BINF2111 – Introduction to Bioinformatics Computing

UNIX 101 part trois (grep regrex and sed)

Richard Allen White III, PhD
RAW Lab

Lecture 4 - Thursday Aug 29th, 2024

Learning Objectives

- Review quiz and bonus
- Regular expressions in grep
- Sed
- Regular expressions in sed
- Quiz 4

Carnegie rule

Carnegie Rule is a rule of thumb suggesting how much outside-of-classroom study time is required to succeed in an average higher education course in the U.S. system.

Is for every hour spent in the classroom that two or more hours of outside work required.

Carnegie rule

Carnegie Rule is a rule of thumb suggesting how much outside-of-classroom study time is required to succeed in an average higher education course in the U.S. system.

Is for every hour spent in the classroom that two or more hours of outside work required.

OUTDATED!

RAW rule of thumb for computational learning is spend quality time at the terminal, googling, and thinking problems at the terminal..

Learning how to code??

As they say ‘Rome wasn’t build in a day,’ to learn a new language or coding language – it takes PRACTICE!

Learning how to code??

As they say ‘Rome wasn’t build in a day,’ to learn a new language or coding language – it takes PRACTICE!

EVERYDAY PRACTICE!

Learning how to code??

As they say ‘Rome wasn’t build in a day,’ to learn a new language or coding language – it takes PRACTICE!

EVERYDAY PRACTICE!

Study groups, outside examples, and reading – helps.

BUT PRACTICE, PRACTICE, PRACTICE!

Missed Quiz questions - 1

What does grep stand for?

Missed Quiz questions - 1

What does grep stand for?

global search for regular expression and print the result

Missed Quiz questions - 2

What is not a way to grab a file from github?

1 - wget
2 - curl
3 - mv
4 - cd

Missed Quiz questions - 2

What is **not** a way to grab a file from github?

1 - wget
2 - curl
3 - mv
4 - cd

Missed Quiz questions - 2

What is **not** a way to grab a file from github?

1 - wget
2 - curl
3 - mv
4 - cd

3 and 4 only

Missed Quiz questions - 3

^ is at the front of character or line

$ is at the end of the character or line

[ ] matches all

T or F

Missed Quiz questions - 3

^ is at the front of character or line

$ is at the end of the character or line

[ ] matches all

T or F

Challenge question 1

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTT

T or F

Challenge question 1

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTT

T or F

Challenge question 2

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTTAT

T or F

Challenge question 2

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTTAT

T or F

Challenge question 2

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTTAT

How would you count it?

Challenge question 2

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTTAT

What this work?
grep -c “AT” string.fna

Challenge question 2

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTTAT

What this work?
grep -o -c “AT” string.fna or
grep -oc “AT” string.fna

Challenge question 2

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTTAT

What this work?
grep -Eo “AT” string.fna --color
grep -o “AT” string.fna

Challenge question 2

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTTAT

What this work?
grep -Eo “AT” string.fna | wc -l
grep -o “AT” string.fna | wc -l

Challenge question 2

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTTAT

What this work?
grep -o “^AT|AT$” string.fna

Challenge question 2

- grep -c “AT” string.fna command is able to count all AT's in this line?

ATCCGCGCGCGCGGCGCGCGCGCGCGGCGCG
CGCGTTAT

What this work?
grep -Eo “^AT|AT$” string.fna | wc -l
egrep -o “^AT|AT$” string.fna | wc -l

grep – syntax to hands of UNIX

grep [option] pattern file

Understanding Regular Expressions:

^ (Caret) match expression at the start of a line, as in ^A.

$ (Question) match expression at the end of a line, as in A$.

\ (Back Slash) turn off the special meaning of the next character, as in \^. To look for a Caret "^" at the start of a line, the expression is ^\^.

[ ] (Brackets) match any one of the enclosed characters, as in [aeiou]. Use Hyphen "-" for a range, as in [0-9].

[^] match any one character except those enclosed in [ ], as in [^0-9].

. (Period) match a single character of any value, except end of line. So b.b will match "bob", "bib", "b-b", etc.

* (Asterisk) match zero or more of the preceding character or expression. An asterisk matches zero or more of what precedes it. Thus [A-Z]* matches any number of upper-case letters, including none, while [A-Z][A-Z]* matches one or more upper-case letters.

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep command print all the lines containing ‘AT’ with line number on the line which they occur?

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep command print all the lines containing ‘AT’ with line number on the line which they occur?

grep -n “AT” example.fasta

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

What happens when I do:

grep -cn “AT” example.fasta ?

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

What happens when I do:

grep -cn “AT” example.fasta ?

Six? Is that right?

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep command the count all ‘AT’ within the example.fasta?

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep command the count all ‘AT’ within the example.fasta?

grep -c “AT” example.fasta

What happens?

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep command the count all ‘AT’ within the example.fasta?

grep -c “AT” example.fasta

What happens?

SIX right?

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep command the count all ‘AT’ within the example.fasta that counts all?

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep command the count all ‘AT’ within the example.fasta?

grep -o “AT” example.fasta | wc -l

SEVEN

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts ‘kite’ in beginning of the string only for kite_data.txt

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts ‘kite’ in beginning of the string only for kite_data.txt

```
grep ‘^kite’ kite_data.txt | wc -l
```

or

```
grep -c ‘^kite’ kite_data.txt
```

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts ‘kite’ in beginning of the string only for kite_data.txt

```
grep ‘^kite’ kite_data.txt | wc -l
```

or

```
grep -c ‘^kite’ kite_data.txt
```

What does the -w option do instead of -c?

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts ‘kite or Kite’ in beginning of the string only for kite_data.txt

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts ‘kite or Kite’ in beginning of the string only for kite_data.txt

```
grep ‘^[Kk]ite’ kite_data.txt | wc -l
```

or

```
grep -c ‘^[Kk]ite’ kite_data.txt
```

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts ‘kite’ in end of the string only for kite_data.txt

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts ‘kite’ in end of the string only for kite_data.txt

```
grep ‘kite$’ kite_data.txt | wc -l
```

or

```
grep -c ‘kite$’ kite_data.txt
```

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts all words ‘kite’ but not ‘Kite’ for kite_data.txt

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts all words ‘kite’ but not ‘Kite’ for kite_data.txt

```
grep ‘kite’ kite_data.txt | wc -l
```

or

```
grep -c ‘kite’ kite_data.txt
```

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts all words ‘kite’ and ‘Kite’ for kite_data.txt

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts all words ‘kite’ and ‘Kite’ for kite_data.txt

```
grep ‘[Kk]ite’ kite_data.txt | wc -l
```

or

```
grep -c ‘[Kk]ite’ kite_data.txt
```

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

Write a grep regex command that counts all words ‘kite’ and ‘Kite’ and ‘red’ for kite_data.txt

grep examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

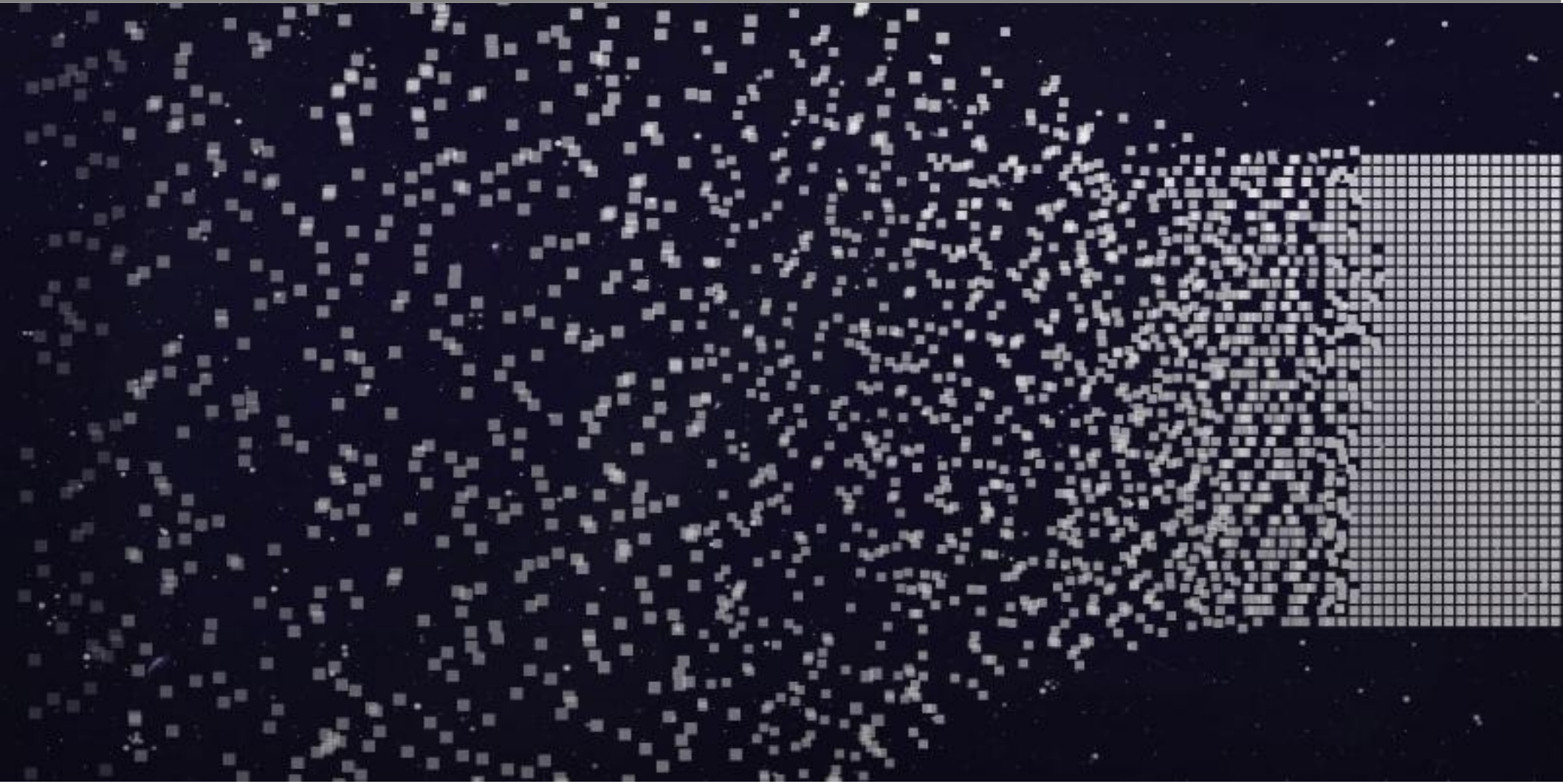
Write a grep regex command that counts all words ‘kite’ and ‘Kite’ and ‘red’ for kite_data.txt

```
grep -Eo ‘[Kk]ite|red’ kite_data.txt | wc -l
```

or

```
egrep -o ‘[Kk]ite|red’ kite_data.txt | wc -l
```

Data science



Data science

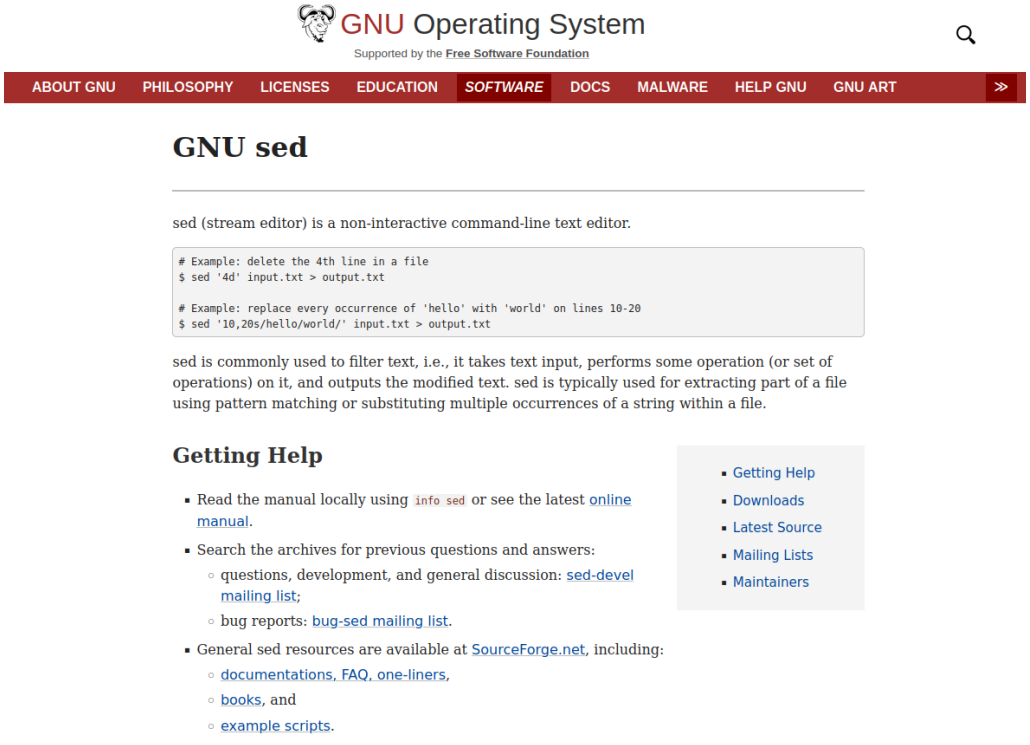
Sed - Stream Editor

sed ("stream editor") is a Unix utility that parses and transforms text, using a simple, compact programming language. sed was developed from 1973-1974 by Lee E. McMahon of Bell Labs

Lee E. McMahon

Sed - Stream Editor

sed ("stream editor") is sed was one of the earliest tools to support regular expressions, and remains in use for text processing, most notably with the substitution command.
Written in C
www.gnu.org/software/sed/



Sed - Stream Editor

Popular alternative tools for plaintext string manipulation and "stream editing" include AWK and Perl.

GNU Operating System

Supported by the Free Software Foundation

ABOUT GNU | PHILOSOPHY | LICENSES | EDUCATION | SOFTWARE | DOCS | MALWARE | HELP GNU | GNU ART | »

GNU sed

sed (stream editor) is a non-interactive command-line text editor.

```
# Example: delete the 4th line in a file
$ sed '4d' input.txt > output.txt

# Example: replace every occurrence of 'hello' with 'world' on lines 10-20
$ sed '10,20s/hello/world/' input.txt > output.txt
```

sed is commonly used to filter text, i.e., it takes text input, performs some operation (or set of operations) on it, and outputs the modified text. sed is typically used for extracting part of a file using pattern matching or substituting multiple occurrences of a string within a file.

Getting Help

- Read the manual locally using `info sed` or see the latest online manual.
- Search the archives for previous questions and answers:
  - questions, development, and general discussion: sed-devel mailing list;
  - bug reports: bug-sed mailing list.
- General sed resources are available at SourceForge.net, including:
  - documentations, FAQ, one-liners,
  - books, and
  - example scripts.

- Getting Help
- Downloads
- Latest Source
- Mailing Lists
- Maintainers

Sed - Stream Editor

```
s/he/she/g
```

Sed - Stream Editor

sed OPTIONS... [SCRIPT] [INPUTFILE…]

--version (Print out the version of sed)

--help (help page)

-n, --quiet, --silent (suppress automatic printing of pattern space)

--debug

-e script, --expression=script (Add the commands in script to the set of commands to be run)

-i [SUFFIX], –in-place[=SUFFIX] (This option specifies that files are to be edited in-place)

Sed - Stream Editor

Substitution command

S command swiss army knife

sed 's/regexp/replacement/g' inputFileName > outputFileName

The **s** stands for substitute, while the **g** stands for global, which means that all matching occurrences in the line would be replaced.

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

echo “Hello World” >>file.txt

Write a sed command the replaces “World” with “There”

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

echo “Hello World” >>file.txt

Write a sed command the replaces “World” with “There”

sed ‘s/World/There/g’ file.txt >>file1.txt

What happens when I do

sed ‘s/World/There/g’ file.txt

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

What happens when I do ?

sed ‘s/World/There/g’ file.txt

Then

More file.txt?

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

What if I want to replace World with There in file.txt?

sed -i ‘s/World/There/g’ file.txt

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

What happens when I do

echo “Hello World” | sed 's/World/There/g'

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

What happens when I do

echo “Hello World” | sed 's/World/There/g'

Prints Hello There right?

Sed – S swiss army knife

sed ‘s/regexp/replacement/flags’.

\L - Turn the replacement to lowercase until a \U or \E is found
\l - Turn the next character to lowercase,
\U - Turn the replacement to uppercase until a \L or \E is found,
\u -Turn the next character to uppercase,
\E - Stop case conversion started by \L or \U.
g - Apply the replacement to all matches to the regexp, not just the first.
d - Delete the pattern space; immediately start next cycle.
a comment, until the next newline.

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

From the kite_data.txt

Write a sed command that makes all the lower case upper case?

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

From the kite_data.txt

Write a sed command that makes all the lower case upper case?

```
sed 's/[a-z]/\U&/g'
```

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

From the kite_data.txt

Write a sed command that makes all the upper case lower case?

sed 's/[A-Z]/\U&/g'

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

From the kite_data.txt

Write a sed command that makes all the upper case lower case?

sed 's/[A-Z]/\l&/g'

or

sed ‘s/[A-Z]/\L&/g'

Sed – delete

```
sed '/^ *$/d' inputFileName.
```

- The caret (^) matches the beginning of the line.
- The dollar sign ($) matches the end of the line.
- The asterisk (*) matches zero or more occurrences of the previous character.
- The plus (+) matches one or more occurrence(s) of the previous character.
- The question mark (?) matches zero or one occurrence of the previous character.
- The dot (.) matches exactly one character.

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

What happens when I do

```
echo “Hello Three” >file.txt
echo “Hello World” >>file.txt

sed -i '/Three/d' file.txt
```

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

I want you do delete all the empty lines using sed?

sed examples

Activities Terminal Mon 18:50

docwhite@system76-pc: ~

File Edit View Search Terminal Help

```
(base) docwhite@system76-pc:~$
```

I want you do delete all the empty lines using sed?

sed '/^$/d' empty_lines.txt

Bonus 4a

- Count both the number of AT and GC in one grep command and in another command print the line number which they appear?

Bonus 4b

- Delete all the empty lines in the empty lines file with
→ grep
→ awk

- Delete all the ‘all white space’ with grep
→ grep
→ awk

Also, in python (think Pandas)

Quiz 4

- On canvas now