School of Computer Science

COMP47470

Lab 2
CLI (Bash) for Big Data

Teaching Assistant:	Leandro Batista de Almeida
Coordinator:	Anthony Ventresque
Date:	Thursday 1st February, 2018
Total Number of Pages:	2

1 Snapshot of a Social Network (facebookdata.csv)

One of the first tasks that data scientists do when they receive a new dataset is to check the data: reading its content, understanding its format, etc.

1. read the first 10 lines of the file. What are the different fields?
2. read the last 10 lines of the file
3. print line 1515 using `tail`, `head`, and a pipe
4. count the number of characters in the first 50 lines ((use `head`, `wc` and a pipe)
5. try to print only columns 4 and 6 (use `cut`). you should realise there is something wrong with the file. What is the problem?
6. fix the issue using a script and sed. Your script should look like that:

```
#!/bin/bash

#we do a bit of copying to make sure we don't corrupt the original file
cp facebookdata.csv facebookdata-clean.csv
while grep -q 'pattern-with-a-bad-character' facebookdata-clean.csv ;do
    #check what the -i option of sed does
        sed -i.bak 's/\(before\)bad-character\(after\)/\1;\2/'
            facebookdata-clean.csv
done
```

   Try first to find the correct regular expression with grep. What is the "bad character"? Where does it appear? What character do you need to see before? After? What do you think the patter is like? Try without the script first, like in:

```
$> grep 'regex' facebookdata.csv
```

   where you will replace regex with a pattern that you think represents the "error" in the file.

7. print only columns 4 and 6 now
8. now print only the lines containing "Tim Duncan"
9. print column 6 of the lines containing "Tim Duncan"

Now write commands and scripts to answer the following questions on the dataset:

10. How many status of each type is there? (remember exercise 2 of your first lab)
11. Find the 10 most popular status entries. For that, add all the values you find in columns 8-15. Your script should look like:

```
#!/bin/bash

#declare 10 variables (initialise them with a 0)
num_comments=0
```

```
num_shares=0
num_likes=0
etc.
#here you're reading the output of a command line by line
for line in $(command-similar-to-previous-question); do
    #get the values (cut) in several variables:
    num_comments=???
    num_shares=???
    num_likes=???
    etc.
    #add the values
    #keep only this sum if it's among the top 10:
    #think insertion sort?
done
#print the 10 status entries
```

Arithmetic expansion and evaluation in Bash is done by placing an integer expression using the following format `$((expression))`, e.g., `$(( n1n2 ))`+. `$(command)`, or `‘command‘` is a command substitution (gives you the outcome of the command).