

Lecture 9: Bash Shell & Command Line

LING 1340/2340: Data Science for Linguists

Na-Rae Han

Objectives

- ▶ Finally, shell (bash, zsh)
 - ◆ Running things in command line
 - ◆ Interacting with text files in command line
 - ◆ Regex-based text search using grep

Bash/Zsh shell

▶ What is a "shell"?

- ◆ [https://en.wikipedia.org/wiki/Shell_\(computing\)](https://en.wikipedia.org/wiki/Shell_(computing))
- ◆ Usually refers to the command-line interface (CLI) as opposed to graphical user interface (GUI).
- ◆ **Bash** is the most common flavor of shell in Unix-like OS.

▶ Mac users

- ◆ Mac OS is a Unix-type OS.
- ◆ **Terminal** is a built-in terminal. **Zsh** is the default shell, very similar to bash.

▶ Windows users

- ◆ We installed "**git bash**": a bash environment for running command-line git.
- ◆ As a bonus, it came with pretty much all of **popular Unix command-line tools**!

Shell introduction, navigating

- ▶ Introducing the shell
 - ◆ <http://swcarpentry.github.io/shell-novice/01-intro/>
- ▶ Navigating & working with files and directories
 - ◆ <http://swcarpentry.github.io/shell-novice/02-filedir/>
 - ◆ <http://swcarpentry.github.io/shell-novice/03-create/>
- ▶ We've been doing some of these already, as part of our git routine. You should know:
 - ◆ `.` `..` `~`
 - ◆ `pwd`
 - ◆ `cd`
 - ◆ `ls`
 - ◆ Command-line history with `↑` and `↓`
 - ◆ Using `TAB` for file name completion
 - ◆ Using `Control+C` to quit

Settling in, customizing

- ▶ You can customize your shell via editing:

`.bash_profile`

`.zprofile`

- ▶ In your **home directory**:

- ◆ *your_editor* `.bash_profile` &

- ◆ After adding entries or editing, you should either log back in, or execute

`source .bash_profile`

- ▶ Aliasing is the most common customization method:

```
alias calc='/c/windows/system32/calc.exe'
```

```
alias ls='ls -hF --color=tty' ←
```

← Your favorite shortcuts and command-line options

Mac users: color option is not supported by default unless you customize Terminal.

PATH, which, where

- ▶ We have been occasionally using `pip` to install Python libraries. Where is this `pip`? Which pip are you using?

```
MINGW64:/c/Users/narae
narae@T450s MINGW64 ~
$ which pip
/c/ProgramData/Anaconda3/Scripts/pip

narae@T450s MINGW64 ~
$ which pip3
/c/Program Files (x86)/Python35-32/Scripts/pip3

narae@T450s MINGW64 ~
$ which -a pip
/c/ProgramData/Anaconda3/Scripts/pip
/c/Program Files (x86)/Python35-32/Scripts/pip

narae@T450s MINGW64 ~
$ echo $PATH
/c/Users/narae/bin:/mingw64/bin:/usr/local/bin:/usr/bin:/bin:/mingw64/bin:/usr/bin:/c/Users/narae/bin:/c/WINDOWS/system32:/c/WINDOWS:/c/WINDOWS/System32/wbem:/c/WINDOWS/System32/windowsPowerShell/v1.0:/c/ProgramData/Oracle/Java/javapath:/c/Program Files (x86)/PDFtk Server/bin:/c/Program Files (x86)/Windows Live/Shared:/c/Program Files (x86)/Skype/Phone:/c/ProgramData/Anaconda3:/c/ProgramData/Anaconda3/Scripts:/c/ProgramData/Anaconda3/Library/bin:/c/Program Files (x86)/Pandoc:/c/Program Files/Intel/WiFi/bin:/c/Program Files/Common Files/Intel/WirelessCommon:/c/Program Files (x86)/windows Kits/8.1/Windows Performance Toolkit:/c/Program Files (x86)/Python35-32:/c/Program Files (x86)/Python35-32/Scripts:/c/Users/narae/AppData/Local/Microsoft/WindowsApps:/c/Program Files/Intel/WiFi/bin:/c/Program Files/Common Files/Intel/WirelessCommon:/c/Users/narae/AppData/Local/atom/bin:/usr/bin/vendor_perl:/usr/bin/core_perl
```

1st hit in PATH

PATH, which, where

If you want to install tweepy for this version of python, you can do:

- (1) `pip3 install tweepy`
- (2) `/c/Program Files(x86)/Python35-32/Scripts/pip3 install tweepy`
- (3) cd into `/c/Program Files(x86)/Python35-32/Scripts` directory and then `./pip install tweepy`

```
MINGW64:/c/Users/narae

narae@T450s MINGW64 ~
$ which pip
/c/ProgramData/Anaconda3/Scripts/pip

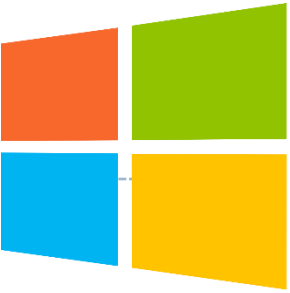
narae@T450s MINGW64 ~
$ which pip3
/c/Program Files (x86)/Python35-32/Scripts/pip3

narae@T450s MINGW64 ~
$ which -a pip
/c/ProgramData/Anaconda3/Scripts/pip
/c/Program Files (x86)/Python35-32/Scripts/pip ← 1st hit in PATH

narae@T450s MINGW64 ~
$ echo $PATH
/c/Users/narae/bin:/mingw64/bin:/usr/local/bin:/usr/bin:/bin:/mingw64/bin:/usr/bin:/c/Users/narae/bin:/c/WINDOWS/system32:/c/WINDOWS:/c/WINDOWS/System32/wbem:/c/WINDOWS/System32/WindowsPowerShell/v1.0:/c/ProgramData/Oracle/Java/javapath:/c/Program Files (x86)/PDFtk Server/bin:/c/Program Files (x86)/Windows Live/Shared:/c/Program Files (x86)/Skype/Phone:/c/ProgramData/Anaconda3:/c/ProgramData/Anaconda3/Scripts:/c/ProgramData/Anaconda3/Library/bin:/c/Program Files (x86)/Pandoc:/c/Program Files/Intel/WiFi/bin:/c/Program Files/Common Files/Intel/WirelessCommon:/c/Program Files (x86)/Windows Kits/8.1/Windows Performance Toolkit:/c/Program Files (x86)/Python35-32:/c/Program Files (x86)/Python35-32/Scripts:/c/Users/narae/AppData/Local/Microsoft/WindowsApps:/c/Program Files/Intel/WiFi/bin:/c/Program Files/Common Files/Intel/WirelessCommon:/c/Users/narae/AppData/Local/atom/bin:/usr/bin/vendor_perl:/usr/bin/core_perl
```

1st hit in PATH

Windows users



- ▶ Because git-bash is not a native command-line shell for Windows (cmd is), there are a few additional wrinkles.
- ▶ Certain programs are designed to run within a console window. Those need to be prefixed with *winpty*. So if you want Python interactive shell:
 - ◆ `winpty python`
- ▶ Pay attention to your directory path.
 - ◆ In git-bash, full path starts with `/c/`.
 - ◆ In cmd (Windows native), it is `C:\...`
 - ◆ In Python, full path can be written as `'C:/...'` or `'C:\\...'` or `r'C:\...'`.
- ▶ Not included:
 - ◆ `more` (use `less` instead)
 - ◆ `man` (you're going to have to Google)

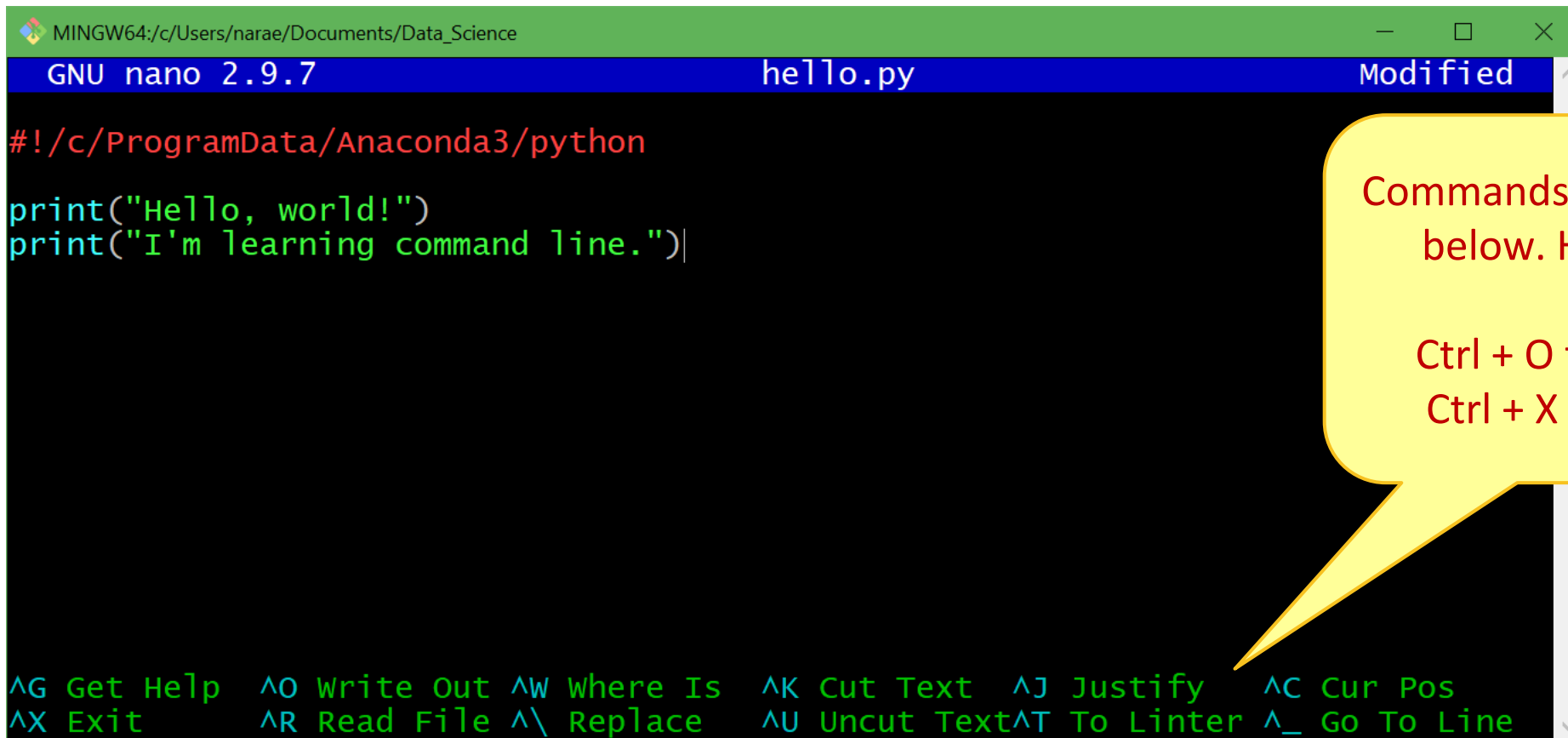
Mac users



- ▶ Add some aliases to your `.zprofile`
- ▶ Like in Windows, you should be able to launch any app that is found in your PATH.
- ▶ Surprise! You also get a handy command for launching *any* GUI application from command-line.
 - ◆ `open -a Application-Name`
 - ◆ <http://osxdaily.com/2007/02/01/how-to-launch-gui-applications-from-the-terminal/>

nano

- ▶ **nano** is a simple command-line based editor. It is found on all Linux distros.
 - ◆ Already present on Macs, and also part of Windows git Bash.



```
MINGW64:/c/Users/narae/Documents/Data_Science
GNU nano 2.9.7 hello.py Modified
#!/c/ProgramData/Anaconda3/python
print("Hello, world!")
print("I'm learning command line.")

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File ^\ Replace  ^U Uncut Text ^T To Linter ^_ Go To Line
```

Commands are listed below. Handy!

Ctrl + O to save
Ctrl + X to exit

Running python script from command-line

1. `python hello.py`

- ◆ Assuming python is in your \$PATH, and hello.py is in your current working directory

2. `hello.py`

- ◆ Assuming your current working directory is in your \$PATH. If not, you should execute `./hello.py`

- ◆ Assuming your script begins with a line (called 'shebang' line):

`#!/systempath/to/python`

- ◆ In my case, it's `#!/c/ProgramData/Anaconda3/python`
- ◆ If your path contains a SPACE... tough luck! (Just kidding, there are ways to handle this.)

Piping and I/O redirection

- ▶ **Piping and I/O redirection** make command-line ever so powerful.
- ▶ For people working mainly with text data (us!), piping enables us to manipulate data on the fly.
 - ◆ `hello.py > out.txt` redirect output to file
 - ◆ `hello.py | wc` pipe output to another application
 - ◆ `hello.py | wc > out.txt` daisy chain!

Also:

- ◆ `<` read in from a file input
- ◆ `>>` *append* to existing file rather than overwriting

Download two files

- ▶ Alice's Adventures in Wonderland

- ◆ <http://www.gutenberg.org/ebooks/11>
- ◆ Download the Plain Text UTF-8 version.
- ◆ Rename the file to "alice.txt"

- ▶ ENABLE word list from Peter Norvig's site:

- ◆ <http://norvig.com/ngrams/>
- ◆ Download "enable1.txt".

← Save them onto your Desktop.

← Then, within bash shell, move the files into your Data_Science directory. (Wait if you are not sure how this is done.)

Files in your Data_Science directory

```
MINGW64:/c/Users/narae/Documents/Data_Science
narae@T450s MINGW64 ~/Documents
$ cd Data_Science/

narae@T450s MINGW64 ~/Documents/Data_Science
$ ls
Class-Practice-Repo/  HW2-Repo/  planets/
Corpus-Resources/    Inaugural-Address-Project/  real_linguistics_data/
HW1-Repo/            foo/

narae@T450s MINGW64 ~/Documents/Data_Science
$ mv ~/Desktop/alice.txt .

narae@T450s MINGW64 ~/Documents/Data_Science
$ mv ~/Desktop/enable1.txt .

narae@T450s MINGW64 ~/Documents/Data_Science
$ ls
Class-Practice-Repo/  Inaugural-Address-Project/  planets/
Corpus-Resources/    alice.txt                    real_linguistics_data/
HW1-Repo/            enable1.txt
HW2-Repo/            foo/

narae@T450s MINGW64 ~/Documents/Data_Science
$ |
```

Examining a text file

▶ `ls (-lahF)`

- ◆ Displays file info

▶ `WC`

- ◆ Displays line count, word count, and character count

▶ `head -n`

- ◆ Displays initial n lines

▶ `tail -n`

- ◆ Displays last n lines

```
MINGW64:~/Documents/Data_Science
narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ ls -l enable1.txt
-rw-r--r-- 1 narae 197121 1916146 Mar 19 12:39 enable1.txt

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ ls -lh enable1.txt
-rw-r--r-- 1 narae 197121 1.9M Mar 19 12:39 enable1.txt

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ wc enable1.txt
172819 172820 1916146 enable1.txt

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ wc alice.txt
3736 29465 173595 alice.txt

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ head enable1.txt
aa
aah
aahed
aahing
aahs
aal
aalii
aaliis
aals
aardvark

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ tail -5 enable1.txt
zymotic
zymurgies
zymurgy
zyzzyva
zyzzyvas

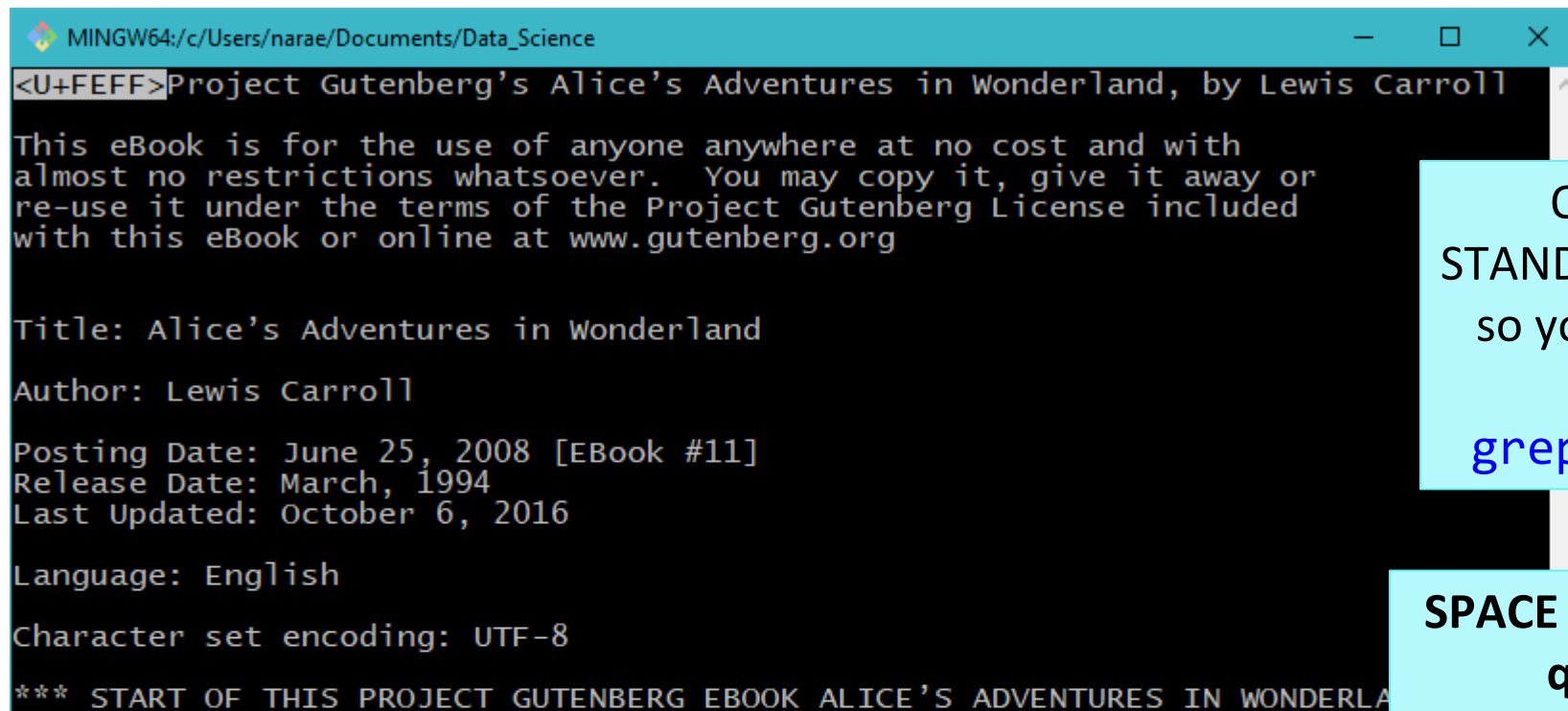
narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ head -5 alice.txt
Project Gutenberg's Alice's Adventures in Wonderland, by Lewis Carroll

This eBook is for the use of anyone anywhere at no cost and with
almost no restrictions whatsoever. You may copy it, give it away or
re-use it under the terms of the Project Gutenberg License included

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$
```

more or less

- ▶ **more** (and **less**) through a text file content, one screen-full at a time. Press **SPACE** for next page, **q** to quit.
 - ◆ Windows users: only **less** is available on git bash.



```
MINGW64:/c/Users/narae/Documents/Data_Science
<U+FEFF>Project Gutenberg's Alice's Adventures in Wonderland, by Lewis Carroll
This eBook is for the use of anyone anywhere at no cost and with
almost no restrictions whatsoever. You may copy it, give it away or
re-use it under the terms of the Project Gutenberg License included
with this eBook or online at www.gutenberg.org

Title: Alice's Adventures in Wonderland
Author: Lewis Carroll
Posting Date: June 25, 2008 [EBook #11]
Release Date: March, 1994
Last Updated: October 6, 2016

Language: English
Character set encoding: UTF-8

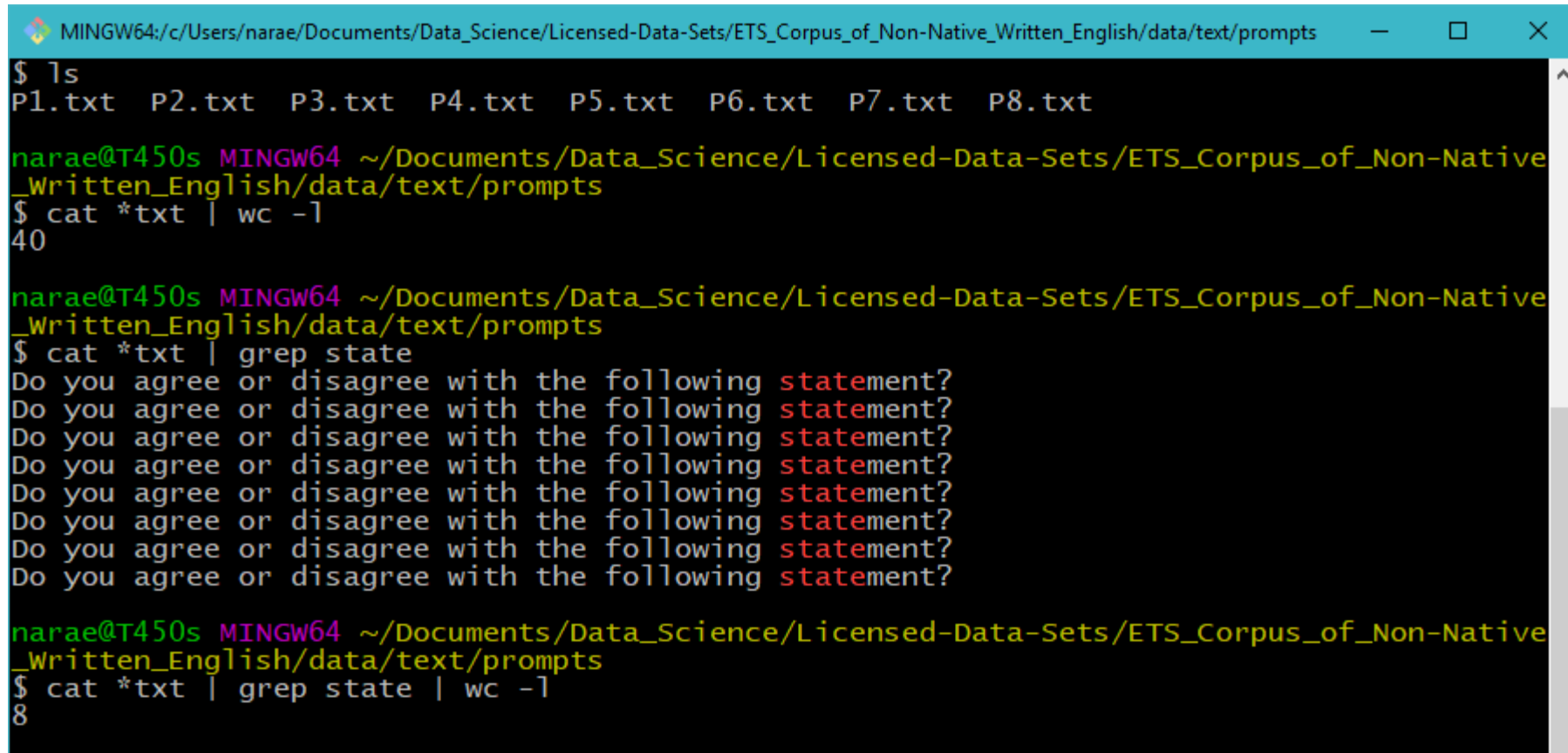
*** START OF THIS PROJECT GUTENBERG EBOOK ALICE'S ADVENTURES IN WONDERLA
```

Often, you **pipe** your STANDARD OUTPUT into more, so you can look through the result, e.g.,
`grep 'q' words | less`

SPACE for next page
q to quit

cat

- ▶ **cat** concatenates text file content and prints on the standard output.
 - ◆ Often used as the first step of piping.
 - ◆ Also useful in concatenating multiple file contents.



```
MINGW64:/c/Users/narae/Documents/Data_Science/Licensed-Data-Sets/ETS_Corpus_of_Non-Native_Written_English/data/text/prompts
$ ls
P1.txt P2.txt P3.txt P4.txt P5.txt P6.txt P7.txt P8.txt

narae@T450s MINGW64 ~/Documents/Data_Science/Licensed-Data-Sets/ETS_Corpus_of_Non-Native_Written_English/data/text/prompts
$ cat *txt | wc -l
40

narae@T450s MINGW64 ~/Documents/Data_Science/Licensed-Data-Sets/ETS_Corpus_of_Non-Native_Written_English/data/text/prompts
$ cat *txt | grep state
Do you agree or disagree with the following statement?
Do you agree or disagree with the following statement?
Do you agree or disagree with the following statement?
Do you agree or disagree with the following statement?
Do you agree or disagree with the following statement?
Do you agree or disagree with the following statement?
Do you agree or disagree with the following statement?
Do you agree or disagree with the following statement?

narae@T450s MINGW64 ~/Documents/Data_Science/Licensed-Data-Sets/ETS_Corpus_of_Non-Native_Written_English/data/text/prompts
$ cat *txt | grep state | wc -l
8
```

grep!!!

▶ grep

- ◆ Searches each line in text for **regular expression** match
- ◆ Excellent intro: <http://www.softpanorama.org/Tols/grep.shtml>

▶ grep -P

- ◆ Only on git-Bash & Linux
 - ◆ **Mac users see next page**
- ◆ Accepts **perl-style** regular expressions
- ◆ Perl-style = Python-style! Can use `\s`, `\d` etc.

```
MINGW64:/c/Users/narae/Documents/Data_Science
narae@T450s MINGW64 ~/Documents/Data_Science
$ grep '\o.*o$' enable1.txt
obligato
obligato
ocotillo
octavo
oho
oleo
olio
oloroso
onto
oratorio
ordo
oregano
ortho
orzo
ostinato
otto
outdo
utecho
outgo
ouzo
overdo
ovoio
oxo

narae@T450s MINGW64 ~/Documents/Data_Science
$ grep '\a.*z$' enable1.txt
abuzz
adz

narae@T450s MINGW64 ~/Documents/Data_Science
$ grep -P '[aeiou]{5,}' enable1.txt
cooeeing
miaoued
miaouing
queueing

narae@T450s MINGW64 ~/Documents/Data_Science
$ |
```

Words with 5+ consecutive "vowel"s

grep on Mac... ☹️



- ▶ Default grep on Mac is an ancient version. No -P option, etc.
- ▶ Alternatives:
 - ◆ GNU grep (has -P)
 - ◆ Pcre grep (perl-style regex)
 - ◆ You will need to install, via **homebrew**.
 - ◆ (Instructions linked on "Learning Resources" section)

```
Jane Eyre@T480s MINGW64 ~/Documents/  
$ grep --version  
grep (GNU grep) 3.1  
Copyright (C) 2017 Free Software Foundation  
License GPLv3+: GNU GPL version 3 or later  
This is free software: you are free to copy, modify, and  
distribute it under the terms of the GNU General Public License  
There is NO WARRANTY, to the extent permitted by law.
```

- ▶ After installation, create an alias in your **.zprofile**

Windows/git-bash
comes with
GNU grep version 3.1,
which is up-to-date

grep is better in color

- ▶ You might want to colorize your grep output.
- ▶ I have `grep` aliased to use color & perl-style regex in my `.bash_profile` configuration file:

```
MINGW64:/c/Users/narae/Documents/Data_Science

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ grep '[aeiou]{5,}' enable1.txt
cooeeing
miaoued
miaouing
queueing

narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ cat ~/.bash_profile
alias more='less'
alias grep='grep -P --color'
```

Mac users: you will want to alias GNU grep or Pcre grep

grep and piping, together

MINGW64:/c/Users/narae/Documents/Data_Science

```
unwarrantable  
unwatchable  
unwearable  
unwinnable  
unworkable
```

```
narae@T450s MINGW64 ~/Documents/Data_Science  
$ grep '^un.*able$' enable1.txt | wc -l  
213
```

Pipe into wc -l to count

```
narae@T450s MINGW64 ~/Documents/Data_Science  
$ grep '^un.*able$' enable1.txt > able.txt
```

Write out to a file

```
narae@T450s MINGW64 ~/Documents/Data_Science  
$ tail -5 able.txt  
unwarrantable  
unwatchable  
unwearable  
unwinnable  
unworkable
```

Take a look at the last 5 lines of file

```
narae@T450s MINGW64 ~/Documents/Data_Science  
$ grep '^in.*able$' enable1.txt >> able.txt
```

Append new search result to file

```
narae@T450s MINGW64 ~/Documents/Data_Science  
$ tail -5 able.txt  
invariable  
investable  
inviabile  
inviolable  
invulnerable
```

Take a look at the last 5 lines of file

```
narae@T450s MINGW64 ~/Documents/Data_Science  
$ wc -l able.txt  
316 able.txt
```

File is now longer

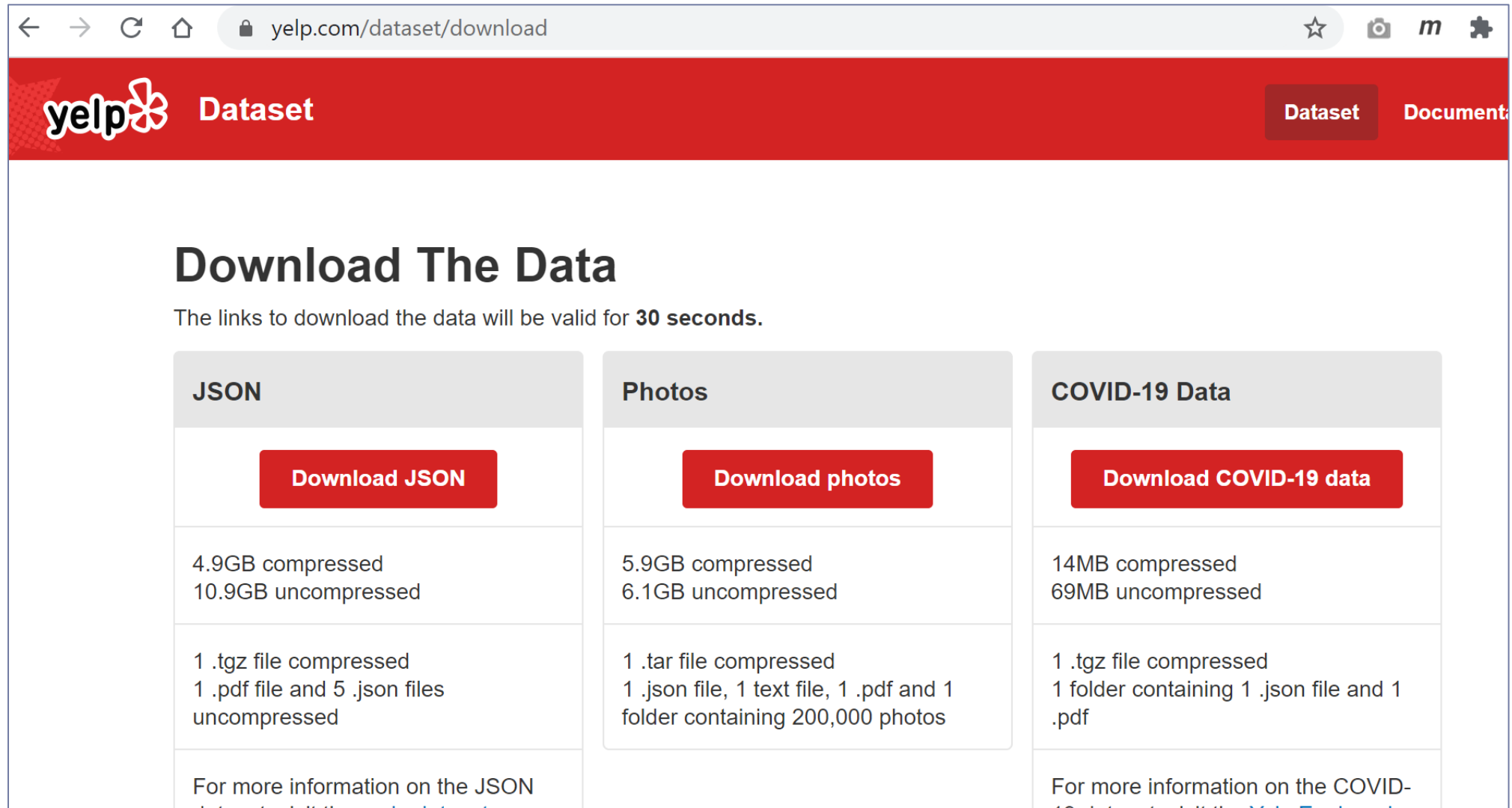
```
narae@T450s MINGW64 ~/Documents/Data_Science  
$ |
```

Not done with grep

- ▶ ... grep continues next class.

Bring on Big Data! The Yelp Dataset

► <https://www.yelp.com/dataset>



The screenshot shows a web browser window with the URL `yelp.com/dataset/download`. The page features a red header with the Yelp logo and the word "Dataset". Below the header, the main heading is "Download The Data", followed by a note: "The links to download the data will be valid for 30 seconds." The page is divided into three columns, each representing a different data type: JSON, Photos, and COVID-19 Data. Each column contains a red button to download the data, along with details about file sizes (compressed and uncompressed) and the contents of the download package.

JSON	Photos	COVID-19 Data
Download JSON	Download photos	Download COVID-19 data
4.9GB compressed 10.9GB uncompressed	5.9GB compressed 6.1GB uncompressed	14MB compressed 69MB uncompressed
1 .tgz file compressed 1 .pdf file and 5 .json files uncompressed	1 .tar file compressed 1 .json file, 1 text file, 1 .pdf and 1 folder containing 200,000 photos	1 .tgz file compressed 1 folder containing 1 .json file and 1 .pdf
For more information on the JSON dataset, visit the JSON Dataset page.		For more information on the COVID-19 dataset, visit the COVID-19 Dataset page.

Working with big data files

```
narae@T480s MINGW64 /d/Corpora/Yelp_dataset
$ ls -lah
total 11G
drwxr-xr-x 1 narae 197121 0 Mar 24 13:55 ./
drwxr-xr-x 1 narae 197121 0 Mar 24 13:55 ../
-rw-r--r-- 1 narae 197121 73K Feb 17 18:50 Dataset_User_Agreement.pdf
-rw-r--r-- 1 narae 197121 119M Jan 28 14:06 yelp_academic_dataset_business.json
-rw-r--r-- 1 narae 197121 380M Jan 28 14:11 yelp_academic_dataset_checkin.json
-rw-r--r-- 1 narae 197121 6.5G Jan 28 14:29 yelp_academic_dataset_review.json
-rw-r--r-- 1 narae 197121 220M Jan 28 14:13 yelp_academic_dataset_tip.json
-rw-r--r-- 1 narae 197121 3.5G Jan 28 14:11 yelp_academic_dataset_user.json

narae@T480s MINGW64 /d/Corpora/Yelp_dataset
$ wc -l yelp_academic_dataset_review.json
8635403 yelp_academic_dataset_review.json

narae@T480s MINGW64 /d/Corpora/Yelp_dataset
$ wc -l yelp_academic_dataset_user.json
2189457 yelp_academic_dataset_user.json
```

Each file is in JSON format, and they are huge:

- ◆ review.json is 6.5GB with 8.6 million records (=lines)
- ◆ user.json is 3.5GB with 2.2 million records (=lines)

- ▶ These are too big to open in most text editors (Notepad++ couldn't.)
- ▶ How to explore them? In command line. [head/tail](#), [grep](#) and [regular expression](#)-based searching.

➔ To-do #11

Wrapping up

▶ To-do #11

- ◆ Fun with big(ish) data -- the Yelp Dataset! <https://www.yelp.com/dataset/>
- ◆ 5Gb zipped, downloading takes 10+ minutes. Allocate enough time for this assignment, especially if you are new to command line.
- ◆ Wed is self-care day. You can submit 1 day late (end of Friday).

▶ Next class

- ◆ More command line, grep, bash shell scripting
- ◆ Supercomputing at CRC
- ◆ HW3 wrap up (finally...??)