Comp1204 Unix Coursework

Charles Gilbert 31198317 cg4g19@soton.ac.uk

25 February 2020

0.1   Countreviews Script

```
#!/bin/bash
for hotel in $@/*
```

My code begins by telling the OS to use the bash shell to execute the script. It then begins a loop through all the files within the folder specified using hotel as the variable name.

```
filename=$(echo $hotel | sed -r 's/.*\/hotel/hotel/' | sed 's/.dat//')
```

For each hotel my code first extracts the name of the file, for example hotel_2388. In order to do this it must first use echo which will output the filename rather than the content of the file. It then takes the filename and uses sed with the regex option and replaces everything up until the underscore and with hotel. This removes the long filename and doesnt matter whether an absolute or relative path is used. This is then piped into a simpler sed which removes the .dat filename

```
grep -E 'Author|Content' $hotel | wc -l | awk '{print $0/2}' | awk -v var="$fi
```

Line five begins by searching the file for Author or Content and then counting the amount of lines. I then use awk to divide this number by two. I have done it this way for security reasons. If just lines with Author or ¡Author¿ are checked then if the review content contains the word Author then the hotel will be seen to have more reviews than there actually are. There are two fields that the user could possibly try to manipulate the script. Those two fields are the author and the content field. This value is then piped into another awk command which uses the variable option with the previously created filename variable to output a line with the hotelname followed by the amount of reviews for the hotel.

```
done | sort -nrk 2,2
```

Line six then pipes the entire output into a sort command which uses options n to sort by numbers, r to reverse the output and k to sort using two columns

```
1  #!/bin/bash
2  for hotel in $@/*
3  do
4  filename=$(echo $hotel | sed -r 's/.*\/hotel/hotel/' | sed 's/.dat//')
5  grep -E 'Author|Content' $hotel | wc -l | awk '{print $0/2}' | awk -v var="$fi
6  done | sort -nrk 2,2
```

1