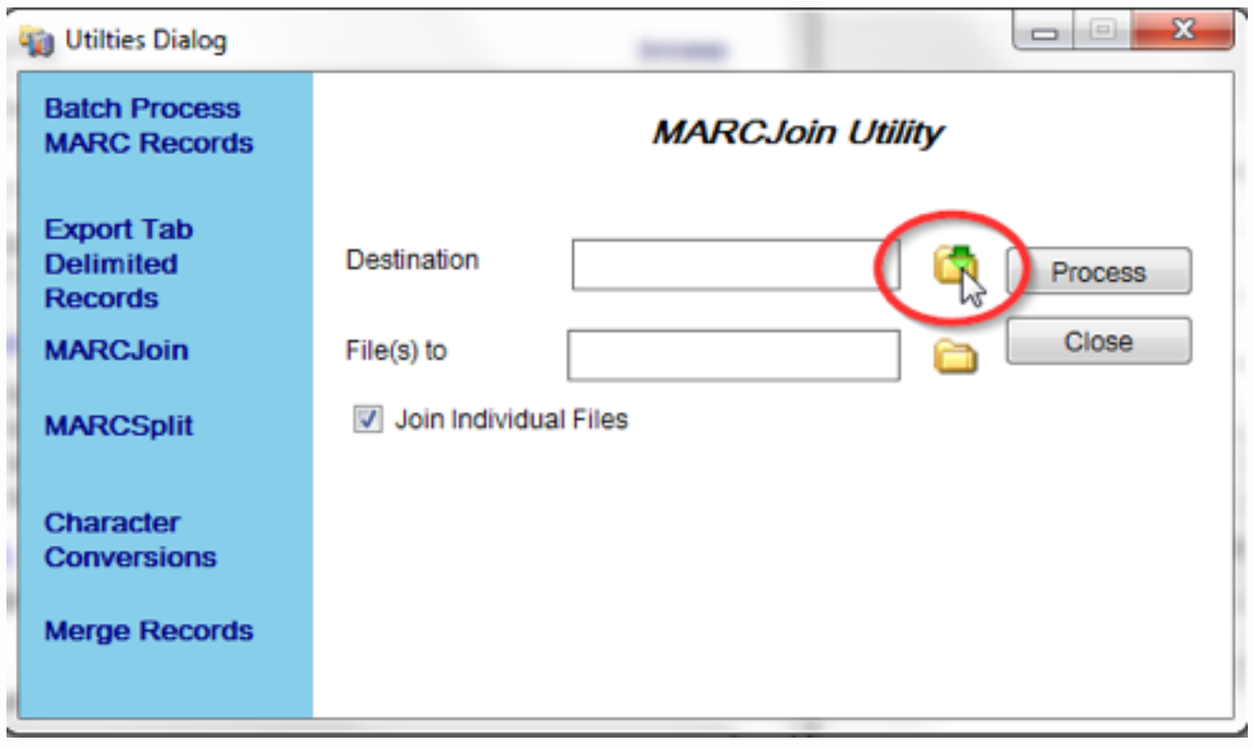
- Quick access to powerful flexible tools
- Tools work well with large files
- Tools are fast, have very little overhead
- Common tools, available on most platforms:
  - unix/linux, mac, Windows 10 (earlier Windows with cygwin)
- “shell access”, “server access”

Example: convert image file from TIFF to PNG

- One method:
  - Double-click portrait.tiff to open in Photoshop
  - (Wait for Photoshop to load all of its plugins)
  - Save as portrait.png
- Command line:
  - `convert portrait.tiff portrait.png`

Example: combine MARC files into one big file

- MarcEdit:



- Command line:
  - `cat file*.mrc > all.mrc`

Are command line programs better?

- Windows-based programs may be friendlier
- Different programs have different features
- For tasks you perform frequently, command-line programs are easy to automate

Sometimes!

- `rsync` - file copying tool
  - command line, so not running in a window
  - good for large directories with subdirectories
  - can copy over a network
  - interruptible! stop a copy, restart where it left off
  - similar windows program — `robocopy`

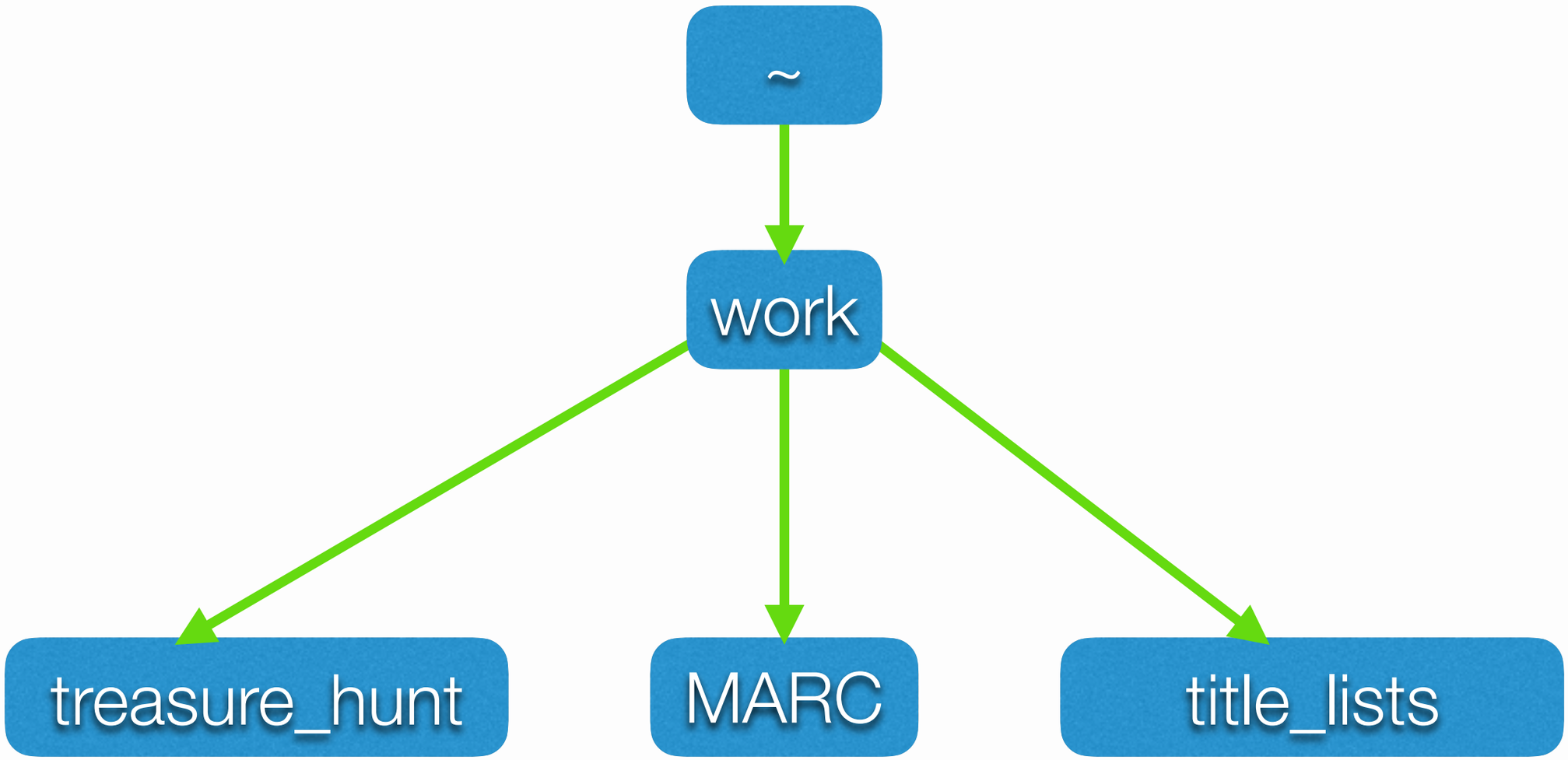
Follow along on the web

- Visit http://olac2017.zemkat.org/
- Login with your Google account
- Start your server
- Open a terminal

Command structure

- `convert -size 320x90 image.png image.jpg`
- command
  - what program are you running?
- switches (starting with hyphen)
  - how should the program behave?
- arguments
  - what (files?) are you running the program on?

Directory structure



path: ~/work/MARC

The Prompt

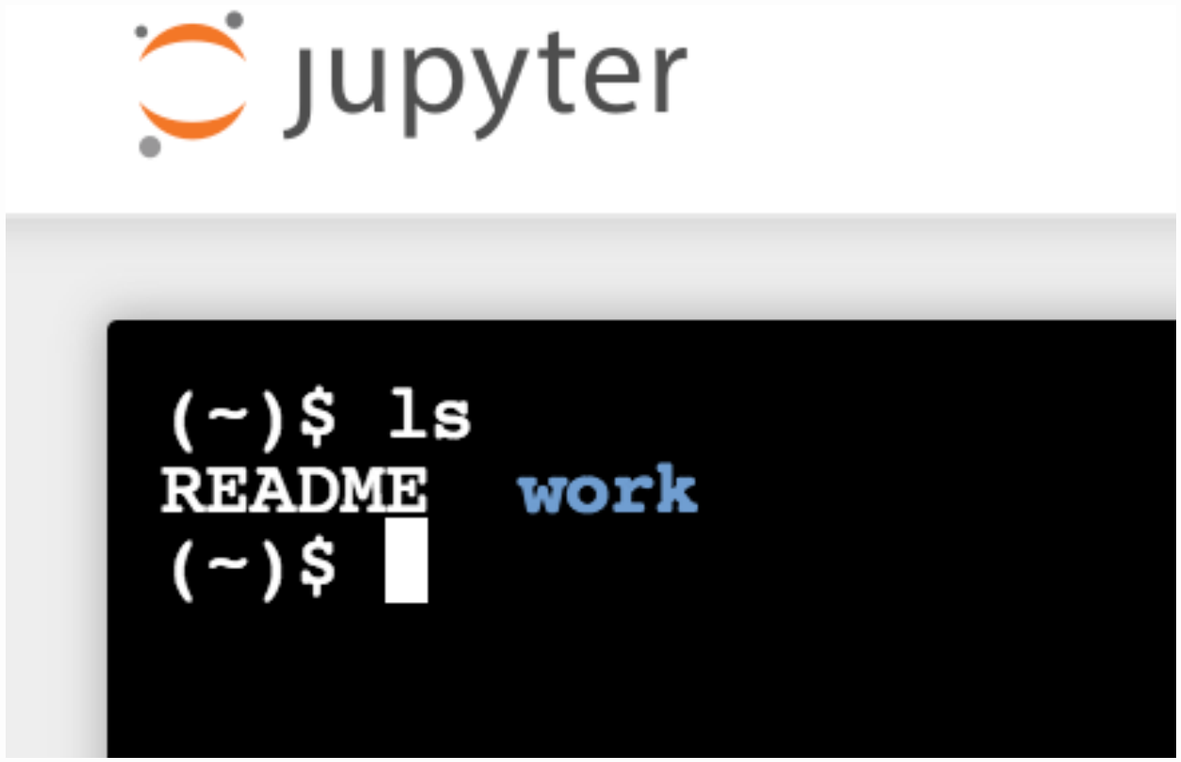
$ — a common prompt, ready for your input

(~) — present working directory

Command: `ls`
list directory contents

- Try this:
  - `ls`
- What do you see?
- What do the colors mean?

What does `ls` do?



White means regular files

Blue means it is a directory/folder

Command: `cat`
print file to screen

- Short for con**cat**enate
  - (as in the MARC example)
- Try this:
  - `cat README`

What does `cat` do?

```
(~)$ cat README

Hello! Welcome to the Jupyter terminal.

Jupyter is a free open-source web software system that provides easy
access to many programming languages, including Julia, Python, and R
(hence the name).

We'll be using Jupyter in this workshop to practice using the bash
command line, including editing running a Python program.

(~)$
```

This is a very readable plain text file!

You can still run cat on other file types (like MARC) but they will be ugly

Command: `cd`
change directory

- Try this:
  - `cd work`
- What is your present working directory now?
- What files or directories are in there?
- (How do you get back to where you were?)

What does **`cd`** do?

```
(~)$ cd work
(~/work)$ ls
MARC  title_lists  treasure_hunt
(~/work)$
```

Notice that the prompt has changed to show **`~/work`** as your present working directory

Special symbol: . .
dot dot = parent directory

- Try this:
  - `cd ..`
- What is your present working directory now?
- How do you get back into the work directory?

Tab completion

- Try this:
  - `cd w<tab>`
- While typing in a bash shell, if you type part of a program or file/directory name and press <tab>, it will often complete it

Activity: Treasure hunt

- From the `work` directory
- Change to the `treasure_hunt` directory
- Many subdirectories have a `treasure` file, but is there really treasure there? Look in the file to find out!
- (Let me know when you've found the treasure!)

Treasure hunting

```
(~)$ cd work
(~/work)$ ls
MARC  title_lists  treasure_hunt
(~/work)$ cd treasure_hunt/
(~/work/treasure_hunt)$ ls
five  four  one  three  treasure  two
(~/work/treasure_hunt)$ cat treasure
Nope, not here!
(~/work/treasure_hunt)$ cd two
(~/work/treasure_hunt/two)$ ls
bear  hippo  lion  tiger  treasure
(~/work/treasure_hunt/two)$ cat treasure
Nope, not here!
(~/work/treasure_hunt/two)$ cd ..
(~/work/treasure_hunt)$
```

Treasure hunt debrief

- Did you hunt through the directory tree in any particular order?

- Is there a better way to find the treasure?

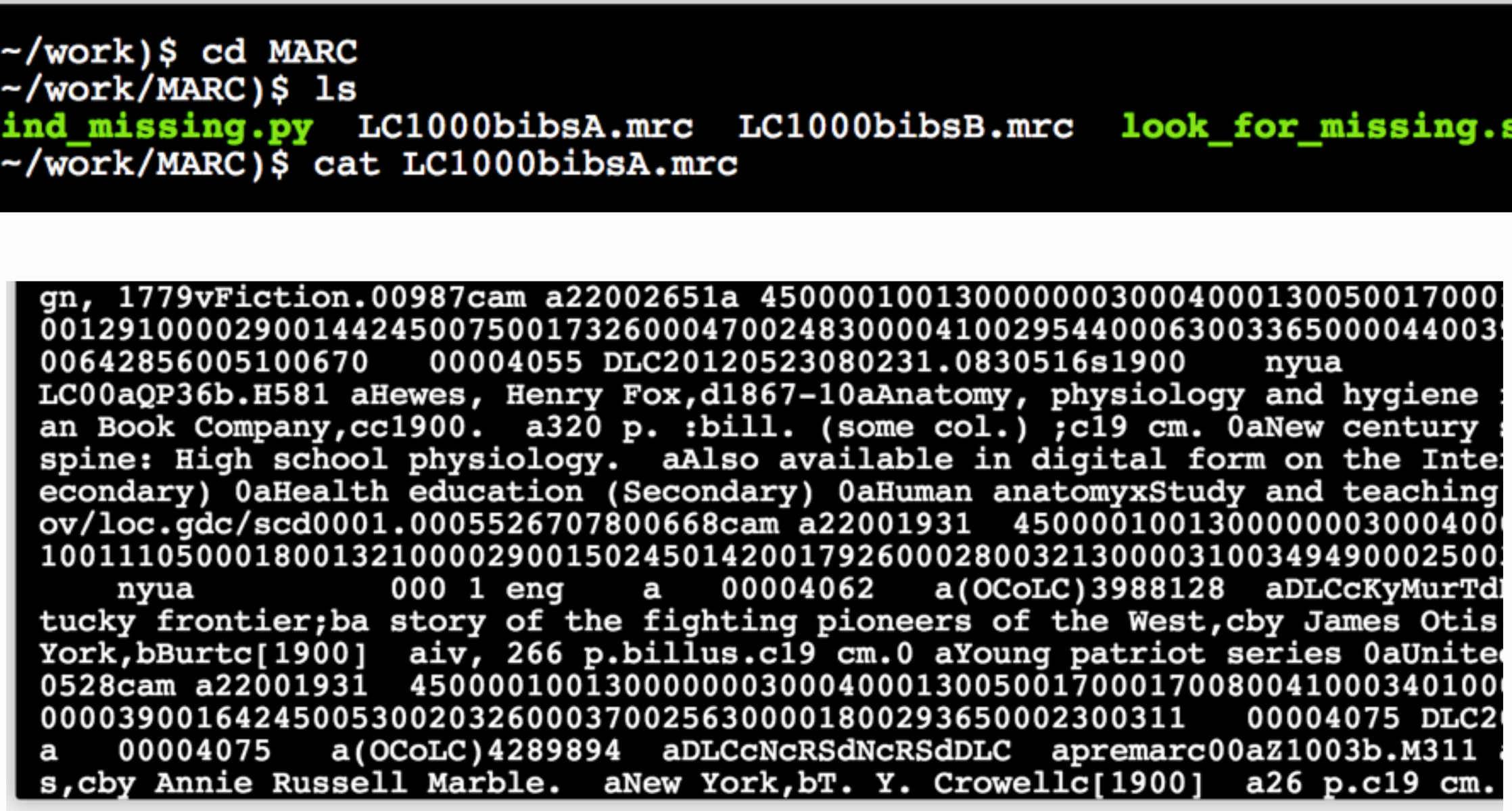
Looking at MARC

- Under the `work` directory, go to the `MARC` directory
- Look at `LC1000bibsA.mrc`

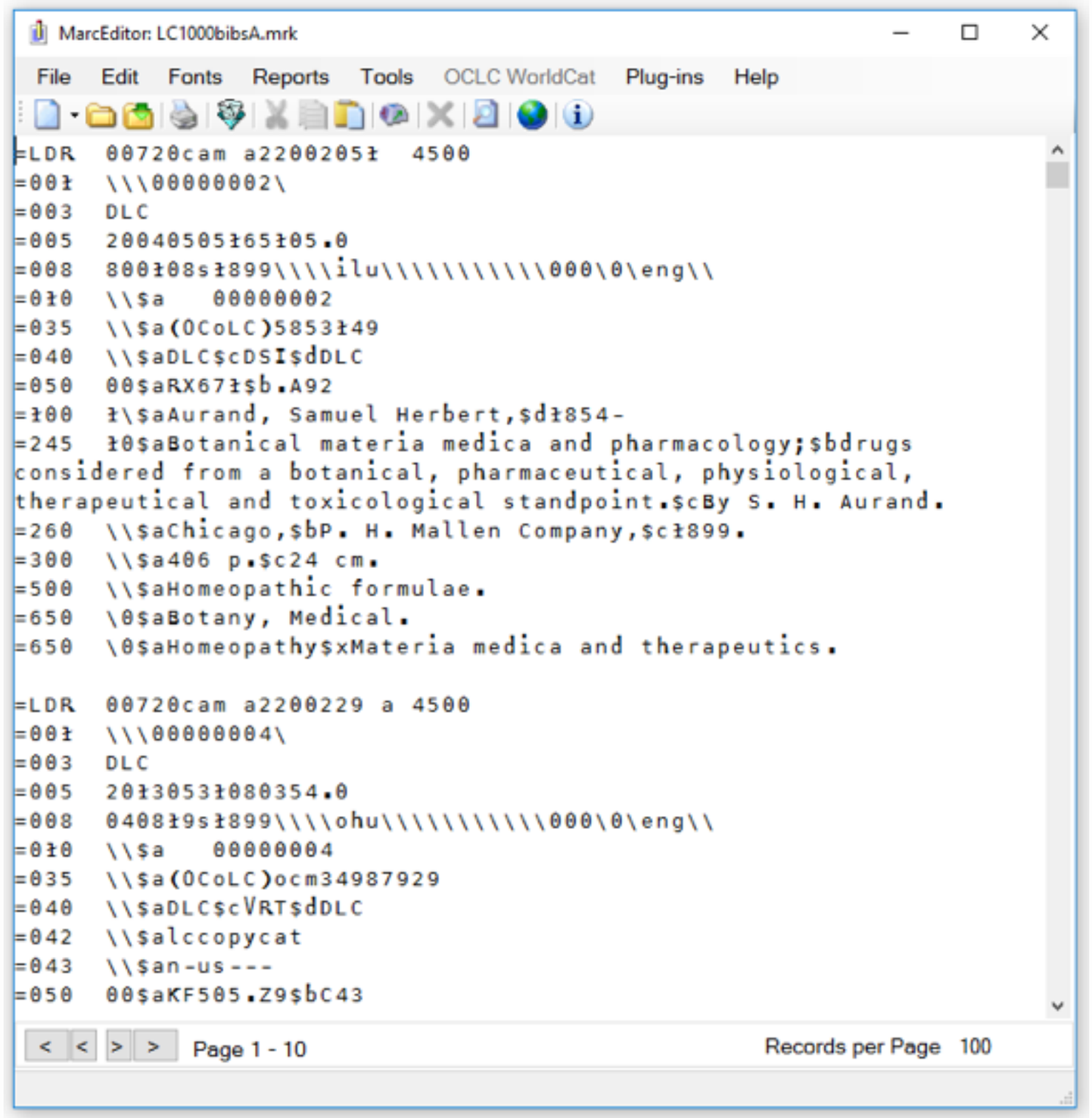
- What does the file look like?
  - Can you read it?
  - Would it be easy to edit?

What does MARC look like?

```
(~/work)$ cd MARC
(~/work/MARC)$ ls
find_missing.py  LC1000bibsA.mrc  LC1000bibsB.mrc  look_for_missing.sh
(~/work/MARC)$ cat LC1000bibsA.mrc
```



MarcEdit


MarcEditor: LC1000bibsA.mrk
File Edit Fonts Reports Tools OCLC WorldCat Plug-ins Help
=LDR 00720cam a2200205ł 4500
=001 \\\00000002\
=003 DLC
=005 20040505ł65ł05.0
=008 800ł08sł899\\\\ilu\\\\\\\\\\\\000\0\eng\\
=0ł0 \\$a 00000002
=035 \\$a(OCoLC)5853ł49
=040 \\$aDLC$cDSI$dDLC
=050 00$aRX67ł$b.A92
=ł00 ł\$aAurand, Samuel Herbert,$dł854-
=245 ł0$aBotanical materia medica and pharmacology;$bdrugs
considered from a botanical, pharmaceutical, physiological,
therapeutical and toxicological standpoint.$cBy S. H. Aurand.
=260 \\$aChicago,$bP. H. Mallen Company,$cł899.
=300 \\$a406 p.$c24 cm.
=500 \\$aHomeopathic formulae.
=650 \0$aBotany, Medical.
=650 \0$aHomeopathy$xMateria medica and therapeutics.
=LDR 00720cam a2200229 a 4500
=001 \\\00000004\
=003 DLC
=005 20ł3053ł080354.0
=008 0408ł9sł899\\\\ohu\\\\\\\\\\\000\0\eng\\
=0ł0 \\$a 00000004
=035 \\$a(OCoLC)ocm34987929
=040 \\$aDLC$cVRT$dDLC
=042 \\$alccopycat
=043 \\$an-us---
=050 00$aKF505.Z9$bC43
< < > > Page 1 - 10
Records per Page 100

Software: YAZ

- YAZ is a command line tool you can use to work with MARC
  - convert between MARC formats (binary, line, MarcXML)
  - convert between MARC-8 and UTF-8 (unicode)
  - Z39.50 / SRU / Solr protocols
- Try this:
  - `yaz-marcdump -i marc -o line LC1000bibsA.mrc`
- Is that better?

What does `yaz-marcdump` do?

Similar to MarcEdit's format; a little different

```
(~/work/MARC)$ yaz-marcdump -i marc -o line LC1000bibsA.mrc
```

```
00528cam a22001931  4500
001    00004075
003 DLC
005 20050903151505.0
008 781012s1900    nyu           000 0 eng
010    $a    00004075
035    $a (OCoLC)4289894
040    $a DLC $c NcRS $d NcRS $d DLC
042    $a premarc
050 00 $a Z1003 $b .M31
100 1  $a Marble, Annie Russell, $d 1864-1936.
245 10 $a Books that nourish us, $c by Annie Russell Marble.
260    $a New York, $b T. Y. Crowell $c [1900]
300    $a 26 p. $c 19 cm.
650  0 $a Books and reading.
```

Command line editing

- Our next command to type is:
  - `yaz-marcdump -i marc -o line LC1000bibsA.mrc > LC1000bibsA.mrk`
- Try pressing up-arrow to bring the last thing you typed back into the command line
  - Then keep typing the rest of the command
  - You can also move around in it with left-right arrows, backspace, etc

Input / Output redirection

- `> filename`
  - Following your command with this means don't put it on the screen, send it to that file
- `| program`
  - Following your command with this means don't put it on the screen, use as input for next program
  - This vertical bar is called a "pipe"

Output to a new file

```
(~/work/MARC)$ yaz-marcdump -i marc -o line LC1000bibsA.mrc > LC1000bibsA.mrk
(~/work/MARC)$ ls
find_missing.py   LC1000bibsA.mrk   look_for_missing.sh   MARC-to-Line.sh
LC1000bibsA.mrc   LC1000bibsB.mrc   MARC8-to-UTF8.sh      OpenTextbooks.mrc
(~/work/MARC)$
```

We created a new file:
`LC1000bibsA.mrk`

We can now run other programs on this file

(Why are some of those files green?)

Commands: `head`, `tail`, `less`

- Try these:
  - `head LC1000bibsA.mrk` — view the top ten lines
  - `tail LC1000bibsA.mrk` — view the last ten lines
  - `less LC1000bibsA.mrk` — view the file a screen at a time

What does `head` do?

```
(~/work/MARC)$ head LC1000bibsA.mrk
00720cam a22002051  4500
001     00000002
003 DLC
005 20040505165105.0
008 800108s1899    ilu                000 0 eng
010     $a     00000002
035     $a (OCoLC)5853149
040     $a DLC $c DSI $d DLC
050 00 $a RX671 $b .A92
100 1  $a Aurand, Samuel Herbert, $d 1854-
(~/work/MARC)$
```

By default shows the first 10 lines
head -15 would show the first 15

A slice of the shelf-list

- But what if you're looking for something specific?
- How many titles are classified in the P schedule?
  - (have an LC call number starting with P?)

Excel

	A	B	C	D	E
1	TAG	IND/DELI	DATA		
7828	050	00 $a	P121 $b .S86 1900		
7845	050	00 $a	PA2289 $b .B7		
7866	050	00 $a	PA25 $b .C7 no. 11		
7884	050	00 $a	PA3951 $b .E5 1900		
7902	050	00 $a	PA3951 $b .E5 1900		
7924	050	00 $a	PA6111 $b .S308 1900		
7943	050	00 $a	PA6111.A7 $b P5 1900		
7957	050	00 $a	PA6519.M5 $b G5		
7976	050	00 $a	PA6807.A8 $b M5		
7993	050	00 $a	PC2111 $b .G75		
8014	050	00 $a	PC4111 $b .R33		
8029	050	00 $a	PC4115 $b .A74		
8045	050	00 $a	PC4115 $b .C9		
8062	050	00 $a	PC4640 $b .V5 1900		
8078	050	00 $a	PE1111 $b .B35 1899		
8092	050	00 $a	PE1111 $b .B6635		
8108	050	00 $a	PE1111 $b .F68		

Sheet1

Normal View | 407 of 18151 records found

Regular expression

- What regular expression would find all of the 050 fields?

Regular expression

- What regular expression would find all of the 050 fields?

- `/^050/`

Command: `grep`
print lines matching a pattern

- “global regular expression print”
- print the lines from a file that match a regular expression
- works best with line-based files
- supports different flavors, we’re using PCRE
- Try this:
  - `grep '^050' LC1000bibsA.mrk`

What does `grep` do?

```
(~/work/MARC)$ grep '^050' LC1000bibsA.mrk
```

```
050  00 $a F74.D95 $b B8
050  00 $a E211 $b .B9796
050  00 $a F234.L9 $b C5
050  00 $a PZ3.C545 $b H $a PS3505.L328
050  00 $a RG381 $b .C9
050  00 $a BJ1451 $b .D6
050  00 $a PZ7.D746 $b A
050  00 $a PZ7
050  00 $a QP36 $b .H58
050  00 $a PZ7.K124 $b Ont
050  00 $a Z1003 $b .M31
(~/work/MARC)$
```

Regular expression

- What regular expression would find all of the 050 fields with a call number starting with P?
- (Don’t shout it out quite yet!)

Command: `wc`
count lines, words, bytes

- Try this:
  - `wc -l LC1000bibsA.mrk`
- The `-l` asks how many lines are in the file
- How many did you find?

What does **wc** do?

```
(~/work/MARC)$ wc -l LC1000bibsA.mrk
18151 LC1000bibsA.mrk
(~/work/MARC)$
```

Command: **`wc`**
we can also pipe to it!

- Try this:
- `grep '`*your regex*`' LC1000bibsA.mrk | wc -l`
- How many did you find?

Regular expression

- What regular expression would find all of the 050 fields with a call number starting with P?

- `grep '^050.*\$a P' LC1000bibsA.mrk`

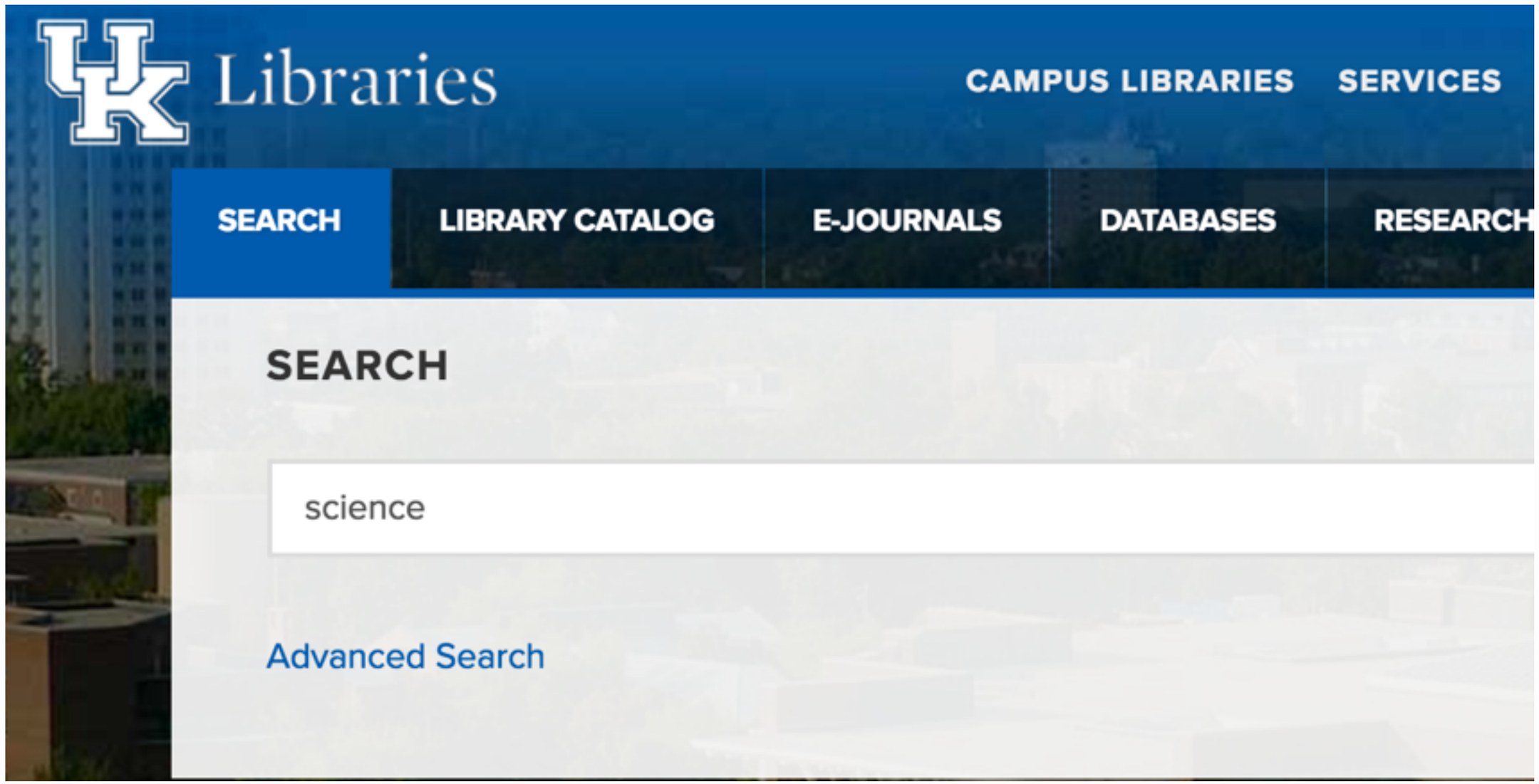
What did that do?

```
(~/work/MARC)$ grep '^050.*\$a P' LC1000bibsA.mrk
050 00 $a PQ2242.M5 $b E5 1900
050 00 $a PA25 $b .C7 no. 11
050 00 $a PS3515.A484 $b B7 1900
050 00 $a PZ3.H435 $b We $a PS1922
050 00 $a PZ7.M265 $b Mai
050 00 $a PS2390.M6 $b L8 1900
050 00 $a PQ2425 $b .V5 1900
050 00 $a PR2833.A2 $b H5
050 00 $a PZ3.C545 $b H $a PS3505.L328
050 00 $a PZ7.D746 $b A
050 00 $a PZ7
050 00 $a PZ7.K124 $b Ont
(~/work/MARC)$
```

Searching MARC

- How many records have “science” in the title?

The Library Catalog



Regular expression

- What regular expression would match lines from this file with “science” in the title?

- /^245.*science/

Command: grep

- We can use a more complex expression
- Try this:
  - `grep '^245.*science' LC1000bibsA.mrk`
- Did that find all of them that the catalog would have?

What did grep do?

```
(~/work/MARC)$ grep '^245.*science' LC1000bibsA.mrk
```

```
245 12 $a A century of science and other essays, $c by John Fiske ..
245 14 $a The cost of living as modified by sanitary science. $c By
245 14 $a The No-din; $b romance, history and science of the pre-his
.
245 10 $a Rahill's Corporation accounting and corporation law; $b a
th in theory and practice, with a digest of the corporation laws of
ining the rules of the New York stock exchange, the San Francisco st
ucers' oil exchange, also the cost and manner of listing stocks, the
brokers on exchange. $c It also contains every form of book used by
245 14 $a The science of higher prisms / $c by Edmund Turney Allen.
245 10 $a Simplified lessons in the science of being, $c by Fanny M.
245 10 $a IGARSS 2000 : $b proceedings : IEEE 2000 International Geo
lanet, the role of remote sensing in managing the environment : 24-2
c [editor, Tammy I. Stein].
245 13 $a An illustrated dictionary of medicine, biology and allied
 literature.
245 12 $a A high school grammar, $b dealing with the science of the
ophy of the changes these have undergone, and present usage respecti
245 12 $a A dictionary of medicine and the allied sciences. $b Compr
ical, pharmaceutical, dental, and veterinary terms $c ...  By Alexan
(~/work/MARC)$
```

Command: grep -i
case insensitive grep

- The original regex was only matching lowercase “science”, so how about Science, SCIENCE, or SCieNCe?
- Try this:
  - `grep -i '^245.*science' LC1000bibsA.mrk`
- (Did that find more?)

What did grep do?

```
(~/work/MARC)$ grep -i '^245.*science' LC1000bibsA.mrk
245 12 $a A century of science and other essays, $c by John Fiske ...
245 14 $a The cost of living as modified by sanitary science. $c By Elle
245 14 $a The No-din; $b romance, history and science of the pre-histori
.
245 10 $a Rahill's Corporation accounting and corporation law; $b a comp
th in theory and practice, with a digest of the corporation laws of all
ining the rules of the New York stock exchange, the San Francisco stock
ucers' oil exchange, also the cost and manner of listing stocks, the met
brokers on exchange. $c It also contains every form of book used by corp
245 14 $a The science of higher prisms / $c by Edmund Turney Allen.
245 10 $a Simplified lessons in the science of being, $c by Fanny M. Har
245 10 $a Science and faith; $b or, Man as an animal, and man as a membe
from the author's ms. by Thomas J. McCormack.
245 10 $a IGARSS 2000 : $b proceedings : IEEE 2000 International Geoscie
lanet, the role of remote sensing in managing the environment : 24-28 Ju
c [editor, Tammy I. Stein].
245 13 $a An illustrated dictionary of medicine, biology and allied scie
 literature.
245 12 $a A high school grammar, $b dealing with the science of the Engl
ophy of the changes these have undergone, and present usage respecting f
245 12 $a A dictionary of medicine and the allied sciences. $b Comprisin
ical, pharmaceutical, dental, and veterinary terms $c ...  By Alexander
(~/work/MARC)$
```

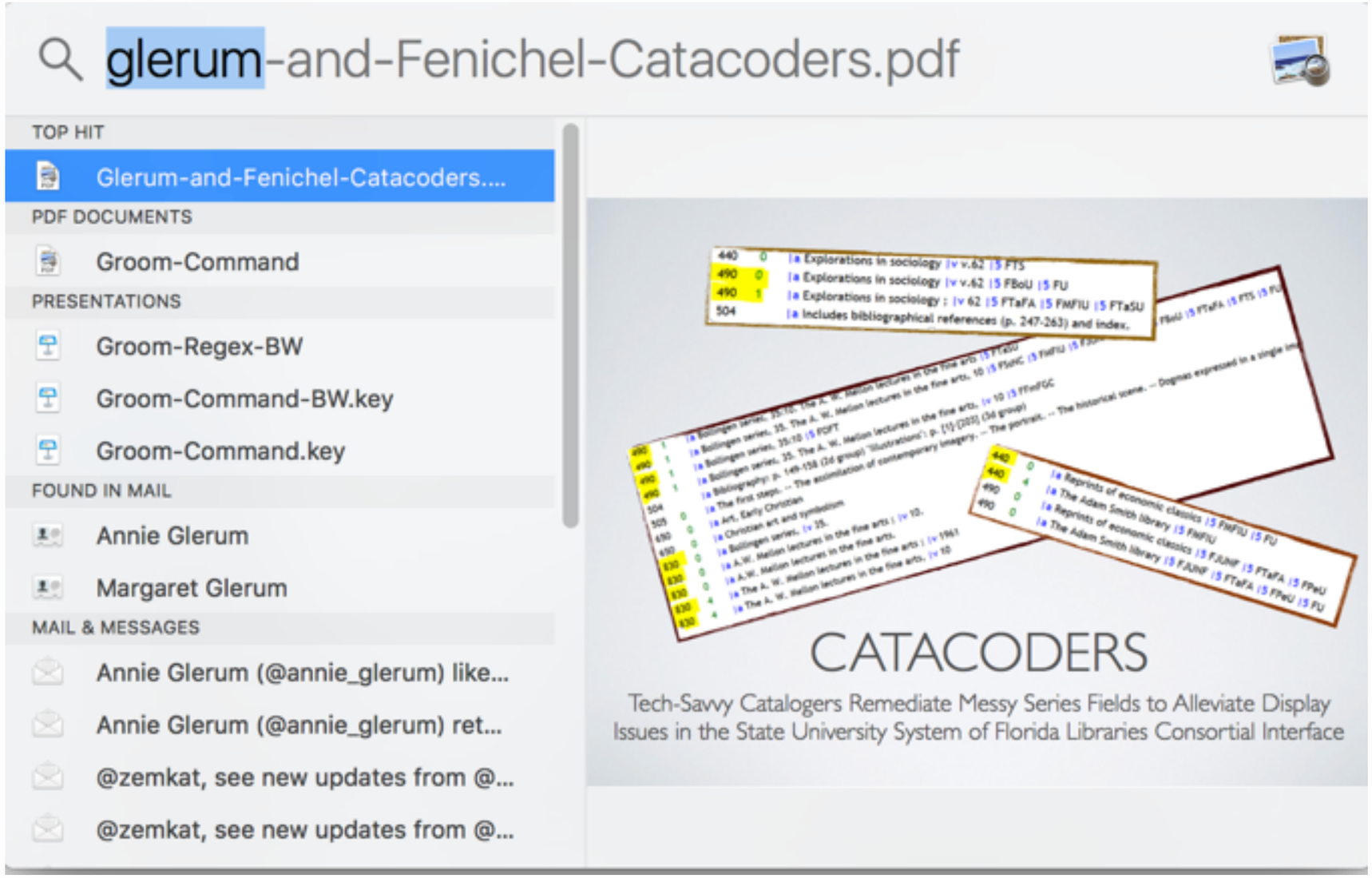
Did it find more?

```
(~/work/MARC)$ grep '^245.*science' LC1000bibsA.mrk | wc -l
10
(~/work/MARC)$ grep -i '^245.*science' LC1000bibsA.mrk | wc -l
11
(~/work/MARC)$
```

Searching multiple files

- Sometimes I have occasion to want to search the MARC files I've loaded
- "This record should be in the catalog… when would I have loaded it?"
- Do I have to convert them all to text first?

Windows Search / Spotlight



Command: grep -l which file matches

- We can use grep on binary MARC, we probably just don't want to see the output
- Try this:
  - `grep -i 'the flowery kingdom' *.mrc`
  - `grep -il 'the flowery kingdom' *.mrc`

What did `grep` do?

```
(~/work/MARC)$ grep -i 'the flowery kingdom' *.mrc
```

```
00040001300500170001700800410003401000170007502000150009204000180010704200090012500
00170030126000340031830000330035249000250038550000220041052001250043265000500055760
00310072780000520075885600940081085600860090400008211 DLC20041202085153.0000128
 a0152021213  aDLCcDLCdDLC  alcac00aPZ7.G9375bIt 200100a[Fic]2211 aGuest, Elissa H
 Elissa Haden Guest ; illustrated by Christine Davenier.30aTrue friends  aSan Dieg
aIris and Walter ;v2  a"Gulliver Books."  aWalter shows Iris how to make friends w
 deal with a problem at school. 0aIris (Fictitious character : Guest)vFiction. 0aW
rsesvFiction. 1aSchoolsvFiction. 1aFriendshipvFiction.1 aDavenier, Christine,eill.
3Contributor biographical informationuhttp://www.loc.gov/catdir/bios/har051/000082
/catdir/description/har021/00008211.html00882cam a22002534a 450000100130000000300
00920400018001070420008001250430012001330500024001450820019001691000018001882450050
03996000018005776500033005950 00008212 DLC20000522095634.0000202s2000    nyua
CcDLCdDLC  apcc  an-mx---00aND259.K33bR86 200000a759.972aB2211 aRummel, Jack.10aFr
aNew York :bCrossroad Pub. Co.,c2000.  a192 p. :bill. ;c22 cm.  aIncludes bibliogr
house -- Carnival of skulls -- The ballerina -- Mexican dionysus -- Colossus of th
 her own terms -- Death is a friend.10aKahlo, Frida. 0aPainterszMexicovBiography.
(~/work/MARC)$
```

What did `grep` do?

```
(~/work/MARC)$ grep -il 'the flowery kingdom' *.mrc
LC1000bibsB.mrc
(~/work/MARC)$
```

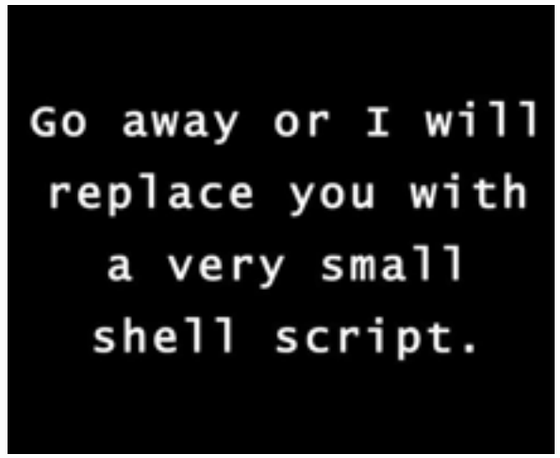
How many MARC records in this file don’t have links?

- How can we answer this?
- (Can we answer this with grep?)

Thinking through this problem…

- If we search for lines that start with either 001 or 856 that will show us all the record IDs and all the links
- We should roughly see: ID, link, ID, maybe a few links, ID link…
- If we see two IDs in a row, that first record didn’t have a link in it

Shell scripts



- A sequence of commands you type on the command line can be saved for later use as a “shell script”
- Even one command with a fancy regex in it
- Try this:
  - `cat look_for_missing.sh`
  - `look_for_missing.sh OpenTextbooks.mrc`

look_for_missing.sh

```
(~/work/MARC)$ cat look_for_missing.sh
#!/bin/bash
#
#       look_for_missing.sh <filename> - convert MARC to line format,
#               browse 001 and 856
#
#       (c)2017 Kathryn Lybarger
#       #sorrynotsorry
#

if [ "$#" -ne 1 ]; then
        echo "Usage: look_for_missing.sh <filename>"
        exit
fi

yaz-marcdump -i marc -o line $1 | grep -P '^(001|856)' | less

(~/work/MARC)$
```

What did the script do?

```
(~/work/MARC)$ look_for_missing.sh OpenTextbooks.mrc
```

```
001 ocn953039218
856 40 $z Access online version $u https://open.umn.edu/
001 ocn911664937
856 40 $z Access online version $u https://open.umn.edu/
001 ocn953263340
856 40 $z Access online version $u https://open.umn.edu/
001 ocn953264176
001 ocn786160475
856 40 $z Access online version $u https://open.umn.edu/
001 ocn953265792
856 40 $u https://open.umn.edu/opentextbooks/BookDetail.
001 ocn851386619
856 40 $z Access online version $u https://open.umn.edu/
001 ocn953700620
```

A better algorithm?

- How would you do it if you were willing to look at each record?
- Look at each record
  - Does it have a 856? If no, write down its name

Command: python
run a python program

- pymarc — python library for working with MARC
- Look at `find_missing.py`
- Try this:
  - `python find_missing.py OpenTextbooks.mrc`
- Did it work?

What did `python` do?

```
(~/work/MARC)$ python find_missing.py OpenTextbooks.mrc
```

```
ocn992721248
ocn992976062
ocn843207614
ocn960723108
ocn985112659
ocn974488881
ocn953031886
ocn953105642
ocn974488513
ocn973884682
ocn974643752
ocn951750571
(~/work/MARC)$
```

Oh dear…

Let's look at the python

```
(~/work/MARC)$ cat find_missing.py
#!/usr/bin/python
#
#       find_missing.py - identify records in a MARC file
#                missing their 856 fields
#
#       (c)2017 Kathryn Lybarger
#

import sys
from pymarc import MARCReader

# Make sure only one filename was passed
if (len(sys.argv) != 2):
    print('Usage: python find_missing.py <filename>')
    sys.exit()

with open(sys.argv[1], 'rb') as fh:
    reader = MARCReader(fh)
    # Loop through records
    for record in reader:
        # If there are no 855 fields, print record number
        if len(record.get_fields('855')) == 0:
            print(record['001'].data)

(~/work/MARC)$
```

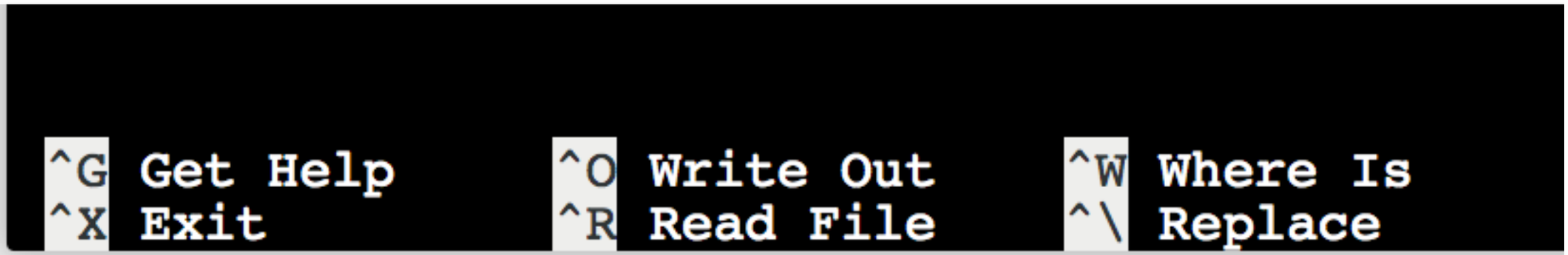
The bug!

```
(~/work/MARC)$ cat find_missing.py
#!/usr/bin/python
#
#       find_missing.py - identify records in a MARC file
#                missing their 856 fields
#
#       (c)2017 Kathryn Lybarger
#

import sys
from pymarc import MARCReader

# Make sure only one filename was passed
if (len(sys.argv) != 2):
    print('Usage: python find_missing.py <filename>')
    sys.exit()

with open(sys.argv[1], 'rb') as fh:
    reader = MARCReader(fh)
    # Loop through records
    for record in reader:
        # If there are no 855 fields, print record number
        if len(record.get_fields('855')) == 0:
            print(record['001'].data)

(~/work/MARC)$
```

Command: nano
a text editor

- Try this:
  - `nano find_missing.py`
- To save the file:
  - `ctrl-o` — save
  - `<enter>`
  - `ctrl-x` — to exit

Using nano



What did python do after you fixed the program?

```
(~/work/MARC)$ python find_missing.py OpenTextbooks.mrc
ocn953264176
(~/work/MARC)$
```

Command: man

- Manual, “man pages”
- Should exist for most common commands
- Tells you:
  - What does the program do?
  - What switches/arguments does it take?
- Try this:
  - `man grep`

What does `man` do?

```
GREP(1)                                                  General Commands Manual

NAME
       grep, egrep, fgrep, rgrep - print lines matching a pattern

SYNOPSIS
       grep [OPTIONS] PATTERN [FILE...]
       grep [OPTIONS] [-e PATTERN]...  [-f FILE]...  [FILE...]

DESCRIPTION
       grep  searches the named input FILEs for lines containing a match to t
       the file "-" is given, grep searches standard input.  By default, grep

       In addition, the variant programs egrep, fgrep and rgrep are the same
       These variants are deprecated, but are provided for backward compatibi

OPTIONS
   Generic Program Information
       --help Output a usage message and exit.

       -V, --version
              Output the version number of grep and exit.
```

Reading the man page

- Navigate with up/down arrows
- <space> will go down one page at a time
- What does the `-r` switch do?

Down in the man page

```
--include=GLOB
       Search only files whose base name matches GLOB (using wi

-r, --recursive
       Read all files under each directory, recursively, follow
       Note  that if no file operand is given, grep searches th
       option.

-R, --dereference-recursive
       Read all files under each directory, recursively.  Follo
```

Recursively: try to match against all files in the directory, all of its subdirectories, all of their subdirectories, etc

A speedier treasure hunt

- We can use this to do the treasure hunt
- Try this:
  - `cd ..`
  - `grep -r 'found' treasure_hunt`

What did `grep` do?

```
(~/work)$ grep -r 'found' treasure_hunt/
treasure_hunt/four/casper/treasure:You found it -- congratulations!
(~/work)$
```

What to catalog?

- We subscribe to some ebook packages that change over time. Throughout the year:
  - some titles are added
  - some titles are deleted
  - some are replaced with new editions
- How to keep current?

Vendor or KnowledgeBase

New Pediatric Edition: Pediatric Clinical Practice Guidelines & Policies

STAT!Ref by STAT!Ref / 481d // keep unread // hide

With some of the highest sought after and used resources in the STAT!Ref platform, the STAT!Ref Core Resources Collection for Pediatrics features some of the most popular resources in this discipline. Within this collection is the newly updated 16th edition of *Pediatric Clinical Practice Guidelines & Policies: A Compendium of Evidence-based Research for Pediatric Practice*.

Keep up with current practice guidelines and policies with the latest, most up-to-date edition of this clinical reference classic. This evidence-based decision-making tool for managing common pediatric conditions has been revised and updated for 2016, with the latest clinical practice guidelines for more than 30 conditions, plus every AAP policy statement, clinical report, and technical report through December 2015.

Updated and expanded for 2016 including:

- Full text of all AAP clinical practice guidelines
- New Periodicity Schedule
- Full text of more than 60 new or revised AAP policies

Pediatric practitioners looking for guidance on managing common conditions in their patients are the intended audience for this Doody's Core Title. Management-level personnel will find the book useful for establishing quality-proven policies for hospital or clinic-based pediatric practices.

STAT!Ref

Type: Selective package
Services: Full Text (Available)
Interface Name: STAT!Ref
Creation Date: 2012-11-25 19:00:00
Modification Date: 2016-12-13 16:49:12
Process type: Acquisition

Other details

Command: diff
show differences between two files

- To check for differences:
  - download a list of titles each periodically
    - check for changes since last month
- Try this in the `title_lists` directory:
  - `diff TitleList-2013-10-30.txt TitleList-2014-02-05.txt`

Diff results

```
(~/work/title_lists)$ diff TitleList-2013-10-30.txt TitleList-2014-02-05.txt
1c1
< AAFP Conditions A to Z (2013)
---
> AAFP Conditions A-Z (2014)
3d2
< ACP Medicine
4a4
> ACP Smart Medicine (SM), Journal Club & AHFS DI® Essentials™
6c6
< AHFS Drug Information® (2013)
---
> AHFS Drug Information® (2014)
10c10
< CPT with RVUs Data File, INGENIX® (2013)
---
> CPT® Data Files, OPTUM™ (2014)
33c33
< Oral Microbiology and Immunology (2006)
---
> Oral Microbiology and Immunology - 2nd Ed. (2014)
41a42
> Scientific American Medicine
(~/work/title_lists)$
```

Left arrow < only in left file
Right arrow > only in right file

Mashcat

- A loose organization of catalogers and coders working together for better communication
- Webinars, twitter chats, in-person conferences
- Slack channels: bash, python