Lecture 15: Command Line, Grep, Supercomputing

LING 1340/2340: Data Science for Linguists

Na-Rae Han

Objectives

- Command-line exploration
  - Interacting with text files in command line
  - Regex-based text search using grep
- Follow up of Lecture 9: Data formats, text file encoding & conversion
  - https://naraehan.github.io/Data-Science-for-Linguists-2023/lecture9.pdf
- Supercomputing at CRC
  - Server access through SSH

Bash/Zsh shell

- What is a "shell"?
  - 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.

To find out which shell you're running:

echo $SHELL

- 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
  - https://swcarpentry.github.io/shell-novice/01-intro/
- Navigating & working with files and directories
  - https://swcarpentry.github.io/shell-novice/02-filedir/
  - https://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
    ```

Without &, your terminal becomes unusable until you close your editor.

- 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: -G option for color. You may also have to customize Terminal.

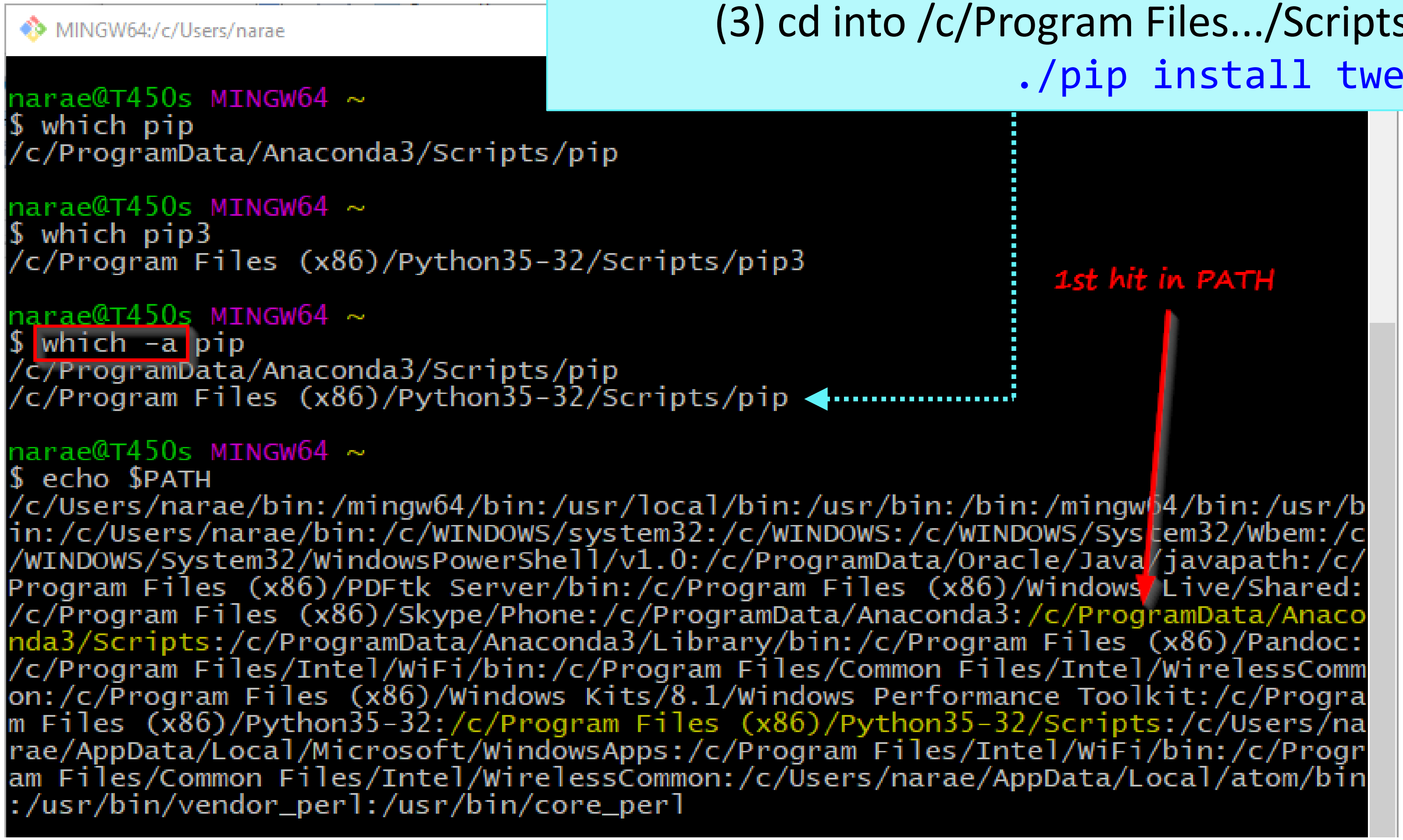
PATH, which, where

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

(1) `pip3 install tweepy`

(2) `/c/Program\ Files.../Scripts/pip install tweepy`

(3) cd into /c/Program Files.../Scripts directory and then `./pip install tweepy`



```
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/b
in:/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/Anaco
nda3/Scripts:/c/ProgramData/Anaconda3/Library/bin:/c/Program Files (x86)/Pandoc:
/c/Program Files/Intel/WiFi/bin:/c/Program Files/Common Files/Intel/WirelessComm
on:/c/Program Files (x86)/Windows Kits/8.1/Windows Performance Toolkit:/c/Progra
m Files (x86)/Python35-32:/c/Program Files (x86)/Python35-32/Scripts:/c/Users/na
rae/AppData/Local/Microsoft/WindowsApps:/c/Program Files/Intel/WiFi/bin:/c/Progr
am 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)

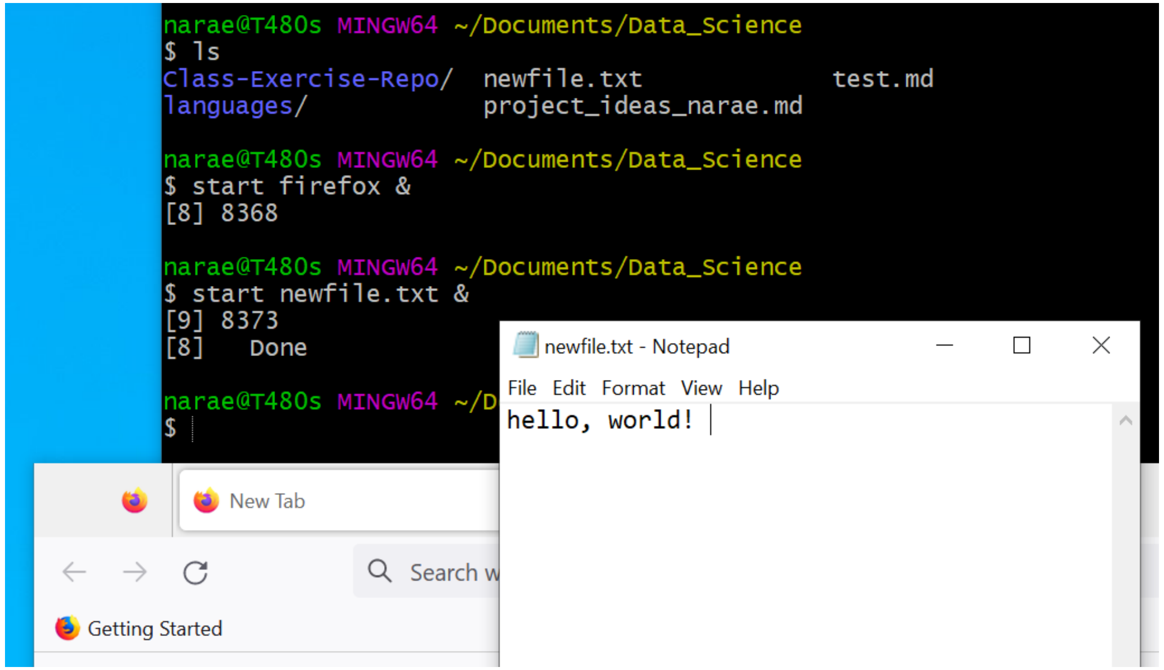
Mac users

- Add some aliases to your `.zprofile`
- Like in Windows, you should be able to launch any app that is found in your OS's PATH variable.

Launching app/file: Windows + OS X perks

Windows

- A handy command for launching *any* file or GUI app from command line
  - `start filename`
  - `start appname`



```
narae@T480s MINGW64 ~/Documents/Data_Science
$ ls
Class-Exercise-Repo/  newfile.txt          test.md
languages/            project_ideas_narae.md

narae@T480s MINGW64 ~/Documents/Data_Science
$ start firefox &
[8] 8368

narae@T480s MINGW64 ~/Documents/Data_Science
$ start newfile.txt &
[9] 8373
[8]   Done

narae@T480s MINGW64 ~/D
$
```

```
hello, world!
```

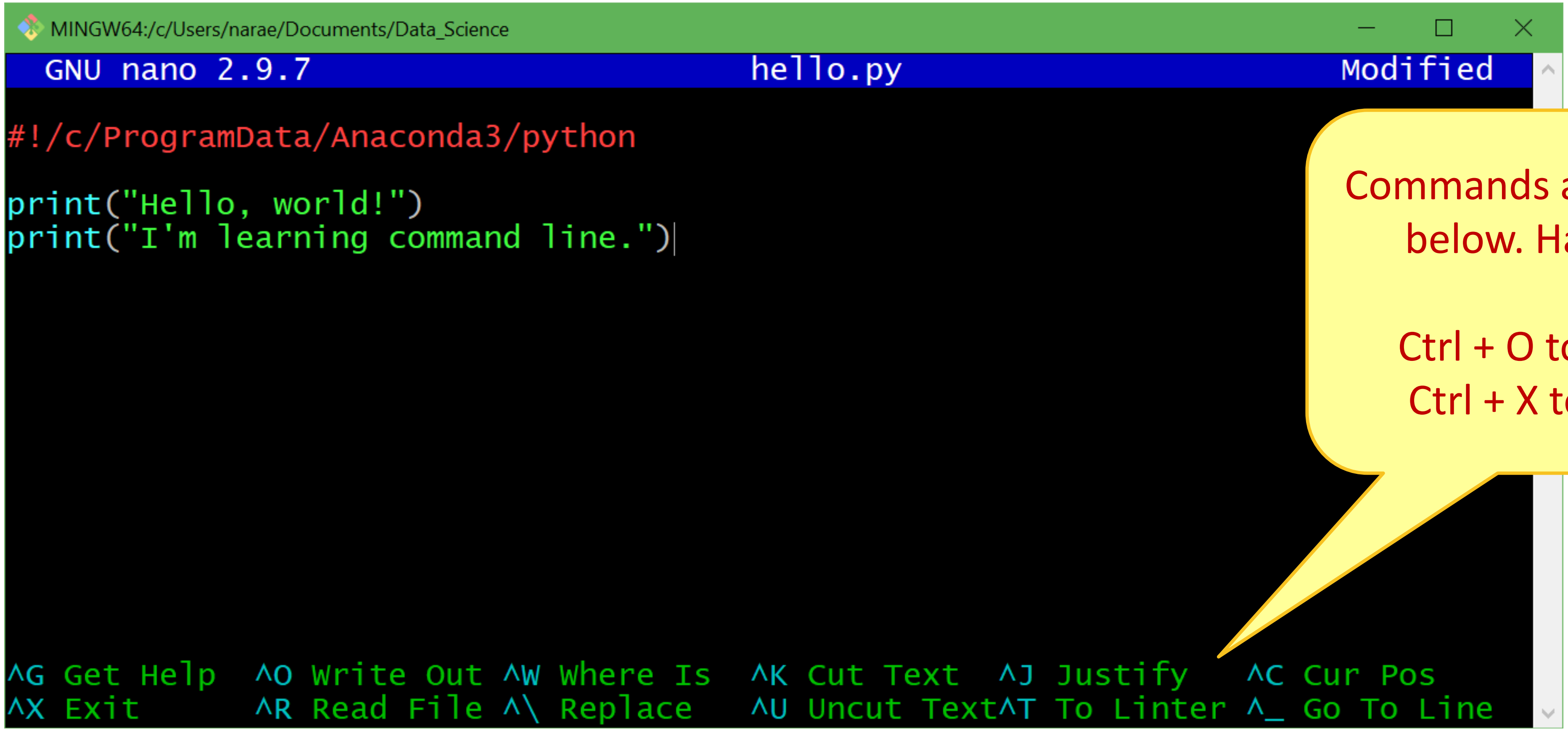
Mac OS

- A handy command for launching *any* GUI application from command line.
  - `open -a Application-Name`
  - https://osxdaily.com/2007/02/01/how-to-launch-gui-applications-from-the-terminal/

NOT part of the bash/zsh!
`start` and `open` are
utilities **provided by your OS**
(Windows, Mac OS)

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.



```
  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 around it.)

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

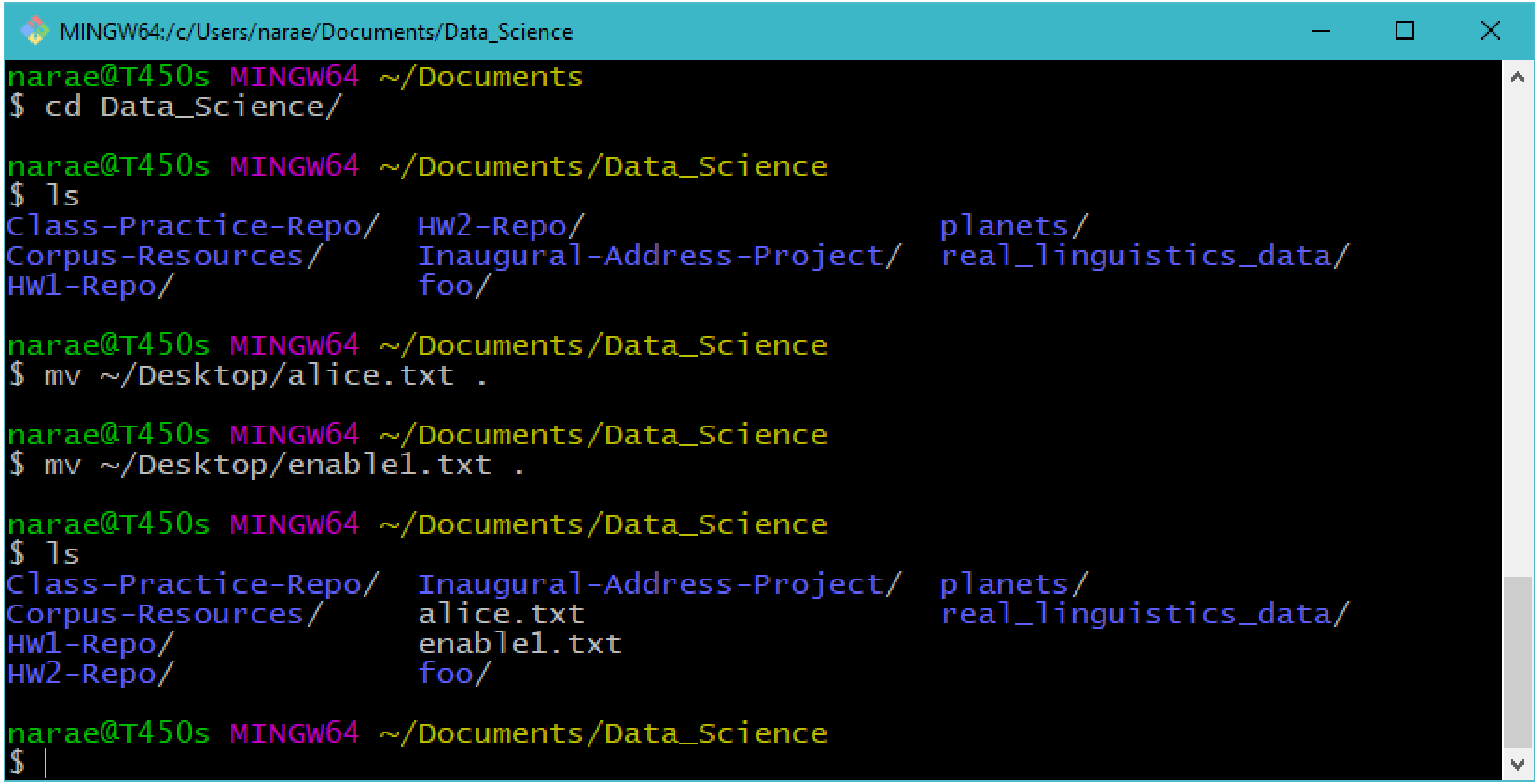
- Alice's Adventures in Wonderland
  - https://www.gutenberg.org/ebooks/11
  - Download the Plain Text UTF-8 version.
  - Rename the file to "alice.txt"
- ENABLE word list from Peter Norvig's site:
  - https://norvig.com/ngrams/
  - Download "enable1.txt"

← 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.)

← In command line, find out as much you can about these files.

Files in your Data_Science directory

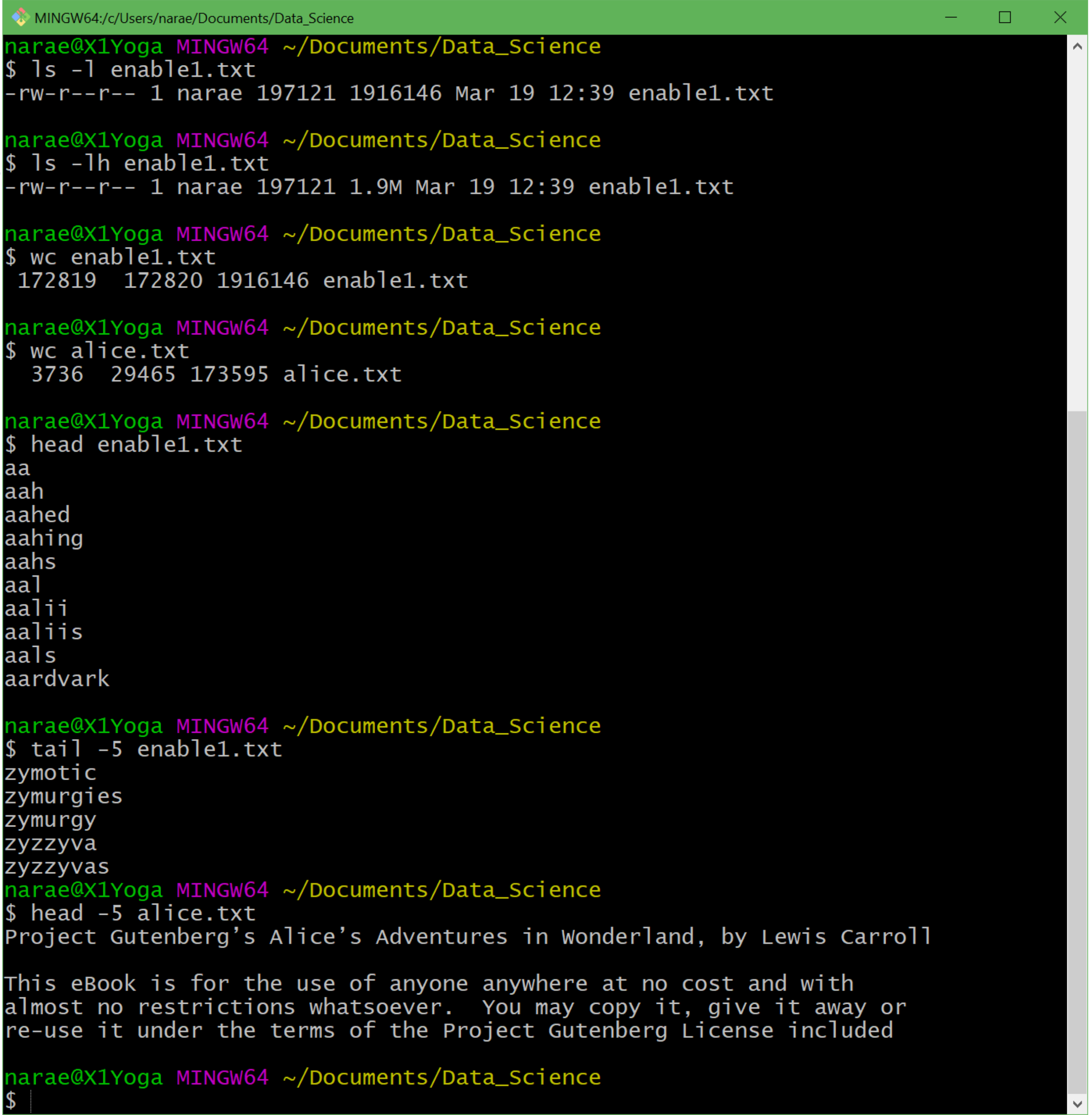


```
narae@T450s MINGW64 ~/Documents
$ cd Data_Science/

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
  - Also: -G (Mac OS)
- `file (-i)`
  - Displays character encoding, line ending
- `wc`
  - Displays line count,

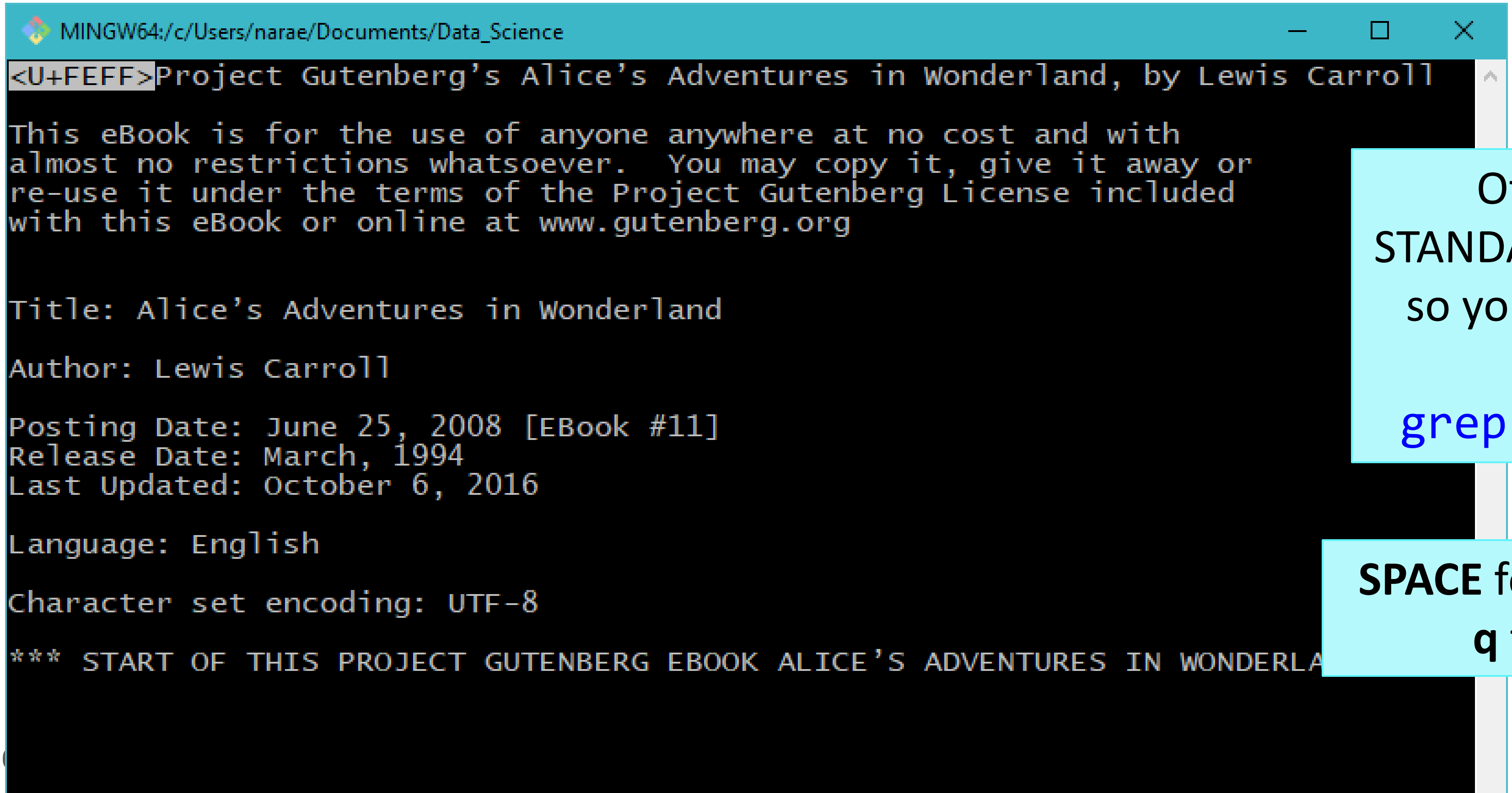
  word count, and character count
- `head -n`
  - Displays initial n lines
- `tail -n`
  - Displays last n lines



```
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.



```
<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
```

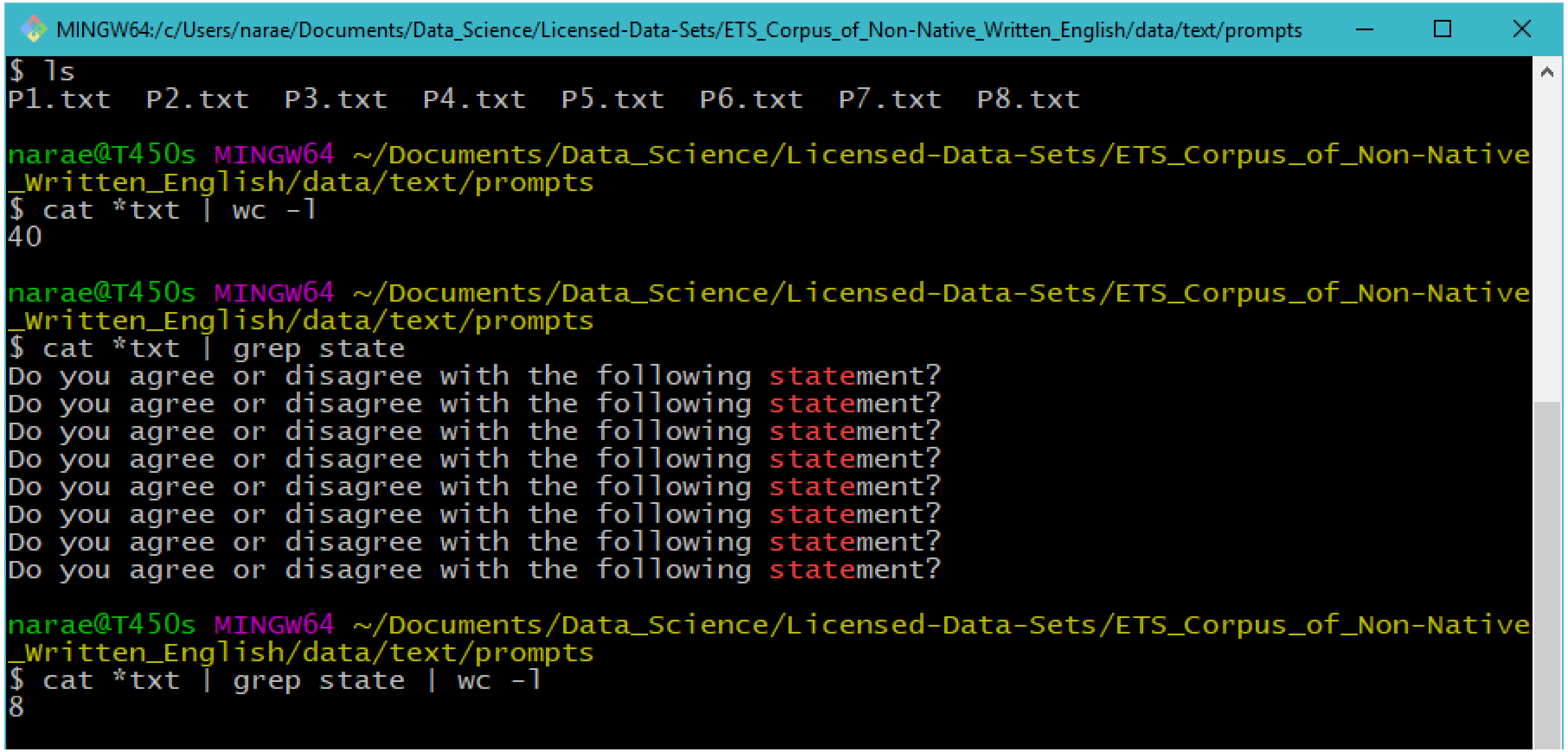
Often, you **pipe** your STANDARD OUTPUT into more, so you can look through the result, e.g.,

```
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.

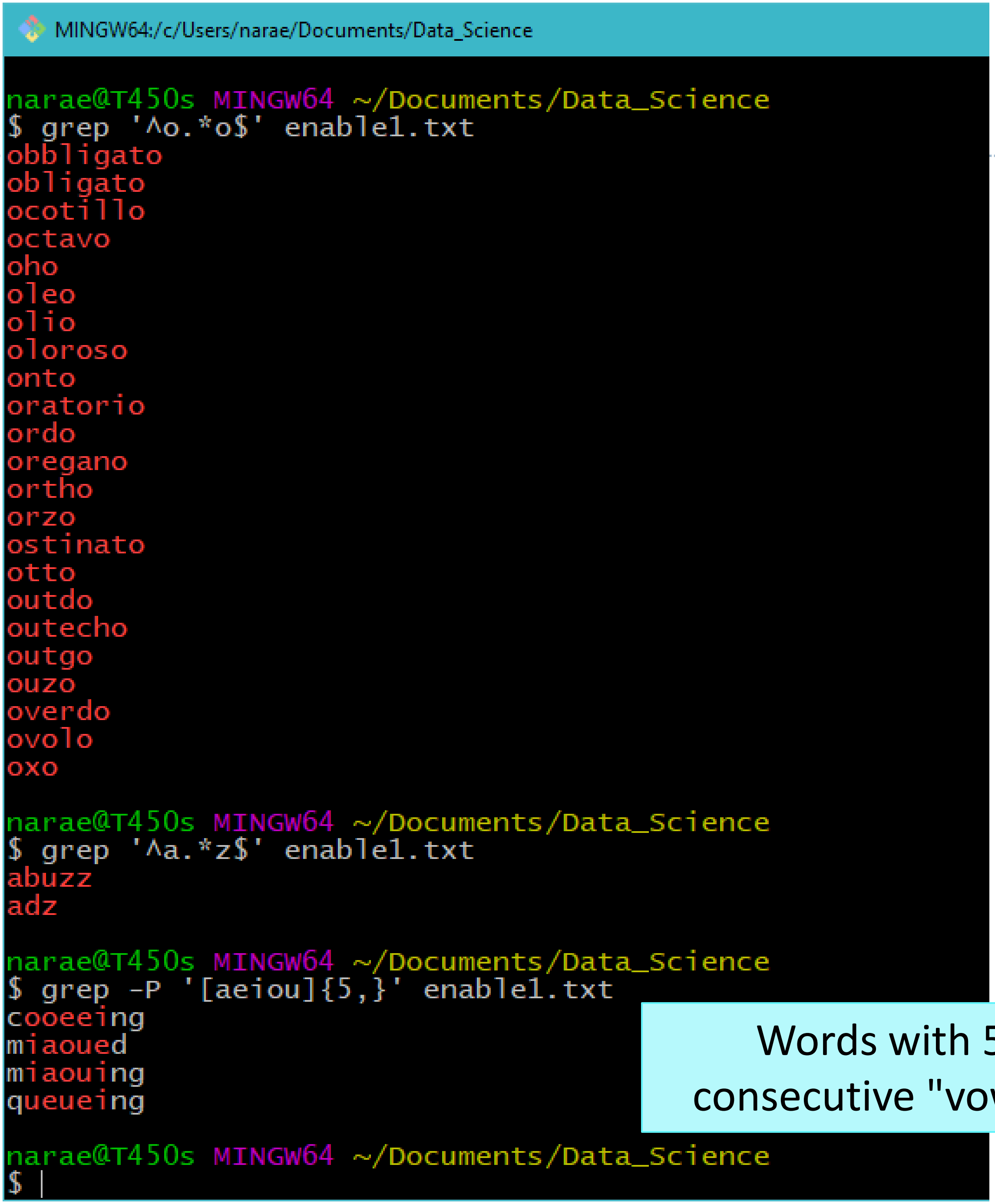


MINGW64:/c/Users/narae/Documents/Data_Science/Licensed-Data-Sets/ETS_Corpus_of_Non-Native_Written_English/data/text/prompts

```
$ 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/Tools/grep.shtml
- `grep -P`
  - Already on git-Bash & Linux
    - **Mac users**: use `egrep` or `grep -E`
  - Accepts **perl-style** regular expressions
  - Perl-style = Python-style! Can use \s, \d etc.

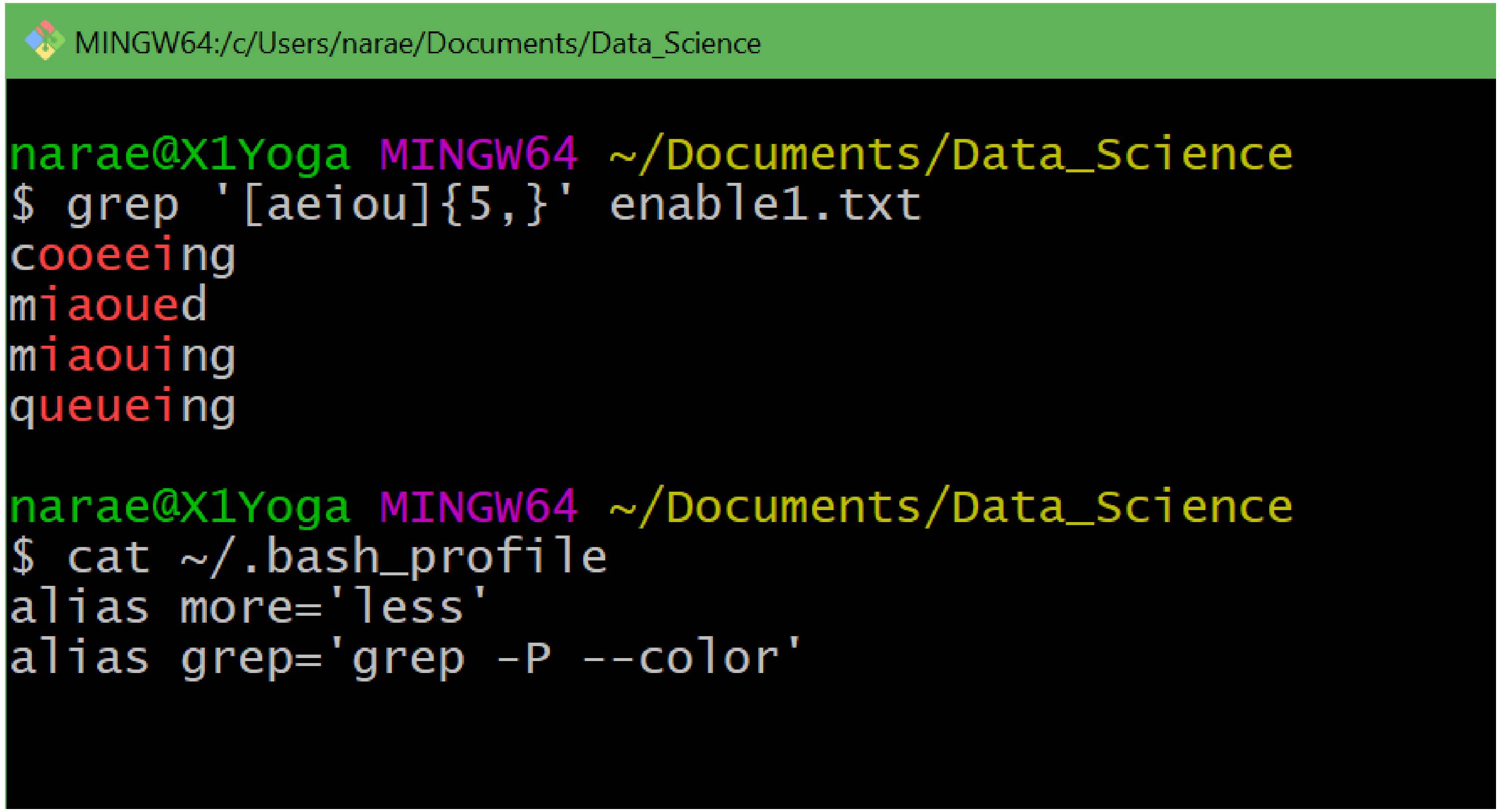


```
narae@T450s MINGW64 ~/Documents/Data_Science
$ grep '^o.*o$' enable1.txt
obbligato
obligato
ocotillo
octavo
oho
oleo
olio
oloroso
onto
oratorio
ordo
oregano
ortho
orzo
ostinato
otto
outdo
outecho
outgo
ouzo
overdo
ovolo
oxo

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 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:

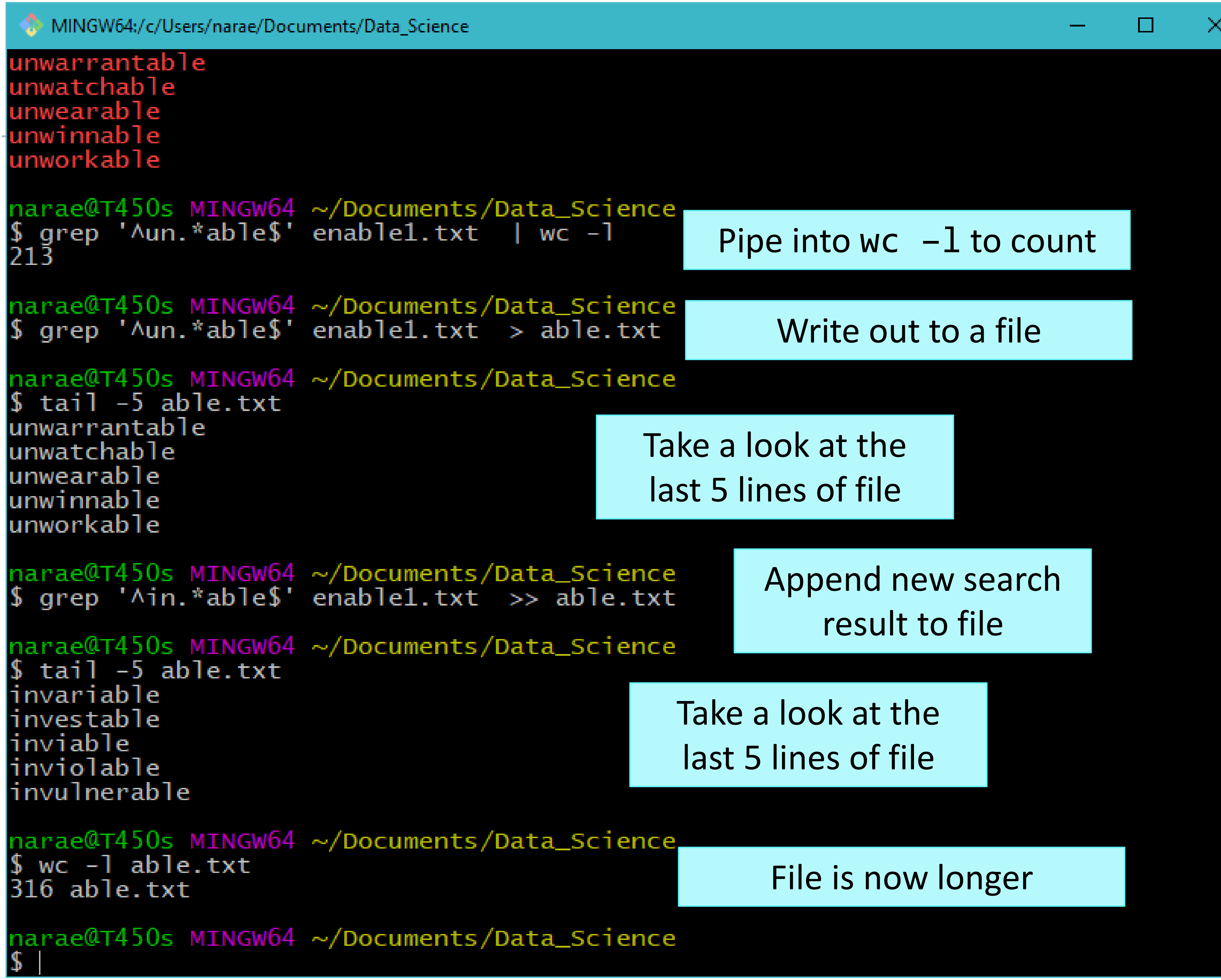


MINGW64:/c/Users/narae/Documents/Data_Science

```
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 `egrep` or `grep -E`

grep and piping, together



```
unwarrantable
unwatchable
unwearable
unwinnable
unworkable

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

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

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

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

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

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

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

Pipe into wc –l to count

Write out to a file

Take a look at the last 5 lines of file

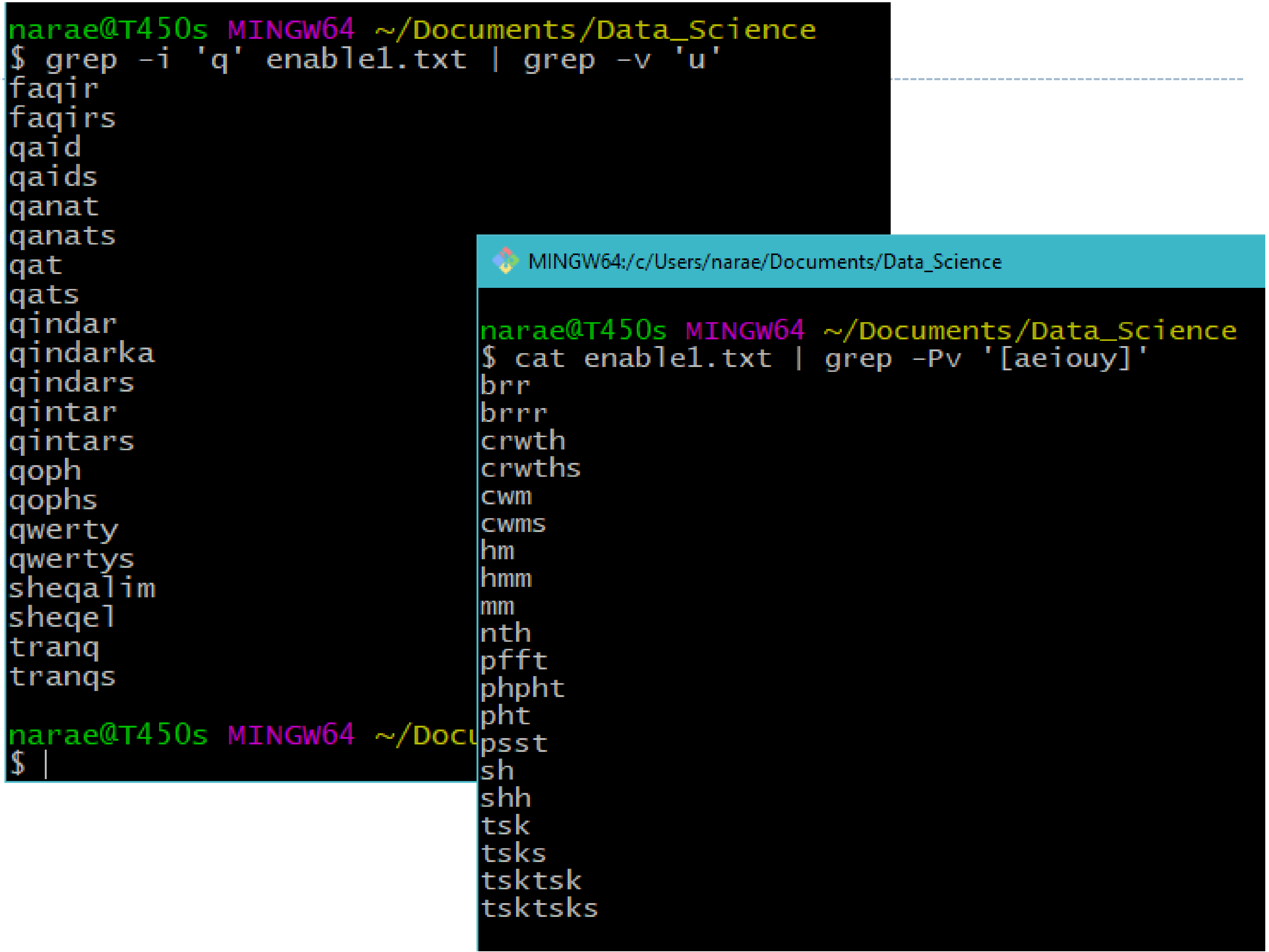
Append new search result to file

Take a look at the last 5 lines of file

File is now longer

grep -i, -v

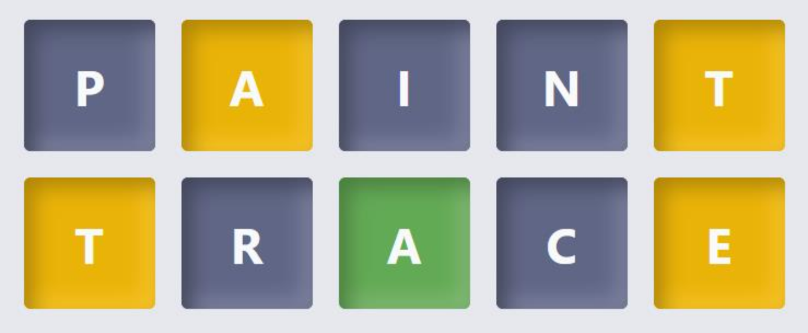
- `grep -i`
  - ignores case
- `grep -v`
  - prints lines that DO NOT match



```
narae@T450s MINGW64 ~/Documents/Data_Science
$ grep -i 'q' enable1.txt | grep -v 'u'
faqir
faqirs
qaid
qaids
qanat
qanats
qat
qats
qindar
qindarka
qindars
qintar
qintars
qoph
qophs
qwerty
qwertys
sheqalim
sheqel
tranq
tranqs

narae@T450s MINGW64 ~/Docu
$ |
```

```
narae@T450s MINGW64 ~/Documents/Data_Science
$ cat enable1.txt | grep -Pv '[aeiouy]'
brr
brrr
crwth
crwths
cwm
cwms
hm
hmm
mm
nth
pffft
phpht
pht
psst
sh
shh
tsk
tsks
tsktsk
tsktsks
```

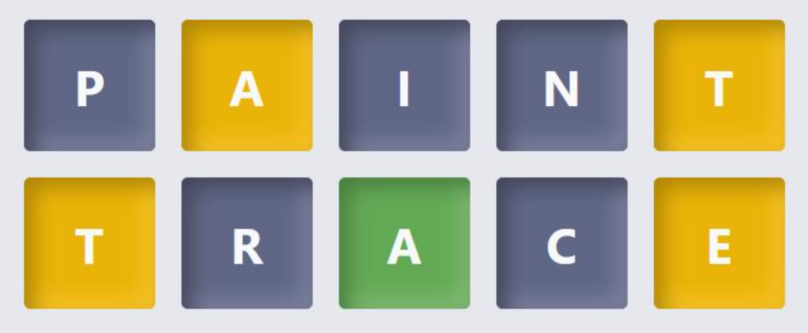
For fun: grepping WORDLE!



P	A	I	N	T
T	R	A	C	E

How to grep the solution on enable.txt?

Anatomy of WORDLE grep



P A I N T

T R A C E

```
grep '^.....$' enable1.txt |
grep -v '[pinrc]' |
grep 't' | grep 'e' |
grep '[^t][^a]a.[^te]'
```

filter in 5-letter words

filter out words with "absent" letters

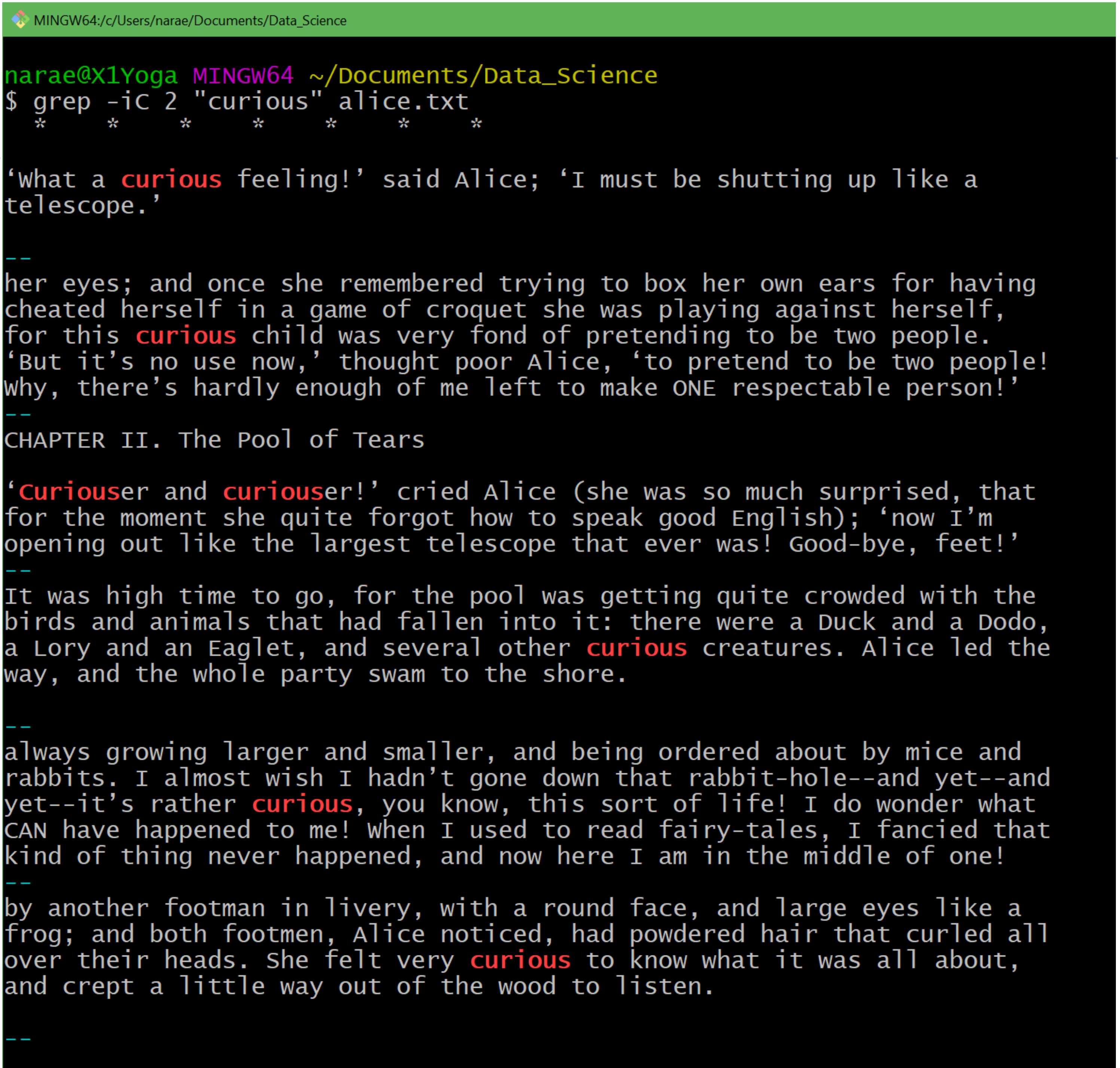
"present but not sure where" letters

positional pattern:
`a` → positively 'a' here
`[^te]` → no 't' or 'e' here
`.` → *any* letter

Each successive "pipe" narrows down the pool!

grep -C n

- `grep -C 2`
  - prints context: 2 lines before and after
  - ← capital C!

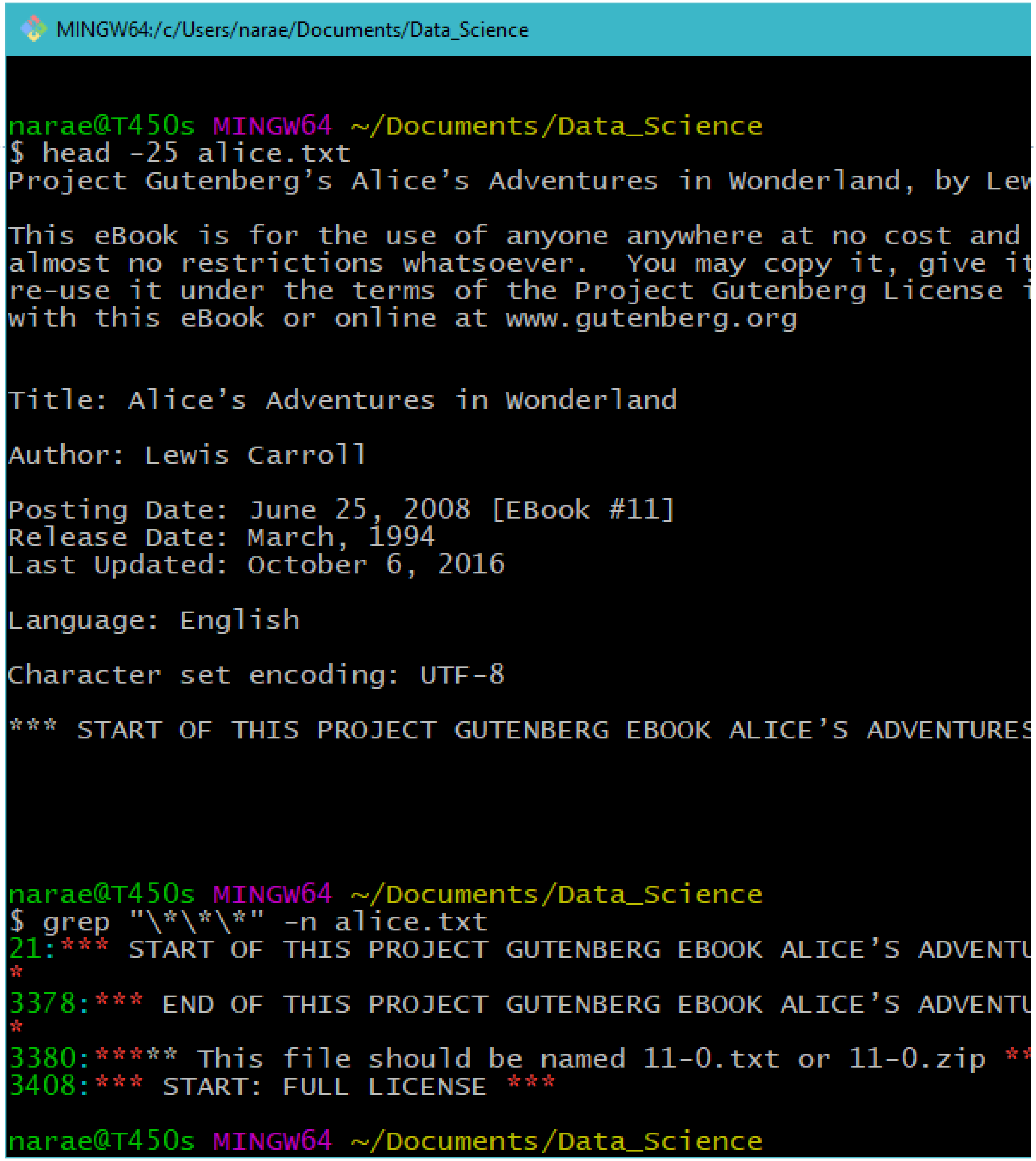


MINGW64:/c/Users/narae/Documents/Data_Science

```
narae@X1Yoga MINGW64 ~/Documents/Data_Science
$ grep -iC 2 "curious" alice.txt
  *    *    *    *    *    *    *

'What a curious feeling!' said Alice; 'I must be shutting up like a
telescope.'

--
her eyes; and once she remembered trying to box her own ears for having
cheated herself in a game of croquet she was playing against herself,
for this curious child was very fond of pretending to be two people.
'But it's no use now,' thought poor Alice, 'to pretend to be two people!
Why, there's hardly enough of me left to make ONE respectable person!'
--
CHAPTER II. The Pool of Tears

'Curiouser and curiouser!' cried Alice (she was so much surprised, that
for the moment she quite forgot how to speak good English); 'now I'm
opening out like the largest telescope that ever was! Good-bye, feet!'
--
It was high time to go, for the pool was getting quite crowded with the
birds and animals that had fallen into it: there were a Duck and a Dodo,
a Lory and an Eaglet, and several other curious creatures. Alice led the
way, and the whole party swam to the shore.

--
always growing larger and smaller, and being ordered about by mice and
rabbits. I almost wish I hadn't gone down that rabbit-hole--and yet--and
yet--it's rather curious, you know, this sort of life! I do wonder what
CAN have happened to me! When I used to read fairy-tales, I fancied that
kind of thing never happened, and now here I am in the middle of one!
--
by another footman in livery, with a round face, and large eyes like a
frog; and both footmen, Alice noticed, had powdered hair that curled all
over their heads. She felt very curious to know what it was all about,
and crept a little way out of the wood to listen.

--
```

grep -n

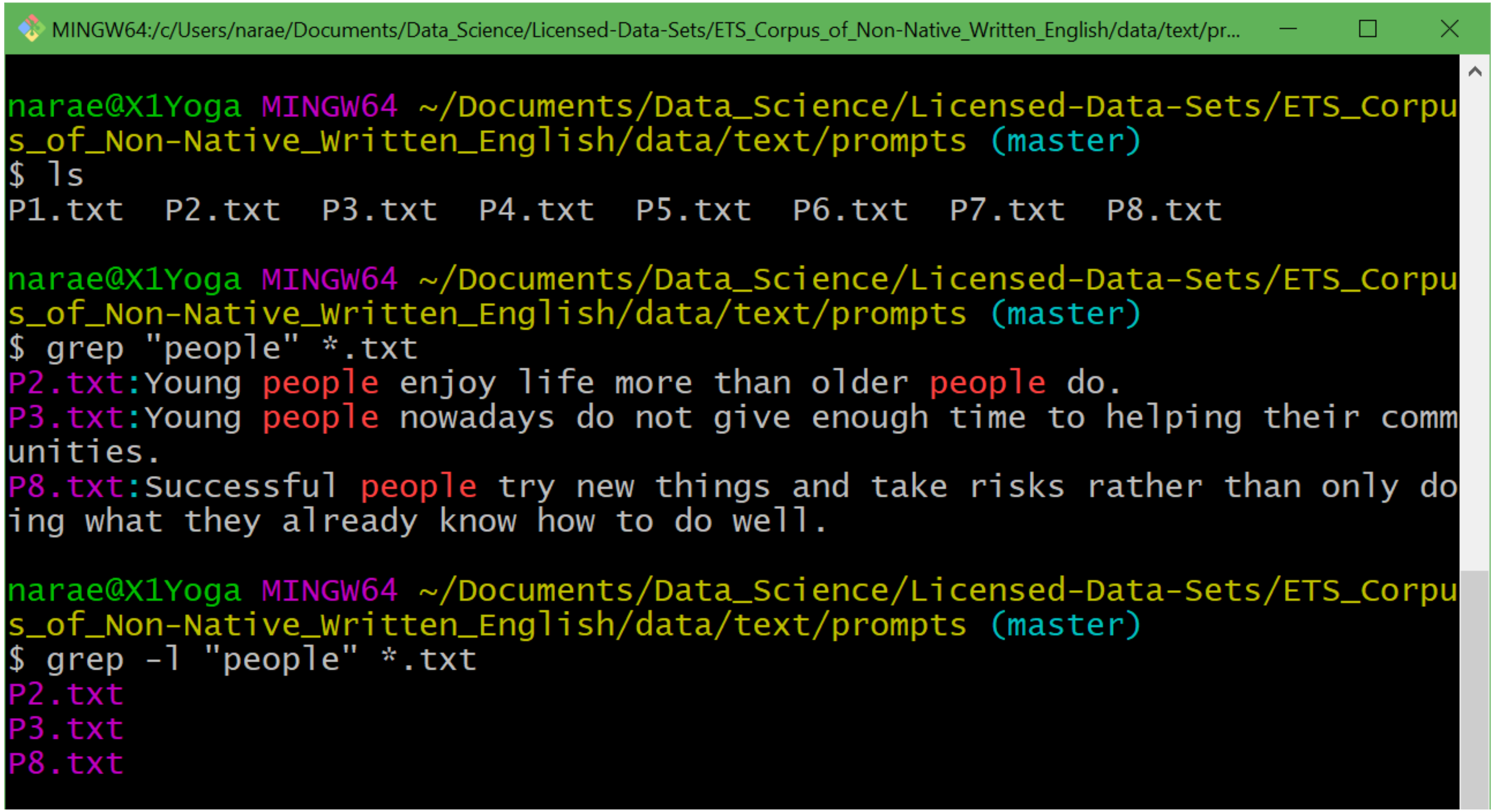
- `grep -n`
  - prints out line number



```
narae@T450s MINGW64 ~/Documents/Data_Science
$ head -25 alice.txt
Project Gutenberg’s Alice’s Adventures in Wonderland, by Lew

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




narae@T450s MINGW64 ~/Documents/Data_Science
$ grep "\*\*\*" -n alice.txt
21:*** START OF THIS PROJECT GUTENBERG EBOOK ALICE’S ADVENTU
*

3378:*** END OF THIS PROJECT GUTENBERG EBOOK ALICE’S ADVENTU
*

3380:***** This file should be named 11-0.txt or 11-0.zip **
3408:*** START: FULL LICENSE ***

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

Searching multiple files

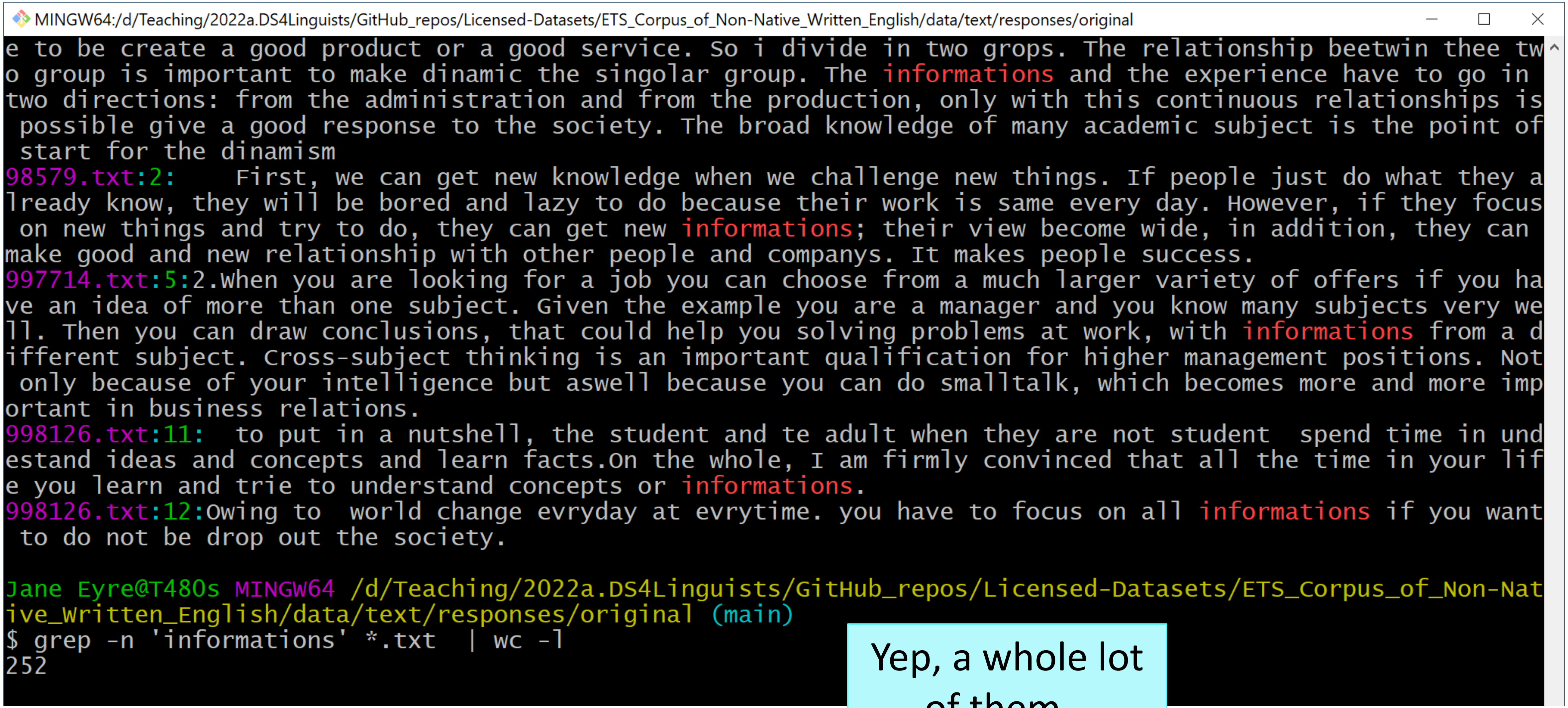
- `grep *.txt`
  - Searches through all files ending in .txt
- `grep -l`
  - prints file names *only if* a match is found



MINGW64:/c/Users/narae/Documents/Data_Science/Licensed-Data-Sets/ETS_Corpus_of_Non-Native_Written_English/data/text/pr...

```
narae@X1Yoga MINGW64 ~/Documents/Data_Science/Licensed-Data-Sets/ETS_Corpu
s_of_Non-Native_Written_English/data/text/prompts (master)
$ ls
P1.txt  P2.txt  P3.txt  P4.txt  P5.txt  P6.txt  P7.txt  P8.txt

narae@X1Yoga MINGW64 ~/Documents/Data_Science/Licensed-Data-Sets/ETS_Corpu
s_of_Non-Native_Written_English/data/text/prompts (master)
$ grep "people" *.txt
P2.txt:Young people enjoy life more than older people do.
P3.txt:Young people nowadays do not give enough time to helping their comm
unities.
P8.txt:Successful people try new things and take risks rather than only do
ing what they already know how to do well.

narae@X1Yoga MINGW64 ~/Documents/Data_Science/Licensed-Data-Sets/ETS_Corpu
s_of_Non-Native_Written_English/data/text/prompts (master)
$ grep -l "people" *.txt
P2.txt
P3.txt
P8.txt
```

“informations”?



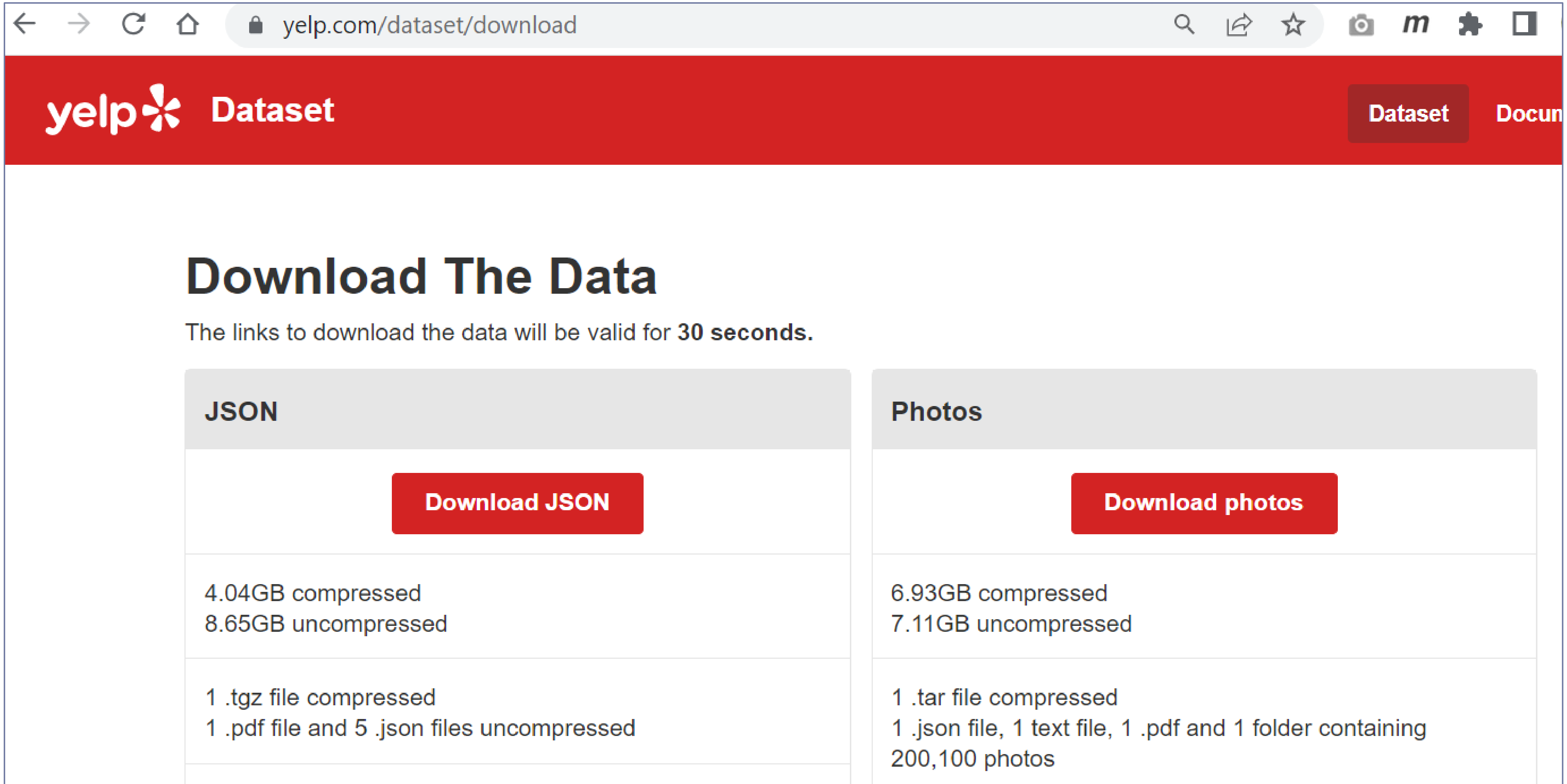
MINGW64:/d/Teaching/2022a.DS4Linguists/GitHub_repos/Licensed-Datasets/ETS_Corpus_of_Non-Native_Written_English/data/text/responses/original

```
e to be create a good product or a good service. So i divide in two grops. The relationship beetwin thee tw
o group is important to make dinamic the singolar group. The informations and the experience have to go in
two directions: from the administration and from the production, only with this continuous relationships is
 possible give a good response to the society. The broad knowledge of many academic subject is the point of
 start for the dinamism
98579.txt:2:    First, we can get new knowledge when we challenge new things. If people just do what they a
lready know, they will be bored and lazy to do because their work is same every day. However, if they focus
 on new things and try to do, they can get new informations; their view become wide, in addition, they can
make good and new relationship with other people and companys. It makes people success.
997714.txt:5:2.When you are looking for a job you can choose from a much larger variety of offers if you ha
ve an idea of more than one subject. Given the example you are a manager and you know many subjects very we
ll. Then you can draw conclusions, that could help you solving problems at work, with informations from a d
ifferent subject. Cross-subject thinking is an important qualification for higher management positions. Not
 only because of your intelligence but aswell because you can do smalltalk, which becomes more and more imp
ortant in business relations.
998126.txt:11:  to put in a nutshell, the student and te adult when they are not student  spend time in und
estand ideas and concepts and learn facts.On the whole, I am firmly convinced that all the time in your lif
e you learn and trie to understand concepts or informations.
998126.txt:12:Owing to  world change evryday at evrytime. you have to focus on all informations if you want
 to do not be drop out the society.

Jane Eyre@T480s MINGW64 /d/Teaching/2022a.DS4Linguists/GitHub_repos/Licensed-Datasets/ETS_Corpus_of_Non-Nat
ive_Written_English/data/text/responses/original (main)
$ grep -n 'informations' *.txt  | wc -l
252
```

Yep, a whole lot of them…

Bring on Big Data! The Yelp Dataset

- https://www.yelp.com/dataset

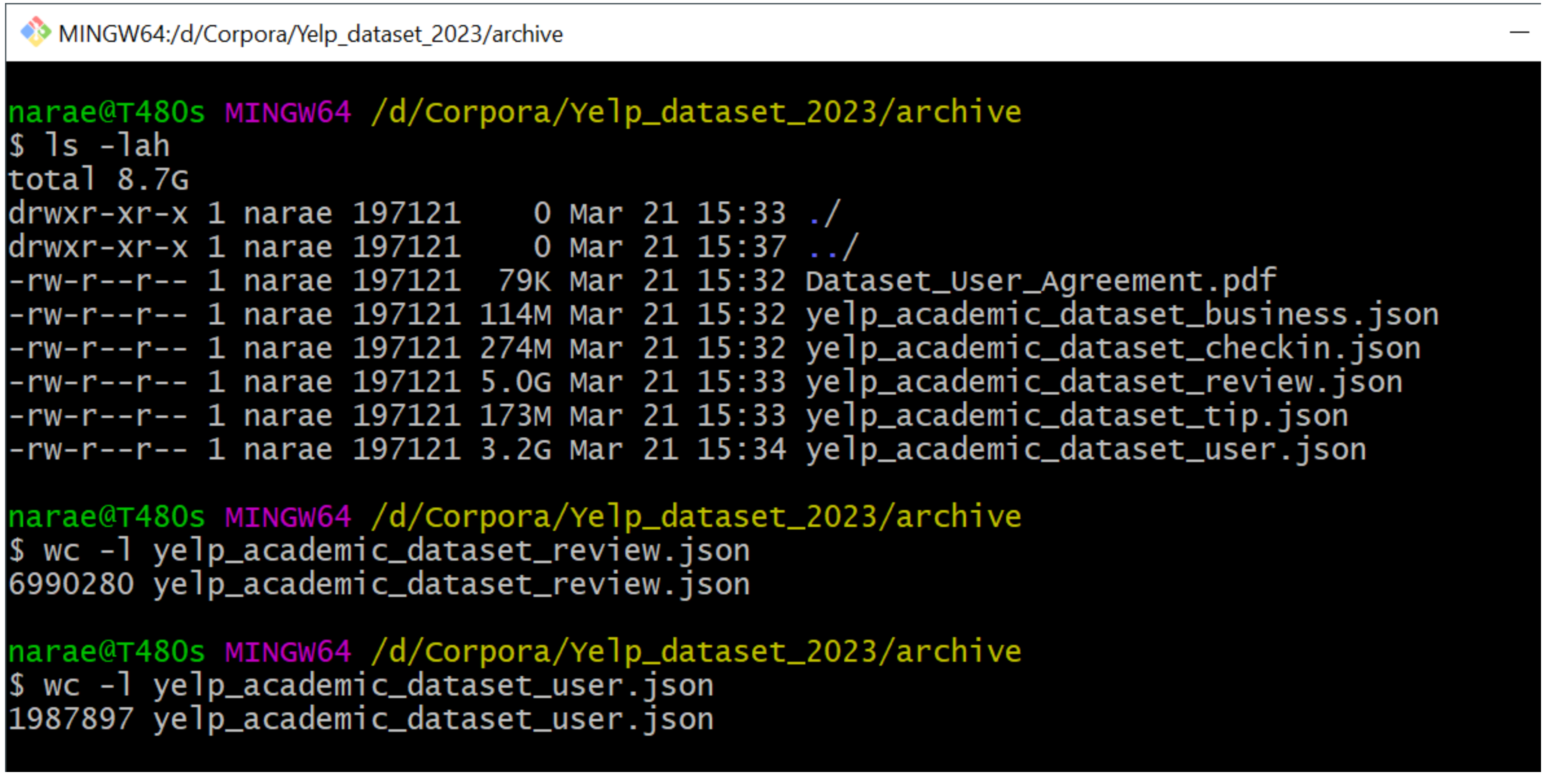


Download The Data

The links to download the data will be valid for **30 seconds.**

JSON	Photos
Download JSON	Download photos
4.04GB compressed 8.65GB uncompressed	6.93GB compressed 7.11GB 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 200,100 photos

Working with big data files



```
narae@T480s MINGW64 /d/Corpora/Yelp_dataset_2023/archive
$ ls -lah
total 8.7G
drwxr-xr-x 1 narae 197121    0 Mar 21 15:33 ./
drwxr-xr-x 1 narae 197121    0 Mar 21 15:37 ../
-rw-r--r-- 1 narae 197121  79K Mar 21 15:32 Dataset_User_Agreement.pdf
-rw-r--r-- 1 narae 197121 114M Mar 21 15:32 yelp_academic_dataset_business.json
-rw-r--r-- 1 narae 197121 274M Mar 21 15:32 yelp_academic_dataset_checkin.json
-rw-r--r-- 1 narae 197121 5.0G Mar 21 15:33 yelp_academic_dataset_review.json
-rw-r--r-- 1 narae 197121 173M Mar 21 15:33 yelp_academic_dataset_tip.json
-rw-r--r-- 1 narae 197121 3.2G Mar 21 15:34 yelp_academic_dataset_user.json

narae@T480s MINGW64 /d/Corpora/Yelp_dataset_2023/archive
$ wc -l yelp_academic_dataset_review.json
6990280 yelp_academic_dataset_review.json

narae@T480s MINGW64 /d/Corpora/Yelp_dataset_2023/archive
$ wc -l yelp_academic_dataset_user.json
1987897 yelp_academic_dataset_user.json
```

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

- review.json is 5GB with 7 million records (=lines)
- user.json is 3.2GB with 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 #13

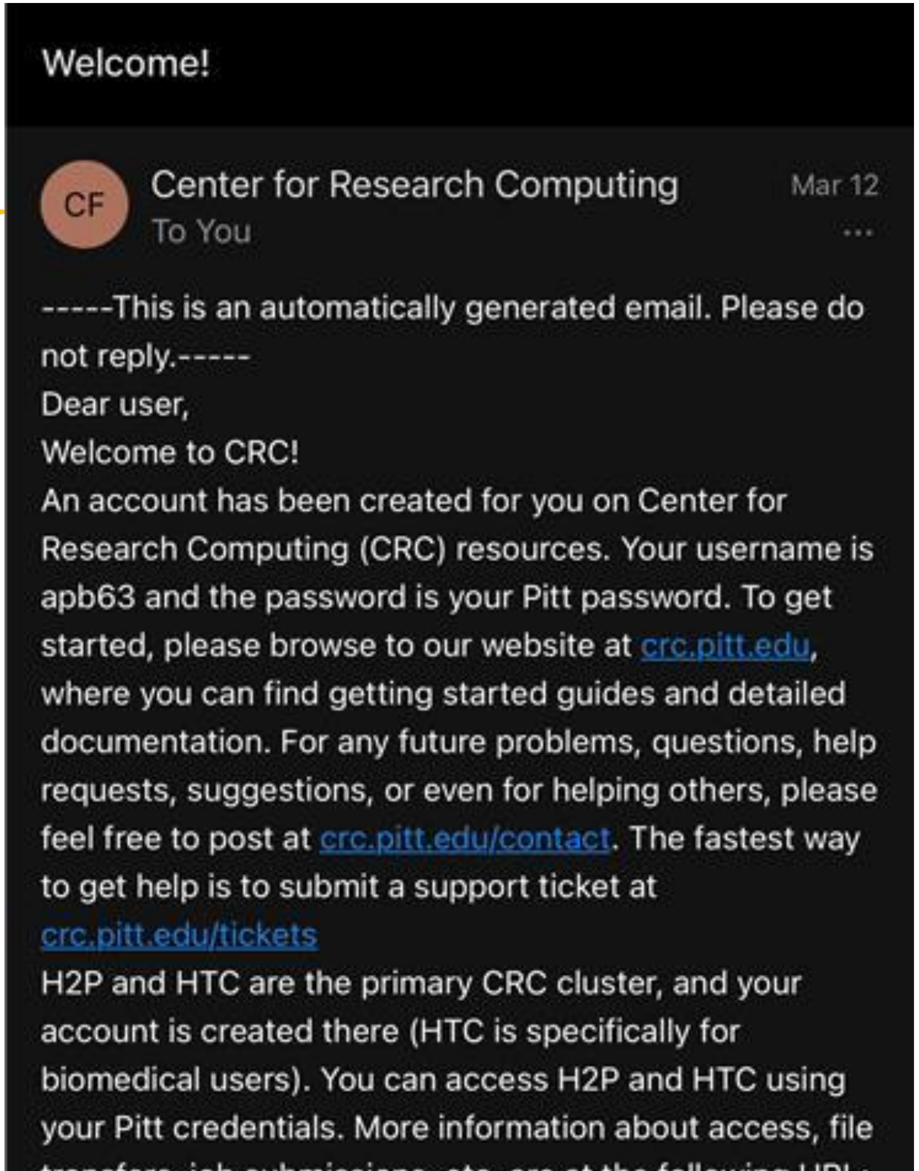
Let us now supercompute.

By Argonne National Laboratory's Flickr page - originally posted to Flickr as Blue Gene / PFrom Argonne National Laboratory Uploaded using F2ComButton, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=6412306

You got a supercomputing account.

- You received this mysterious email:

I got you all an account at Pitt's **Center for Research Computing** (CRC)



Welcome!

CF Center for Research Computing Mar 12
To You

-----This is an automatically generated email. Please do not reply.-----
Dear user,
Welcome to CRC!
An account has been created for you on Center for Research Computing (CRC) resources. Your username is apb63 and the password is your Pitt password. To get started, please browse to our website at crc.pitt.edu, where you can find getting started guides and detailed documentation. For any future problems, questions, help requests, suggestions, or even for helping others, please feel free to post at crc.pitt.edu/contact. The fastest way to get help is to submit a support ticket at crc.pitt.edu/tickets
H2P and HTC are the primary CRC cluster, and your account is created there (HTC is specifically for biomedical users). You can access H2P and HTC using your Pitt credentials. More information about access, file transfers, job submissions, etc. are at the following URL:

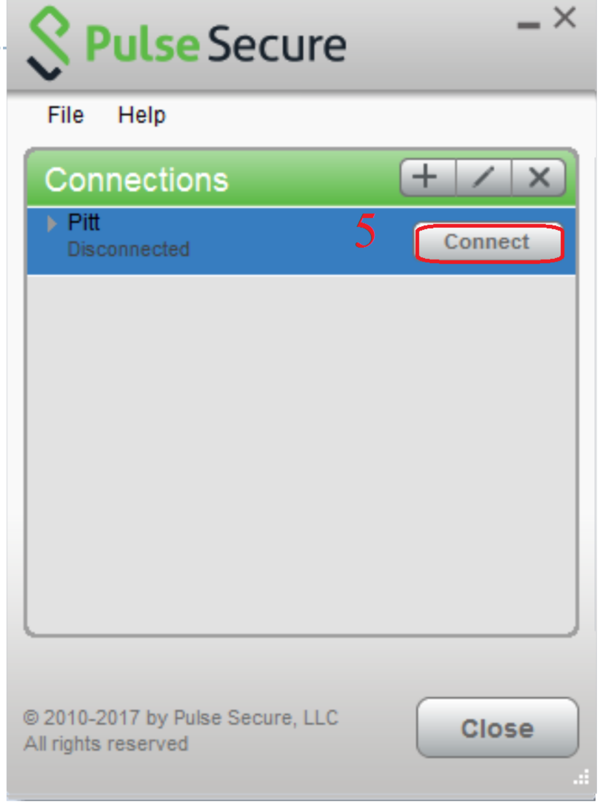
- CRC: Center for Research Computing
  - https://crc.pitt.edu
  - Handy links in "Resource" page!

Accessing CRC's cluster

- If you're **OFF CAMPUS**, your laptop should be running a **Secure Remote Access client.**
  - Install and run PulseSecure →
  - Details in the h2p cluster user guide: https://crc.pitt.edu/resources/h2p-user-guide
- Remote-access your account via SSH:
  - `ssh yourpittid@h2p.crc.pitt.edu`
- Getting your bearings:
  - Where are you? `pwd`
  - What is your user 'group'? `groups`
  - Is python installed on this machine? `which python`
  - What are your configuration files? `ls -a`
    - `.bash_profile`

      ← Customize with your own aliases, etc.
    - `.bash_history`

      ← Bash commands you typed in are logged here.



Wrapping up

- To-do #13
  - Fun with big(ish) data -- the Yelp Dataset! https://www.yelp.com/dataset/
  - 4Gb zipped, downloading takes 10+ minutes. Allocate enough time for this assignment, especially if you are new to command line.